



Wowza Streaming Engine™

Server-Side API

Wowza Streaming Engine: Server-Side API



Version 4.1

Copyright © 2006 – 2014 Wowza Media Systems, LLC
<http://www.wowza.com>

This document is for informational purposes only and in no way shall be interpreted or construed to create warranties of any kind, either express or implied, regarding the information contained herein.

No Endorsement or Warranty for Third-Party Links and Software

This document contains links to third-party websites ("Linked Sites") that are not under the control of Wowza® Media Systems, LLC ("Wowza"). Wowza is not responsible for the content on or operation of Linked Sites. If you access Linked Sites, you do so at your own risk and understand that Wowza accepts no responsibility or liability for the content or operation of Linked Sites. Wowza provides these links only as a convenience, and the inclusion of a link does not imply that Wowza endorses such Linked Sites or any content, products, or services available from Linked Sites.

This document also refers to third-party software that is not licensed, sold, or distributed by Wowza (collectively, "Third-Party Software"). Wowza does not endorse, is not responsible for, and accepts no liability related to Third-Party Software. Please ensure that any and all use of Wowza® software and third-party software is properly licensed.

Wowza Trademarks

Wowza®, Wowza Media Systems, Wowza Streaming Engine™, along with other trademarks, logos, trade dress, and other proprietary colors and markings, are each trademarks or registered trademarks of Wowza in the United States and in other countries (collectively, "Wowza Marks"). No right to use Wowza Marks in any way is granted hereunder. Contact sales@wowza.com for information on obtaining the right to use Wowza Marks. Any use of Wowza Marks, authorized or otherwise, shall inure to the sole benefit of Wowza.

Third-Party Trademarks and Copyrights

Trademarks, product names, logos, designs, trade dress, and other proprietary markings of non-Wowza third parties (collectively, "Third-Party Marks") may be trademarks or registered trademarks of their respective owners. Use of Third-Party Marks is for the sole purpose of identifying third-party products and services and does not represent endorsement, sponsorship, partnership, or other affiliation between Wowza and such third parties.

A list of applicable copyright notices related to content in this document is available on the [Wowza website](#).

Document History

Version	Description	Date
Doc v4.1.0	Initial document release for Wowza Streaming Engine 4.1	09-02-2014
Doc v4.1.1	Updated for Wowza Streaming Engine 4.1.1	11-21-2014

Note

A more recent version of this document may be available online. See the [Wowza Media Systems Documentation webpage](http://www.wowza.com/forums/content.php?188-documentation) (<http://www.wowza.com/forums/content.php?188-documentation>) for the latest updates.

Package

com.wowza.io

com.wowza.io Interface IRandomAccessReader

public interface **IRandomAccessReader**
extends

Field Summary

public static final	<u>FORWARD</u> Value: 1
public static final	<u>REVERSE</u> Value: -1

Method Summary

void	<u>close()</u> Close the media asset
boolean	<u>exists()</u> Does the media asset exist
String	<u>getBasePath()</u> Get the basePath for the random access reader
int	<u>getDirecton()</u> Get the current direction hint for the random access reader.
long	<u>getFilePointer()</u> Get the current byte location in the media asset
String	<u>getMediaExtension()</u> Get the media extension
String	<u>getMediaName()</u> Get the media name
String	<u>getPath()</u> Get the unique path to the media asset item
void	<u>init(IApplicationInstance appInstance, IMediaStream stream, String basePath, String mediaName, String mediaExtension)</u> Intialize RandomAccessReader
boolean	<u>isOpen()</u> Is the media asset open
long	<u>lastModified()</u> Return the lastModified date (same format as File.lastModified)
long	<u>length()</u> Get the media asset length in bytes

void	<code>open()</code> Open the media asset
int	<code>read(byte[] buf, int off, int size)</code> Read bytes from the media asset
void	<code>seek(long pos)</code> Seek to a position in the media asset
void	<code>setDirecton(int directon)</code> Set the current direction hint

Fields

FORWARD

```
public static final int FORWARD
```

Constant value: **1**

REVERSE

```
public static final int REVERSE
```

Constant value: **-1**

Methods

init

```
public void init(IApplicationInstance appInstance,  
                IMediaStream stream,  
                String basePath,  
                String mediaName,  
                String mediaExtension)
```

Intialize RandomAccessReader

Parameters:

appInstance - application instance
stream - parent stream if one exists
basePath - basePath for IApplicationInstance
mediaName - media name
mediaExtension - media extension from mediaReaders

open

```
public void open()  
    throws java.io.IOException
```

Open the media asset

Throws:

IOException

close

```
public void close()  
    throws java.io.IOException
```

Close the media asset

Throws:

IOException

isOpen

```
public boolean isOpen()
```

Is the media asset open

Returns:

true if media asset is open

getFilePointer

```
public long getFilePointer()
```

Get the current byte location in the media asset

Returns:

current byte location in the media asset

seek

```
public void seek(long pos)
```

Seek to a position in the media asset

Parameters:

pos - position to seek to

read

```
public int read(byte[] buf,  
               int off,  
               int size)
```

Read bytes from the media asset

Parameters:

buf - buffer to fill

off - offset in buffer

size - size of block to read

Returns:

number of bytes read, -1 if failure

getDirecton

```
public int getDirecton()
```

(continued on next page)

(continued from last page)

Get the current direction hint for the random access reader. The underlying system will call `setDirection` to hint at the current direction the media assets is being read.

Returns:

current direction hint for the random access reader

setDirecton

```
public void setDirecton(int directon)
```

Set the current direction hint

Parameters:

`directon` - current direction hint for the random access reader

getBasePath

```
public String getBasePath()
```

Get the `basePath` for the random access reader

Returns:

`basePath` for the random access reader

getMediaName

```
public String getMediaName()
```

Get the media name

Returns:

media name

getMediaExtension

```
public String getMediaExtension()
```

Get the media extension

Returns:

media extension

exists

```
public boolean exists()
```

Does the media asset exist

Returns:

true if media assets exists

lastModified

```
public long lastModified()
```

Return the `lastModified` date (same format as `File.lastModified`)

Returns:

`lastModified` date (same format as `File.lastModified`)

length

```
public long length()
```

Get the media asset length in bytes

Returns:

media asset length in bytes

getPath

```
public String getPath()
```

Get the unique path to the media asset item

Returns:

unique path to the media asset item

Package

com.wowza.util

com.wowza.util Class AMFUtils

java.lang.Object

└─com.wowza.util.AMFUtils

public class **AMFUtils**
extends Object

Utilities for the conversion between Java and AMF

Constructor Summary

public	AMFUtils()
--------	----------------------------

Method Summary

static AMFData[]	convertParams (Object[] params) Converts an array of Java native data values and class to AMF data types.
----------------------------------	--

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

AMFUtils

public **AMFUtils**()

Methods

convertParams

public static [AMFData\[\]](#) **convertParams**(Object[] params)

(continued from last page)

Converts an array of Java native data values and class to AMF data types. It will box primitive data types into wrapper classes. Supported input data types are:

- null
- Double
- Float
- Long
- Integer
- Short
- Byte
- Number
- Date
- Boolean
- String
- StringBuffer
- byte[] - assumed to be raw AMFData
- ByteBuffer - assumed to be raw AMFData
- AMFData
- AMFDataMixedArray
- AMFDataArray
- AMFDataItem
- AMFDataList
- AMFDataObj

Parameters:

params - Array of Java

Returns:

Array of AMFData[] objects

com.wowza.util Class Base64

```
java.lang.Object
|
+-com.wowza.util.Base64
```

public class **Base64**
extends Object

Encodes and decodes to and from Base64 notation.

Homepage: <http://iharder.net/base64>.

Change Log:

- v2.2.1 - Fixed bug using URL_SAFE and ORDERED encodings. Fixed bug when using very small files (~< 40 bytes).
- v2.2 - Added some helper methods for encoding/decoding directly from one file to the next. Also added a main() method to support command line encoding/decoding from one file to the next. Also added these Base64 dialects:
 1. The default is RFC3548 format.
 2. Calling Base64.setFormat(Base64.BASE64_FORMAT.URLSAFE_FORMAT) generates URL and file name friendly format as described in Section 4 of RFC3548. <http://www.faqs.org/rfcs/rfc3548.html>
 3. Calling Base64.setFormat(Base64.BASE64_FORMAT.ORDERED_FORMAT) generates URL and file name friendly format that preserves lexical ordering as described in <http://www.faqs.org/qa/rfcc-1940.html>
 Special thanks to Jim Kellerman at <http://www.powerset.com/> for contributing the new Base64 dialects.
- v2.1 - Cleaned up javadoc comments and unused variables and methods. Added some convenience methods for reading and writing to and from files.
- v2.0.2 - Now specifies UTF-8 encoding in places where the code fails on systems with other encodings (like EBCDIC).
- v2.0.1 - Fixed an error when decoding a single byte, that is, when the encoded data was a single byte.
- v2.0 - I got rid of methods that used booleans to set options. Now everything is more consolidated and cleaner. The code now detects when data that's being decoded is gzip-compressed and will decompress it automatically. Generally things are cleaner. You'll probably have to change some method calls that you were making to support the new options format (ints that you "OR" together).
- v1.5.1 - Fixed bug when decompressing and decoding to a byte[] using decode(String s, boolean gzipCompressed). Added the ability to "suspend" encoding in the Output Stream so you can turn on and off the encoding if you need to embed base64 data in an otherwise "normal" stream (like an XML file).
- v1.5 - Output stream pases on flush() command but doesn't do anything itself. This helps when using GZIP streams. Added the ability to GZip-compress objects before encoding them.
- v1.4 - Added helper methods to read/write files.
- v1.3.6 - Fixed OutputStream.flush() so that 'position' is reset.
- v1.3.5 - Added flag to turn on and off line breaks. Fixed bug in input stream where last buffer being read, if not completely full, was not returned.
- v1.3.4 - Fixed when "improperly padded stream" error was thrown at the wrong time.
- v1.3.3 - Fixed I/O streams which were totally messed up.

I am placing this code in the Public Domain. Do with it as you will. This software comes with no guarantees or warranties but with plenty of well-wishing instead! Please visit <http://iharder.net/base64> periodically to check for updates or to contribute improvements.

Nested Class Summary

class	Base64.InputStream Base64.InputStream
class	Base64.OutputStream Base64.OutputStream

Field Summary

<code>public static final</code>	<u>DECODE</u> Specify decoding. Value: 0
<code>public static final</code>	<u>DONT_BREAK_LINES</u> Don't break lines when encoding (violates strict Base64 specification) Value: 8
<code>public static final</code>	<u>ENCODE</u> Specify encoding. Value: 1
<code>public static final</code>	<u>GZIP</u> Specify that data should be gzip-compressed. Value: 2
<code>public static final</code>	<u>NO_OPTIONS</u> No options specified. Value: 0
<code>public static final</code>	<u>ORDERED</u> Encode using the special "ordered" dialect of Base64 described here: http://www.faqs.org/qa/rfcc-1940.html . Value: 32
<code>public static final</code>	<u>URL_SAFE</u> Encode using Base64-like encoding that is URL- and Filename-safe as described in Section 4 of RFC3548: http://www.faqs.org/rfcs/rfc3548.html . Value: 16

Method Summary

<code>static byte[]</code>	<u>decode</u> (byte[] source, int off, int len, int options) Very low-level access to decoding ASCII characters in the form of a byte array.
<code>static byte[]</code>	<u>decode</u> (String s) Decodes data from Base64 notation, automatically detecting gzip-compressed data and decompressing it.
<code>static byte[]</code>	<u>decode</u> (String s, int options) Decodes data from Base64 notation, automatically detecting gzip-compressed data and decompressing it.
<code>static void</code>	<u>decodeFileToFile</u> (String infile, String outfile) Reads infile and decodes it to outfile.
<code>static byte[]</code>	<u>decodeFromFile</u> (String filename) Convenience method for reading a base64-encoded file and decoding it.
<code>static void</code>	<u>decodeString</u> (String inString) Decodes inString and outputs result to stdout
<code>static boolean</code>	<u>decodeToFile</u> (String dataToDecode, String filename) Convenience method for decoding data to a file.

static Object	<code>decodeToObject</code> (String encodedObject) Attempts to decode Base64 data and deserialize a Java Object within.
static String	<code>encodeBytes</code> (byte[] source) Encodes a byte array into Base64 notation.
static String	<code>encodeBytes</code> (byte[] source, int options) Encodes a byte array into Base64 notation.
static String	<code>encodeBytes</code> (byte[] source, int off, int len) Encodes a byte array into Base64 notation.
static String	<code>encodeBytes</code> (byte[] source, int off, int len, int options) Encodes a byte array into Base64 notation.
static void	<code>encodeFileToFile</code> (String infile, String outfile) Reads infile and encodes it to outfile.
static String	<code>encodeFromFile</code> (String filename) Convenience method for reading a binary file and base64-encoding it.
static String	<code>encodeObject</code> (java.io.Serializable serializableObject) Serializes an object and returns the Base64-encoded version of that serialized object.
static String	<code>encodeObject</code> (java.io.Serializable serializableObject, int options) Serializes an object and returns the Base64-encoded version of that serialized object.
static void	<code>encodeString</code> (String inString) Encodes inString and outputs result to stdout
static boolean	<code>encodeToFile</code> (byte[] dataToEncode, String filename) Convenience method for encoding data to a file.
static void	<code>main</code> (String[] args) Encodes or decodes two files from the command line; feel free to delete this method (in fact you probably should) if you're embedding this code into a larger program.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

NO_OPTIONS

public static final int **NO_OPTIONS**

No options specified. Value is zero.
Constant value: **0**

ENCODE

public static final int **ENCODE**

Specify encoding.
Constant value: **1**

DECODE

```
public static final int DECODE
```

Specify decoding.
Constant value: **0**

GZIP

```
public static final int GZIP
```

Specify that data should be gzip-compressed.
Constant value: **2**

DONT_BREAK_LINES

```
public static final int DONT_BREAK_LINES
```

Don't break lines when encoding (violates strict Base64 specification)
Constant value: **8**

URL_SAFE

```
public static final int URL_SAFE
```

Encode using Base64-like encoding that is URL- and Filename-safe as described in Section 4 of RFC3548: <http://www.faqs.org/rfcs/rfc3548.html>. It is important to note that data encoded this way is *not* officially valid Base64, or at the very least should not be called Base64 without also specifying that it was encoded using the URL- and Filename-safe dialect.
Constant value: **16**

ORDERED

```
public static final int ORDERED
```

Encode using the special "ordered" dialect of Base64 described here: <http://www.faqs.org/qa/rfc-1940.html>.
Constant value: **32**

Methods

main

```
public final static void main(String[] args)
```

Encodes or decodes two files from the command line; **feel free to delete this method (in fact you probably should) if you're embedding this code into a larger program.**

encodeObject

```
public static String encodeObject(java.io.Serializable serializableObject)
```

Serializes an object and returns the Base64-encoded version of that serialized object. If the object cannot be serialized or there is another error, the method will return null. The object is not GZip-compressed before being encoded.

Parameters:

serializableObject - The object to encode

Returns:

The Base64-encoded object

encodeObject

```
public static String encodeObject(java.io.Serializable serializableObject,  
    int options)
```

Serializes an object and returns the Base64-encoded version of that serialized object. If the object cannot be serialized or there is another error, the method will return null.

Valid options:

```
GZIP: gzip-compresses object before encoding it.  
DONT_BREAK_LINES: don't break lines at 76 characters  
    Note: Technically, this makes your encoding non-compliant.
```

Example: `encodeObject(myObj, Base64.GZIP)` or

Example: `encodeObject(myObj, Base64.GZIP | Base64.DONT_BREAK_LINES)`

Parameters:

`serializableObject` - The object to encode
`options` - Specified options

Returns:

The Base64-encoded object

See Also:

[GZIP](#)

[DONT_BREAK_LINES](#)

encodeBytes

```
public static String encodeBytes(byte[] source)
```

Encodes a byte array into Base64 notation. Does not GZip-compress data.

Parameters:

`source` - The data to convert

encodeBytes

```
public static String encodeBytes(byte[] source,  
    int options)
```

(continued from last page)

Encodes a byte array into Base64 notation.

Valid options:

GZIP: gzip-compresses object before encoding it.
DONT_BREAK_LINES: don't break lines at 76 characters
Note: Technically, this makes your encoding non-compliant.

Example: `encodeBytes(myData, Base64.GZIP)` or

Example: `encodeBytes(myData, Base64.GZIP | Base64.DONT_BREAK_LINES)`

Parameters:

source - The data to convert
options - Specified options

See Also:

[GZIP](#)
[DONT_BREAK_LINES](#)

encodeBytes

```
public static String encodeBytes(byte[] source,  
    int off,  
    int len)
```

Encodes a byte array into Base64 notation. Does not GZip-compress data.

Parameters:

source - The data to convert
off - Offset in array where conversion should begin
len - Length of data to convert

encodeBytes

```
public static String encodeBytes(byte[] source,  
    int off,  
    int len,  
    int options)
```

Encodes a byte array into Base64 notation.

Valid options:

GZIP: gzip-compresses object before encoding it.
DONT_BREAK_LINES: don't break lines at 76 characters
Note: Technically, this makes your encoding non-compliant.

Example: `encodeBytes(myData, Base64.GZIP)` or

Example: `encodeBytes(myData, Base64.GZIP | Base64.DONT_BREAK_LINES)`

(continued from last page)

Parameters:

- source - The data to convert
- off - Offset in array where conversion should begin
- len - Length of data to convert
- options - Specified options, alphabet type is pulled from this (standard, url-safe, ordered)

See Also:

[GZIP](#)
[DONT_BREAK_LINES](#)

decode

```
public static byte[] decode(byte[] source,  
    int off,  
    int len,  
    int options)
```

Very low-level access to decoding ASCII characters in the form of a byte array. Does not support automatically gunzipping or any other "fancy" features.

Parameters:

- source - The Base64 encoded data
- off - The offset of where to begin decoding
- len - The length of characters to decode

Returns:

decoded data

decode

```
public static byte[] decode(String s)
```

Decodes data from Base64 notation, automatically detecting gzip-compressed data and decompressing it.

Parameters:

- s - the string to decode

Returns:

the decoded data

decode

```
public static byte[] decode(String s,  
    int options)
```

Decodes data from Base64 notation, automatically detecting gzip-compressed data and decompressing it.

Parameters:

- s - the string to decode
- options - encode options such as URL_SAFE

Returns:

the decoded data

decodeToObject

```
public static Object decodeToObject(String encodedObject)
```

(continued from last page)

Attempts to decode Base64 data and deserialize a Java Object within. Returns null if there was an error.

Parameters:

encodedObject - The Base64 data to decode

Returns:

The decoded and deserialized object

encodeToFile

```
public static boolean encodeToFile(byte[] dataToEncode,  
    String filename)
```

Convenience method for encoding data to a file.

Parameters:

dataToEncode - byte array of data to encode in base64 form

filename - Filename for saving encoded data

Returns:

true if successful, false otherwise

decodeToFile

```
public static boolean decodeToFile(String dataToDecode,  
    String filename)
```

Convenience method for decoding data to a file.

Parameters:

dataToDecode - Base64-encoded data as a string

filename - Filename for saving decoded data

Returns:

true if successful, false otherwise

decodeFromFile

```
public static byte[] decodeFromFile(String filename)
```

Convenience method for reading a base64-encoded file and decoding it.

Parameters:

filename - Filename for reading encoded data

Returns:

decoded byte array or null if unsuccessful

encodeFromFile

```
public static String encodeFromFile(String filename)
```

Convenience method for reading a binary file and base64-encoding it.

Parameters:

filename - Filename for reading binary data

Returns:

base64-encoded string or null if unsuccessful

encodeFileToFile

```
public static void encodeFileToFile(String infile,  
                                     String outfile)
```

Reads infile and encodes it to outfile.

Parameters:

infile - Input file

outfile - Output file

encodeString

```
public static void encodeString(String inString)
```

Encodes inString and outputs result to stdout

Parameters:

inString - Input string

decodeString

```
public static void decodeString(String inString)
```

Decodes inString and outputs result to stdout

Parameters:

inString - Input string

decodeFileToFile

```
public static void decodeFileToFile(String infile,  
                                     String outfile)
```

Reads infile and decodes it to outfile.

Parameters:

infile - Input file

outfile - Output file

com.wowza.util Class Base64.InputStream

```

java.lang.Object
  |
  +- java.io.InputStream
        |
        +- java.io.FilterInputStream
              |
              +- com.wowza.util.Base64.InputStream
  
```

All Implemented Interfaces:

java.io.Closeable

public static class **Base64.InputStream**
extends java.io.FilterInputStream

A [Base64.InputStream](#) will read data from another java.io.InputStream, given in the constructor, and encode/decode to/from Base64 notation on the fly.

See Also:

[Base64](#)

Fields inherited from class java.io.FilterInputStream

in

Constructor Summary

public	Base64.InputStream (java.io.InputStream in) Constructs a Base64.InputStream in DECODE mode.
public	Base64.InputStream (java.io.InputStream in, int options) Constructs a Base64.InputStream in either ENCODE or DECODE mode.

Method Summary

int	read () Reads enough of the input stream to convert to/from Base64 and returns the next byte.
int	read (byte[] dest, int off, int len) Calls read() repeatedly until the end of stream is reached or len bytes are read.

Methods inherited from class java.io.FilterInputStream

available, close, mark, markSupported, read, read, read, reset, skip

Methods inherited from class java.io.InputStream

available, close, mark, markSupported, read, read, read, reset, skip

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface java.io.Closeable

```
close
```

Constructors

Base64.InputStream

```
public Base64.InputStream( java.io.InputStream in)
```

Constructs a [Base64.InputStream](#) in DECODE mode.

Parameters:

in - the java.io.InputStream from which to read data.

Base64.InputStream

```
public Base64.InputStream( java.io.InputStream in,  
                           int options)
```

Constructs a [Base64.InputStream](#) in either ENCODE or DECODE mode.

Valid options:

```
ENCODE or DECODE: Encode or Decode as data is read.  
DONT_BREAK_LINES: don't break lines at 76 characters  
    (only meaningful when encoding)  
Note: Technically, this makes your encoding non-compliant.
```

Example: new Base64.InputStream(in, Base64.DECODE)

Parameters:

in - the java.io.InputStream from which to read data.

options - Specified options

See Also:

[Base64.ENCODE](#)

[Base64.DECODE](#)

[Base64.DONT_BREAK_LINES](#)

Methods

read

```
public int read()  
    throws java.io.IOException
```

Reads enough of the input stream to convert to/from Base64 and returns the next byte.

Returns:

next byte

(continued from last page)

read

```
public int read(byte[] dest,  
               int off,  
               int len)  
throws java.io.IOException
```

Calls [read\(\)](#) repeatedly until the end of stream is reached or len bytes are read. Returns number of bytes read into array or -1 if end of stream is encountered.

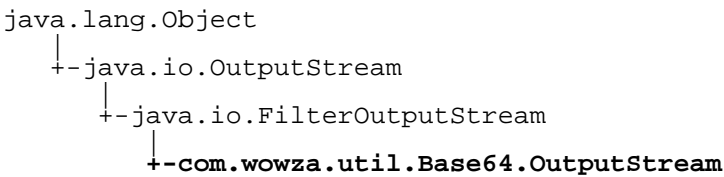
Parameters:

dest - array to hold values
off - offset for array
len - max number of bytes to read into array

Returns:

bytes read into array or -1 if end of stream is encountered.

com.wowza.util
Class Base64.OutputStream



All Implemented Interfaces:
java.io.Flushable, java.io.Closeable

public static class **Base64.OutputStream**
extends java.io.FilterOutputStream

A [Base64.OutputStream](#) will write data to another java.io.OutputStream, given in the constructor, and encode/decode to/from Base64 notation on the fly.
See Also: [Base64](#)

Fields inherited from class java.io.FilterOutputStream	
	out

Constructor Summary	
public	Base64.OutputStream (java.io.OutputStream out) Constructs a Base64.OutputStream in ENCODE mode.
public	Base64.OutputStream (java.io.OutputStream out, int options) Constructs a Base64.OutputStream in either ENCODE or DECODE mode.

Method Summary	
void	close () Flushes and closes (I think, in the superclass) the stream.
void	flushBase64 () Method added by PHIL.
void	resumeEncoding () Resumes encoding of the stream.
void	suspendEncoding () Suspends encoding of the stream.
void	write (byte[] theBytes, int off, int len) Calls write(int) repeatedly until len bytes are written.
void	write (int theByte) Writes the byte to the output stream after converting to/from Base64 notation.

Methods inherited from class java.io.FilterOutputStream

```
close, flush, write, write, write
```

Methods inherited from class `java.io.OutputStream`

```
close, flush, write, write, write
```

Methods inherited from class `java.lang.Object`

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

Methods inherited from interface `java.io.Closeable`

```
close
```

Methods inherited from interface `java.io.Flushable`

```
flush
```

Constructors

Base64.OutputStream

```
public Base64.OutputStream(java.io.OutputStream out)
```

Constructs a [Base64.OutputStream](#) in ENCODE mode.

Parameters:

out - the java.io.OutputStream to which data will be written.

Base64.OutputStream

```
public Base64.OutputStream(java.io.OutputStream out,  
                           int options)
```

Constructs a [Base64.OutputStream](#) in either ENCODE or DECODE mode.

Valid options:

```
ENCODE or DECODE: Encode or Decode as data is read.  
DONT_BREAK_LINES: don't break lines at 76 characters  
    (only meaningful when encoding)  
Note: Technically, this makes your encoding non-compliant.
```

Example: new Base64.OutputStream(out, Base64.ENCODE)

Parameters:

out - the java.io.OutputStream to which data will be written.

options - Specified options.

(continued from last page)

See Also:[Base64.ENCODER](#)[Base64.DECODER](#)[Base64.DONT_BREAK_LINES](#)

Methods

write

```
public void write(int theByte)
    throws java.io.IOException
```

Writes the byte to the output stream after converting to/from Base64 notation. When encoding, bytes are buffered three at a time before the output stream actually gets a `write()` call. When decoding, bytes are buffered four at a time.

Parameters:

`theByte` - the byte to write

write

```
public void write(byte[] theBytes,
    int off,
    int len)
    throws java.io.IOException
```

Calls [write\(int\)](#) repeatedly until `len` bytes are written.

Parameters:

`theBytes` - array from which to read bytes

`off` - offset for array

`len` - max number of bytes to read into array

flushBase64

```
public void flushBase64()
    throws java.io.IOException
```

Method added by PHIL. [Thanks, PHIL. -Rob] This pads the buffer without closing the stream.

close

```
public void close()
    throws java.io.IOException
```

Flushes and closes (I think, in the superclass) the stream.

suspendEncoding

```
public void suspendEncoding()
    throws java.io.IOException
```

Suspends encoding of the stream. May be helpful if you need to embed a piece of base64-encoded data in a stream.

resumeEncoding

```
public void resumeEncoding()
```

Resumes encoding of the stream. May be helpful if you need to embed a piece of base64-encoded data in a stream.

com.wowza.util Class BufferUtils

java.lang.Object

└─com.wowza.util.BufferUtils

public class **BufferUtils**
extends Object

BufferUtils: utilities for converting between binary data and Java primitive types. Faster than Java runtime equivalents

Field Summary

public static final	alphas
public static final	hexadecimal

Constructor Summary

public	BufferUtils()
--------	-------------------------------

Method Summary

static int	byteArrayToInt (byte[] b) Convert byte array to int
static int	byteArrayToInt (byte[] b, int offset) Convert byte array to int with offset
static int	byteArrayToInt (byte[] b, int offset, int count) Convert byte array to int with offset.
static int	byteArrayToInt (byte[] b, int offset, int count, boolean isReverse) Convert byte array to int with offset.
static long	byteArrayToLong (byte[] b) Convert byte array to long
static long	byteArrayToLong (byte[] b, int offset) Convert byte array to long with offset
static long	byteArrayToLong (byte[] b, int offset, int count) Convert byte array to long with offset.
static long	byteArrayToLong (byte[] b, int offset, int count, boolean isReverse) Convert byte array to long with offset.
static int	byteArrayToShort (byte[] b) Convert byte array to int

static int	<code>byteArrayToShort</code> (byte[] b, int offset) Convert byte array to int with offset
static int	<code>byteArrayToShort</code> (byte[] b, int offset, int count) Convert byte array to int with offset.
static int	<code>byteArrayToShort</code> (byte[] b, int offset, int count, boolean isReverse) Convert byte array to int with offset.
static String	<code>byteArrayToString</code> (byte[] b) Convert a byte array to a String (UTF-8 encoding assumed)
static String	<code>byteArrayToString</code> (byte[] b, int offset, int count) Convert a byte array to a String (UTF-8 encoding assumed)
static String	<code>byteArrayToString</code> (byte[] b, int offset, int count, String charsetName) Convert a byte array to a String
static byte[]	<code>decodeHexString</code> (String hexStr) Decode a string as a byte array
static int	<code>doCRC32</code> (int crc, byte[] buffer, int offset, int len) Calculate an IEEE CRC32 value for MPEG transport stream from a starting crc value
static String	<code>encodeHexString</code> (byte[] bytes) Encode a byte array as a string
static String	<code>encodeHexString</code> (byte[] bytes, int offset, int len) Encode a byte array as a string
static int	<code>getUnsignedShort</code> (java.nio.ByteBuffer buffer)
static int	<code>indexOf</code> (byte[] source, byte[] pattern) Finds the first occurrence of a byte pattern in a byte buffer.
static int	<code>indexOfDifferent</code> (byte[] buffer1, byte[] buffer2) Compare two byte buffers, and return the index of the first byte that is different.
static byte[]	<code>intToByteArray</code> (int value) Convert a int value to a byte array in network order
static void	<code>intToByteArray</code> (int value, byte[] buffer, int offset, int size) Convert a int value to a byte array in network order
static void	<code>intToByteArray</code> (int value, byte[] buffer, int offset, int size, boolean isReverse) Convert a int value to a byte array in network order
static byte[]	<code>intToByteArray</code> (int value, int size) Convert a int value to a byte array in network order
static byte[]	<code>longToByteArray</code> (long value) Convert a long value to a byte array in network order
static void	<code>longToByteArray</code> (long value, byte[] buffer, int offset, int size) Convert a long value to a byte array in network order
static void	<code>longToByteArray</code> (long value, byte[] buffer, int offset, int size, boolean isReverse) Convert a long value to a byte array in network order

static byte[]	<code>longToByteArray</code> (long value, int size) Convert a long value to a byte array in network order
static boolean	<code>startsWith</code> (byte[] source, byte[] pattern)

Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Fields

hexadecimal

public static final java.lang.String **hexadecimal**

alphas

public static final java.lang.String **alphas**

Constructors

BufferUtils

public **BufferUtils**()

Methods

encodeHexString

public static String **encodeHexString**(byte[] bytes)

Encode a byte array as a string

Parameters:

bytes - byte array

Returns:

string

encodeHexString

public static String **encodeHexString**(byte[] bytes,
int offset,
int len)

Encode a byte array as a string

Parameters:

(continued from last page)

bytes - byte array
offset - offset
len - length

Returns:
string

decodeHexString

```
public static byte[] decodeHexString(String hexStr)
```

Decode a string as a byte array

Parameters:
hexStr - string

Returns:
byte array

byteArrayToString

```
public static String byteArrayToString(byte[] b)
```

Convert a byte array to a String (UTF-8 encoding assumed)

Parameters:
b - byte array

Returns:
resultant string

byteArrayToString

```
public static String byteArrayToString(byte[] b,  
    int offset,  
    int count)
```

Convert a byte array to a String (UTF-8 encoding assumed)

Parameters:
b - byte array
offset - offset
count - len

Returns:
resultant string

byteArrayToString

```
public static String byteArrayToString(byte[] b,  
    int offset,  
    int count,  
    String charsetName)
```

Convert a byte array to a String

Parameters:
b - byte array
offset - offset

(continued from last page)

count - len
charsetName - character set name

Returns:

resultant string

byteArrayToLong

```
public static long byteArrayToLong(byte[] b)
```

Convert byte array to long

Parameters:

b - byte array (8 bytes)

Returns:

long value

byteArrayToLong

```
public static long byteArrayToLong(byte[] b,  
    int offset)
```

Conver byte array to long with offset

Parameters:

b - byte array (8 bytes)
offset - offset

Returns:

long value

byteArrayToLong

```
public static long byteArrayToLong(byte[] b,  
    int offset,  
    int count)
```

Convert byte array to long with offset. Count is the number of bytes. Can be less than 8. If less than 8 just fills lower bits in value.

Parameters:

b - byte array
offset - offset
count - number of bytes

Returns:

long value

byteArrayToLong

```
public static long byteArrayToLong(byte[] b,  
    int offset,  
    int count,  
    boolean isReverse)
```

Convert byte array to long with offset. Count is the number of bytes. Can be less than 8. If less than 8 just fills lower bits in value. The reverse flag allows data to be in reverse order.

Parameters:

(continued from last page)

b - byte array
offset - offset
count - number of bytes
isReverse - is data in reverse order

Returns:

long value

byteArrayToInt

```
public static int byteArrayToInt(byte[] b)
```

Convert byte array to int

Parameters:

b - byte array (4 bytes)

Returns:

int value

byteArrayToInt

```
public static int byteArrayToInt(byte[] b,  
    int offset)
```

Convert byte array to int with offset

Parameters:

b - byte array (4 bytes)
offset - offset

Returns:

int value

byteArrayToInt

```
public static int byteArrayToInt(byte[] b,  
    int offset,  
    int count)
```

Convert byte array to int with offset. Count is the number of bytes. Can be less than 4. If less than 4 just fills lower bits in value.

Parameters:

b - byte array
offset - offset
count - number of bytes

Returns:

int value

byteArrayToInt

```
public static int byteArrayToInt(byte[] b,  
    int offset,  
    int count,  
    boolean isReverse)
```

Convert byte array to int with offset. Count is the number of bytes. Can be less than 4. If less than 4 just fills lower bits in value. The reverse flag allows data to be in reverse order.

(continued from last page)

Parameters:

b - byte array
offset - offset
count - number of bytes
isReverse - is data in reverse order

Returns:

int value

byteArrayToShort

```
public static int byteArrayToShort(byte[] b)
```

Convert byte array to int

Parameters:

b - byte array (2 bytes)

Returns:

int value

byteArrayToShort

```
public static int byteArrayToShort(byte[] b,  
    int offset)
```

Conver byte array to int with offset

Parameters:

b - byte array (2 bytes)
offset - offset

Returns:

int value

byteArrayToShort

```
public static int byteArrayToShort(byte[] b,  
    int offset,  
    int count)
```

Convert byte array to int with offset. Count is the number of bytes. Can be less than 2. If less than 2 just fills lower bits in value.

Parameters:

b - byte array
offset - offset
count - number of bytes

Returns:

int value

byteArrayToShort

```
public static int byteArrayToShort(byte[] b,  
    int offset,  
    int count,  
    boolean isReverse)
```

Convert byte array to int with offset. Count is the number of bytes. Can be less than 2. If less than 2 just fills lower bits in value. The reverse flag allows data to be in reverse order.

(continued from last page)

Parameters:

b - byte array
offset - offset
count - number of bytes
isReverse - is data in reverse order

Returns:

int value

intToByteArray

```
public static byte[] intToByteArray(int value)
```

Convert a int value to a byte array in network order

Parameters:

value - value

Returns:

4-byte array with value

intToByteArray

```
public static byte[] intToByteArray(int value,  
                                     int size)
```

Convert a int value to a byte array in network order

Parameters:

value - value
size - size of resultant byte array

Returns:

size-byte array with value

intToByteArray

```
public static void intToByteArray(int value,  
                                   byte[] buffer,  
                                   int offset,  
                                   int size)
```

Convert a int value to a byte array in network order

Parameters:

value - value
buffer - destination byte array
offset - starting offset in byte array
size - number of bytes to write

intToByteArray

```
public static void intToByteArray(int value,  
                                   byte[] buffer,  
                                   int offset,  
                                   int size,  
                                   boolean isReverse)
```

Convert a int value to a byte array in network order

(continued from last page)

Parameters:

value - value
buffer - destination byte array
offset - starting offset in byte array
size - number of bytes to write
isReverse - is data in reverse order

longToByteArray

```
public static byte[] longToByteArray(long value)
```

Convert a long value to a byte array in network order

Parameters:

value - value

Returns:

8-byte array with value

longToByteArray

```
public static byte[] longToByteArray(long value,  
int size)
```

Convert a long value to a byte array in network order

Parameters:

value - value
size - size of resultant byte array

Returns:

size-byte array with value

longToByteArray

```
public static void longToByteArray(long value,  
byte[] buffer,  
int offset,  
int size)
```

Convert a long value to a byte array in network order

Parameters:

value - value
buffer - destination byte array
offset - starting offset in byte array
size - number of bytes to write

longToByteArray

```
public static void longToByteArray(long value,  
byte[] buffer,  
int offset,  
int size,  
boolean isReverse)
```

Convert a long value to a byte array in network order

Parameters:

(continued from last page)

value - value
buffer - destination byte array
offset - starting offset in byte array
size - number of bytes to write
isReverse - is data in reverse order

getUnsignedShort

```
public static int getUnsignedShort(java.nio.ByteBuffer buffer)
```

doCRC32

```
public static int doCRC32(int crc,  
    byte[] buffer,  
    int offset,  
    int len)
```

Calculate an IEEE CRC32 value for MPEG transport stream from a starting crc value

Parameters:

crc - starting crc value
buffer - buffer
offset - offset
len - len

Returns:

crc value

indexOf

```
public static int indexOf(byte[] source,  
    byte[] pattern)
```

Finds the first occurrence of a byte pattern in a byte buffer.

Note: If we want to improve the performance, we could implement the Knuth-Morris-Pratt algorithm. But I don't need the speed nor do I have time today!

Parameters:

source - The source byte buffer
pattern - the pattern we're looking for.

Returns:

the index of source where the pattern is found, else -1

indexOfDifferent

```
public static int indexOfDifferent(byte[] buffer1,  
    byte[] buffer2)
```

Compare two byte buffers, and return the index of the first byte that is different.

Parameters:

buffer1 - byte array1
buffer2 - byte array2

Returns:

index where buffers differ, or -1.

startsWith

```
public static boolean startsWith(byte[] source,  
                                byte[] pattern)
```

com.wowza.util Class DebugUtils

java.lang.Object

└─com.wowza.util.DebugUtils

public final class **DebugUtils**
extends Object

DebugUtils: Debugging utilities.

Constructor Summary

public	DebugUtils()
--------	------------------------------

Method Summary

static java.io.File	byteArrayToFile (byte[] data, String filePath) Dump a buffer of bytes to a file.
static boolean	doesStackContainMethod (String methodName) Examine current stack trace to determine if the specified method name in the the trace.
static String	formatBytes (byte[] data) Format byte array for printing.
static String	formatBytes (byte[] data, boolean showIndex) Format byte array for printing.
static String	formatBytes (byte[] data, int offset, int len) Format byte array for printing.
static String	formatBytes (byte[] data, int offset, int len, boolean showIndex) Format byte array for printing.
static String	formatBytesInSingleRow (byte[] data, int offset, int len) Format byte array for printing.
static String	formatBytesShort (byte[] data) Format byte array for printing.
static String	formatBytesShort (byte[] data, int offset, int len) Format byte array for printing.
static String	formatBytesStruct (byte[] data)
static String	formatBytesStruct (byte[] data, int offset, int len) Format byte array for printing.
static String	formatMilliseconds (long ms) Give a time in seconds, return a String representing the time in hh:mm:ss.mmm.

static String	<code>formatSeconds</code> (long t) Give a time in seconds, return a String representing the time in hh:mm:ss.
static String	<code>formatUtcTime</code> (long utcTime)
static String	<code>stackTraceToSingleLineString</code> (Throwable e, int maxLines) Given an exception, return single line String representation.
static String	<code>stackTraceToString</code> (Throwable e) Given an exception, return String representation.
static String	<code>stackTraceToString</code> (Throwable e, int maxLines, String separator) Given an exception, return single line String representation.
static String	<code>toHex</code> (byte value) Format a byte value to a 0xff format
static String	<code>toHex</code> (int value) Format a byte value to a 0xffffffff format
static String	<code>toLong</code> (long value) Format a long value to a 0xffffffffffffffff format

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

DebugUtils

```
public DebugUtils()
```

Methods

formatBytesShort

```
public static String formatBytesShort(byte[] data)
```

Format byte array for printing. Simple format as rows of hex values (16 values per row)

Parameters:

data - byte array

Returns:

formatted string

formatBytesShort

```
public static String formatBytesShort(byte[] data,  
    int offset,  
    int len)
```

(continued from last page)

Format byte array for printing. Simple format as rows of hex values (16 values per row)

Parameters:

data - byte array
offset - offset in array
len - data len

Returns:

formatted string

formatBytes

```
public static String formatBytes(byte[] data)
```

Format byte array for printing. Format as rows of hex values (16 values per row) along with text representation of data.

Parameters:

data - byte array

Returns:

formatted string

formatBytes

```
public static String formatBytes(byte[] data,  
    boolean showIndex)
```

Format byte array for printing. Format as rows of hex values (16 values per row) along with text representation of data.

Parameters:

data - byte array

Returns:

formatted string

formatBytes

```
public static String formatBytes(byte[] data,  
    int offset,  
    int len)
```

Format byte array for printing. Format as rows of hex values (16 values per row) along with text representation of data.

Parameters:

data - byte array
offset - start index in array
len - length to format

Returns:

formatted string

formatBytes

```
public static String formatBytes(byte[] data,  
    int offset,  
    int len,  
    boolean showIndex)
```

Format byte array for printing. Format as rows of hex values (16 values per row) along with text representation of data.

(continued from last page)

Parameters:

data - byte array
offset - start index in array
len - length to format

Returns:

formatted string

formatBytesInSingleRow

```
public static String formatBytesInSingleRow(byte[] data,  
    int offset,  
    int len)
```

Format byte array for printing. Format as rows of hex values all in one row along with text representation of data.

Parameters:

data - byte array
offset - start index in array
len - length to format

Returns:

formatted string

formatBytesStruct

```
public static String formatBytesStruct(byte[] data)
```

formatBytesStruct

```
public static String formatBytesStruct(byte[] data,  
    int offset,  
    int len)
```

Format byte array for printing. Format as Java primitive byte values (8 per row).

Parameters:

data - byte array

Returns:

formatted string

toHex

```
public static String toHex(byte value)
```

Format a byte value to a 0xff format

Parameters:

value - byte value

Returns:

return string

toHex

```
public static String toHex(int value)
```


(continued from last page)

Format a byte value to a 0xffffffff format

Parameters:

value - int value

Returns:

return string

toLong

```
public static String toLong(long value)
```

Format a long value to a 0xffffffffffffffff format

Parameters:

value - long value

Returns:

return string

byteArrayToFile

```
public static java.io.File byteArrayToFile(byte[] data,  
String filePath)
```

Dump a buffer of bytes to a file. Useful for debugging.

Parameters:

data - a buffer of bytes

filePath - The path and filename

formatSeconds

```
public static String formatSeconds(long t)
```

Give a time in seconds, return a String representing the time in hh:mm:ss.

123 s --> 00:02:03

Parameters:

t

Returns:

time representation

formatMilliseconds

```
public static String formatMilliseconds(long ms)
```

Give a time in seconds, return a String representing the time in hh:mm:ss.mmm.

123456 ms --> 00:02:03.456

Parameters:

ms

Returns:

time representation

formatUtcTime

```
public static String formatUtcTime(long utcTime)
```

stackTraceToString

```
public static String stackTraceToString(Throwable e)
```

Given an exception, return String representation. Note: this is the entire raw stack trace including tabs and line endings

Parameters:

e - throwable

Returns:

stack trace representation

stackTraceToSingleLineString

```
public static String stackTraceToSingleLineString(Throwable e,  
int maxLines)
```

Given an exception, return single line String representation. Note: line endings are replaced with "|" character. Tabs are removed.

Parameters:

e - throwable

maxLines - If greater than zero, returns only first maxLines of trace

Returns:

stack trace info

stackTraceToString

```
public static String stackTraceToString(Throwable e,  
int maxLines,  
String seperator)
```

Given an exception, return single line String representation. Note: line endings are replaced with "|" character. Tabs are removed.

Parameters:

e - throwable

maxLines - If greater than zero, returns only first maxLines of trace

seperator - character to replace line endings with

Returns:

stack trace info

doesStackContainMethod

```
public static boolean doesStackContainMethod(String methodName)
```

Examine current stack trace to determine if the specified method name in the the trace.

Could be enhanced quite a bit by also checking for Class and method. But simple for now.

Parameters:

(continued from last page)

methodName

Returns:

true if in trace, else false

com.wowza.util Class ElapsedTimer

java.lang.Object

└─com.wowza.util.ElapsedTimer

```
public class ElapsedTimer
extends Object
```

ElapsedTimer: Utility class for keep track of the duration an object has been in existence.

Constructor Summary

public	ElapsedTimer() Construct a new ElapsedTimer and start the clock
--------	--

Method Summary

java.util.Date	getDate() Get the date the object was created
String	getDateString() Get the date object was created as formatted String
long	getTime() Get elapsed time object in existence (milliseconds)
double	getTimeSeconds() Get elapsed time object in seconds
String	getTimeString() Get elapsed time object in existence as formatted String (Ex: 3 days 2 minutes 5 seconds)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

ElapsedTimer

```
public ElapsedTimer()
```

Construct a new ElapsedTimer and start the clock

Methods

(continued from last page)

getDate

```
public java.util.Date getDate()
```

Get the date the object was created

Returns:

date object created

getDateString

```
public String getDateString()
```

Get the date object was created as formatted String

Returns:

date object created as formatted String

getTime

```
public long getTime()
```

Get elapsed time object in existence (milliseconds)

Returns:

elapsed time (milliseconds)

getTimeSeconds

```
public double getTimeSeconds()
```

Get elapsed time object in seconds

Returns:

elapsed time in seconds

getTimeString

```
public String getTimeString()
```

Get elapsed time object in existence as formatted String (Ex: 3 days 2 minutes 5 seconds)

Returns:

elapsed time as formatted String

com.wowza.util
Class FileUtils



```
public class FileUtils
extends Object
```

FileUtils: File utilities

Constructor Summary	
public	FileUtils()

Method Summary	
static void	closeQuietly (java.io.InputStream stream)
static void	closeQuietly (java.io.OutputStream stream)
static void	closeQuietly (java.io.Reader reader)
static void	copyFile (java.io.File fromFile, java.io.File toFile) Simple file copy routine
static void	copyFile2 (java.io.File fromFile, java.io.File toFile)
static boolean	deleteDirectory (java.io.File path)
static void	deleteFolder (java.io.File folder)
static byte[]	fileToByteArray (java.io.File file)
static String	streamNameToValidFilename (String name) Encode a stream name (deal with path elements) to a valid filename.
static String	toValidFilename (String name) Encode a name to a valid filename.
static void	traverseDirectory (java.io.File dir, IFileProcess fileNotify) Traverse a directory recursively calling fileNotify for each file and folder encountered
static java.io.File	versionFile (java.io.File newFile) Rename a file using the file format [original-name]_#[.ext].

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

FileUtils

```
public FileUtils()
```

Methods

fileToByteArray

```
public static byte[] fileToByteArray(java.io.File file)
```

streamNameToValidFilename

```
public static String streamNameToValidFilename(String name)
```

Encode a stream name (deal with path elements) to a valid filename.

Parameters:

name - stream name

Returns:

encoded filename

toValidFilename

```
public static String toValidFilename(String name)
```

Encode a name to a valid filename.

Parameters:

name

Returns:

encoded filename

copyFile

```
public static void copyFile(java.io.File fromFile,  
                             java.io.File toFile)
```

Simple file copy routine

Parameters:

fromFile - from file

toFile - to file

(continued from last page)

copyFile2

```
public static void copyFile2(java.io.File fromFile,  
                             java.io.File toFile)  
    throws java.io.IOException
```

closeQuietly

```
public static void closeQuietly(java.io.InputStream stream)
```

closeQuietly

```
public static void closeQuietly(java.io.OutputStream stream)
```

closeQuietly

```
public static void closeQuietly(java.io.Reader reader)
```

deleteDirectory

```
public static boolean deleteDirectory(java.io.File path)
```

versionFile

```
public static java.io.File versionFile(java.io.File newFile)
```

Rename a file using the file format [original-name]_#[.ext].

Parameters:

newFile - input file

Returns:

renamed file

traverseDirectory

```
public static void traverseDirectory(java.io.File dir,  
    IFileProcess fileNotify)
```

Traverse a directory recursively calling fileNotify for each file and folder encountered

Parameters:

dir - starting directory

fileNotify - file notify object

deleteFolder

```
public static void deleteFolder(java.io.File folder)
```

(continued from last page)

com.wowza.util Class FLVUtils

java.lang.Object

└─com.wowza.util.FLVUtils

public final class **FLVUtils**
extends Object

FLVUtils: utility for reading and writing .flv files.

Field Summary

public static final	<u>FLV_CHUNKHEADER_BUFFER_SIZE</u> Size of temporary buffer needed for flv reading (byte[]) Value: 13
public static final	<u>FLV_CHUNKHEADER_FIRST_BYTE</u> Header values: first byte of packet data Value: 3
public static final	<u>FLV_CHUNKHEADER_HEADER_SIZE</u> Size of packet header (byte[]) Value: 11
public static final	<u>FLV_CHUNKHEADER_I_SIZE</u> Header values: packet size Value: 1
public static final	<u>FLV_CHUNKHEADER_I_TIMECODE</u> Header values: timecode (milliseconds) Value: 2
public static final	<u>FLV_CHUNKHEADER_I_TYPE</u> Header values: packet type Value: 0
public static final	<u>FLV_CHUNKHEADER_SECOND_BYTE</u> Header values: second byte of packet data Value: 4
public static final	<u>FLV_CHUNKHEADER_VALUES_SIZE</u> Size of header values array (long[]) Value: 5
public static final	<u>FLV_DFRAME</u> D video frame type (partial frame based on key frame) Value: 3
public static final	<u>FLV_KFRAME</u> Key video frame type Value: 1

public static final	FLV_PFRAME P video frame type (partial frame based on previous frame) Value: 2
public static final	FLV_TCINDEXAUDIO Value: 0
public static final	FLV_TCINDEXDATA Value: 2
public static final	FLV_TCINDEXVIDEO Value: 1
public static final	FLV_UFRAME Unknown video frame type Value: 0

Constructor Summary

public	FLVUtils()
--------	----------------------------

Method Summary

static long	adjustFirstPacketTCs (java.util.List audioTCs, java.util.List videoTCs, java.util.List dataTCs) Align list of timecode for each data type.
static int	audioCodecStringToId (String codecString) Parse a string to get the codec ID defined by IVHost.CODEC_AUDIO_*
static String	audioCodecToMetaDataString (int codec) Get a printable string representation of the audio codecs defined as IVHost.CODEC_AUDIO_* as the string used in the onMetaData event
static String	audioCodecToString (int codec) Get a printable string representation of the audio codecs defined as IVHost.CODEC_AUDIO_*
static String	frameTypeToString (int frameType)
static int	getAudioCodec (AMFPacket packet) Get the codec id for this audio packet.
static int	getAudioCodec (int value) Return the codec portion of the first byte of an audio packet.
static int	getAudioMP3Layer (AMFPacket packet) Audio marked as MP3 is really MPEG1 Layer 1-3.
static int	getFrameType (byte value) Given the first byte of a video packet, determine the frame type (FLV_*FRAME)
static int	getFrameType (int value) Given the first byte of a video packet, determine the frame type (FLV_*FRAME)

static int	<u>getFrameType</u> (int[] values) Given the headers values (including first byte of the packet), determine the type of video frame (FLV_*FRAME)
static byte	<u>getHeaderFlags</u> (java.io.File file)
static long	<u>getLastTC</u> (java.io.File file) Get the duration of an .flv file.
static OnMetadataBasic	<u>getOnMetadataData</u> (<u>AMFPacket</u> metaDataPacket)
static int	<u>getVideoCodec</u> (<u>AMFPacket</u> packet) Get the codec id for this video packet.
static int	<u>getVideoCodec</u> (int value) Return the codec portion of the first byte of an video packet.
static int	<u>getVideoFrameType</u> (<u>AMFPacket</u> packet)
static int	<u>getVideoTimecodeOffset</u> (<u>AMFPacket</u> packet) Get the timecode offset in milliseconds between the PTS and DTS for this frame.
static int	<u>getVideoTimecodeOffset</u> (byte[] buffer) Get the timecode offset in milliseconds between the PTS and DTS for this frame.
static java.util.List	<u>interleavePackets</u> (java.util.List audioPackets, java.util.List videoPackets, java.util.List dataPackets, java.util.List audioTCs, java.util.List videoTCs, java.util.List dataTCs, java.util.List dataTypes, long[] currentTCs) This is a utility function primarily used for IMediaWriters.
static java.util.List	<u>interleavePackets</u> (java.util.List audioPackets, java.util.List videoPackets, java.util.List dataPackets, java.util.List audioTCs, java.util.List videoTCs, java.util.List dataTCs, long[] currentTCs) This is a utility function primarily used for IMediaWriters.
static boolean	<u>isAudioCodecConfig</u> (<u>AMFPacket</u> packet) Returns true if the packet is a video codec config packet
static boolean	<u>isOnMetadataPacket</u> (<u>AMFPacket</u> packet) Returns true if packet is onMetaData or [@setDataFrame, onMetaData] data packet.
static boolean	<u>isVideoCodecConfig</u> (<u>AMFPacket</u> packet) Returns true if the packet is a video codec config packet
static boolean	<u>isVideoCodecConfig</u> (int firstByte, int secondByte) Returns true if the packet is a video codec config packet
static boolean	<u>isVideoKeyFrame</u> (<u>AMFPacket</u> packet) Returns true if the packet is a video key frame
static boolean	<u>isVideoKeyFrame</u> (byte[] buffer) Returns true if the packet is a video key frame
static boolean	<u>isVideoKeyFrame</u> (java.nio.ByteBuffer buffer) Returns true if the packet is a video key frame
static boolean	<u>isVideoKeyFrame</u> (int[] chunkHeaderValues) Returns true if the packet is a video key frame

static AMFPacket	readChunk (java.io.DataInput is) Read a packets worth of .flv data from an InputStream and return as an AMFPacket
static AMFPacket	readChunk (java.io.InputStream is) Read a packets worth of .flv data from an InputStream and return as an AMFPacket
static void	readChunkHeader (java.io.RandomAccessFile is, byte[] buffer, int[] values) Read packet header.
static boolean	readHeader (java.io.DataInput is) Read file header.
static boolean	readHeader (java.io.InputStream is) Read file header.
static void	readPrevChunkHeader (java.io.RandomAccessFile is, byte[] buffer, int[] values) Back up one packet from current position in the file and read the packet header.
static String	streamCodecToString (int codec) Get a printable string representation of the stream codecs defined as IVHost.CODEC_STREAM_*
static AMFPacket	updateOnCuePointTimecode (AMFPacket packet, long timecode)
static java.nio.ByteBuffer	updateOnCuePointTimecode (java.nio.ByteBuffer data, int dataType, long timecode)
static int	videoCodecStringToId (String codecString) Parse a string to get the codec ID defined by IVHost.CODEC_VIDEO_*
static String	videoCodecToMetaDataString (int codec) Get a printable string representation of the video codecs defined as IVHost.CODEC_VIDEO_* as the string used in the onMetaData event
static String	videoCodecToString (int codec) Get a printable string representation of the video codecs defined as IVHost.CODEC_VIDEO_*
static com.wowza.wms.mediawriter.MediaWriterStatus	writeChunk (java.io.DataOutput ds, java.nio.ByteBuffer data, int size, long timecode, byte type)
static com.wowza.wms.mediawriter.MediaWriterStatus	writeChunk (java.io.OutputStream ds, java.nio.ByteBuffer data, int size, long timecode, byte type) Write a packets worth of data.
static void	writeDuration (java.io.File file, double duration) Write the duration to an existing .flv file.
static void	writeHeader (java.io.OutputStream ds, double duration, int audiocodecid, int videocodecid, String createdBy, java.util.Map extraMetadata) Write file header including onMetaData packet.
static void	writeHeader (java.io.OutputStream ds, double duration, java.util.Map extraMetadata) Write file header including onMetaData packet.

static void	writeHeaderFlags (java.io.File file, byte newFlags)
static void	writePackets (java.io.OutputStream ds, java.util.List audioPackets, java.util.List videoPackets, java.util.List dataPackets, java.util.List audioTCs, java.util.List videoTCs, java.util.List dataTCs, java.util.List dataTypes, long[] currentTCs) Write a bunch of packets to .flv file all at once.
static void	writePackets (java.io.OutputStream ds, java.util.List audioPackets, java.util.List videoPackets, java.util.List dataPackets, java.util.List audioTCs, java.util.List videoTCs, java.util.List dataTCs, java.util.List dataTypes, long[] currentTCs, IFLVWriterAdjustTimecode dataPacketTimecodeAdjuster) Write a bunch of packets to .flv file all at once.
static void	writePackets (java.io.OutputStream ds, java.util.List audioPackets, java.util.List videoPackets, java.util.List dataPackets, java.util.List audioTCs, java.util.List videoTCs, java.util.List dataTCs, long[] currentTCs) Write a bunch of packets to .flv file all at once.
static void	writePackets (java.io.OutputStream ds, java.util.List packetList, long tcOffset) Write audio/video/data packets to an .flv file.
static void	writeShortHeader (java.io.DataOutput ds)
static void	writeShortHeader (java.io.OutputStream ds) Write just the FLV file header (without the metadata packet)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

FLV_CHUNKHEADER_ITYPE

public static final int **FLV_CHUNKHEADER_ITYPE**

Header values: packet type
Constant value: **0**

FLV_CHUNKHEADER_ISIZE

public static final int **FLV_CHUNKHEADER_ISIZE**

Header values: packet size
Constant value: **1**

FLV_CHUNKHEADER_ITIMECODE

public static final int **FLV_CHUNKHEADER_ITIMECODE**

Header values: timecode (milliseconds)
Constant value: **2**

FLV_CHUNKHEADER_FIRSTBYTE

```
public static final int FLV_CHUNKHEADER_FIRSTBYTE
```

Header values: first byte of packet data
Constant value: **3**

FLV_CHUNKHEADER_SECONDBYTE

```
public static final int FLV_CHUNKHEADER_SECONDBYTE
```

Header values: second byte of packet data
Constant value: **4**

FLV_CHUNKHEADER_HEADERSIZE

```
public static final int FLV_CHUNKHEADER_HEADERSIZE
```

Size of packet header (byte[])
Constant value: **11**

FLV_CHUNKHEADER_BUFFERSIZE

```
public static final int FLV_CHUNKHEADER_BUFFERSIZE
```

Size of temporary buffer needed for flv reading (byte[])
Constant value: **13**

FLV_CHUNKHEADER_VALUESIZE

```
public static final int FLV_CHUNKHEADER_VALUESIZE
```

Size of header values array (long[])
Constant value: **5**

FLV_UFRAME

```
public static final int FLV_UFRAME
```

Unknown video frame type
Constant value: **0**

FLV_KFRAME

```
public static final int FLV_KFRAME
```

Key video frame type
Constant value: **1**

FLV_DFRAME

```
public static final int FLV_DFRAME
```

D video frame type (partial frame based on key frame)
Constant value: **3**

FLV_PFRAME

```
public static final int FLV_PFRAME
```

(continued from last page)

P video frame type (partial frame based on previous frame)
Constant value: **2**

FLV_TCINDEXAUDIO

```
public static final int FLV_TCINDEXAUDIO
```

Constant value: **0**

FLV_TCINDEXVIDEO

```
public static final int FLV_TCINDEXVIDEO
```

Constant value: **1**

FLV_TCINDEXDATA

```
public static final int FLV_TCINDEXDATA
```

Constant value: **2**

Constructors

FLVUtils

```
public FLVUtils()
```

Methods

streamCodecToString

```
public static String streamCodecToString(int codec)
```

Get a printable string representation of the stream codecs defined as IVHost.CODEC_STREAM_*

Parameters:

codec - codec id defined IVHost.CODEC_STREAM_*

Returns:

codec name

audioCodecStringToId

```
public static int audioCodecStringToId(String codecString)
```

Parse a string to get the codec ID defined by IVHost.CODEC_AUDIO_*

Parameters:

codecString - codec string

Returns:

codec ID

videoCodecToMetaDataString

```
public static String videoCodecToMetaDataString(int codec)
```

Get a printable string representation of the video codecs defined as IVHost.CODEC_VIDEO_* as the string used in the onMetaData event

Parameters:

codec - codec id defined IVHost.CODEC_VIDEO_*

Returns:

codec name

audioCodecToMetaDataString

```
public static String audioCodecToMetaDataString(int codec)
```

Get a printable string representation of the audio codecs defined as IVHost.CODEC_AUDIO_* as the string used in the onMetaData event

Parameters:

codec - codec id defined IVHost.CODEC_AUDIO_*

Returns:

codec name

audioCodecToString

```
public static String audioCodecToString(int codec)
```

Get a printable string representation of the audio codecs defined as IVHost.CODEC_AUDIO_*

Parameters:

codec - codec id defined IVHost.CODEC_AUDIO_*

Returns:

codec name

videoCodecStringToId

```
public static int videoCodecStringToId(String codecString)
```

Parse a string to get the codec ID defined by IVHost.CODEC_VIDEO_*

Parameters:

codecString - codec string

Returns:

codec ID

videoCodecToString

```
public static String videoCodecToString(int codec)
```

Get a printable string representation of the video codecs defined as IVHost.CODEC_VIDEO_*

Parameters:

codec - codec id defined IVHost.CODEC_VIDEO_*

(continued from last page)

Returns:

codec name

frameTypeToString

```
public static String frameTypeToString(int frameType)
```

getFrameType

```
public static int getFrameType(int[] values)
```

Given the headers values (including first byte of the packet), determine the type of video frame (FLV_*FRAME)

Parameters:

values - header values

Returns:

type of video frame (FLV_*FRAME)

getFrameType

```
public static int getFrameType(int value)
```

Given the first byte of a video packet, determine the frame type (FLV_*FRAME)

Parameters:

value - first byte of packet

Returns:

type of video frame (FLV_*FRAME)

getAudioCodec

```
public static int getAudioCodec(int value)
```

Return the codec portion of the first byte of an audio packet. Return should be one of IVHost.CODEC_AUDIO_*

Parameters:

value - first byte of audio packet

Returns:

codec id

getVideoCodec

```
public static int getVideoCodec(int value)
```

Return the codec portion of the first byte of an video packet. Return should be one of IVHost.CODEC_VIDEO_*

Parameters:

value - first byte of audio packet

Returns:

codec id

getFrameType

```
public static int getFrameType(byte value)
```

Given the first byte of a video packet, determine the frame type (FLV_*FRAME)

Parameters:

value - first byte of packet

Returns:

type of video frame (FLV_*FRAME)

readChunk

```
public static AMFPacket readChunk(java.io.InputStream is)
```

Read a packets worth of .flv data from an InputStream and return as an AMFPacket

Parameters:

is - InputStream

Returns:

AMFPacket of data or null if unsuccessful or end of file

readChunk

```
public static AMFPacket readChunk(java.io.DataInput is)
```

Read a packets worth of .flv data from an InputStream and return as an AMFPacket

Parameters:

is - InputStream

Returns:

AMFPacket of data or null if unsuccessful or end of file

readPrevChunkHeader

```
public static void readPrevChunkHeader(java.io.RandomAccessFile is,  
    byte[] buffer,  
    int[] values)
```

Back up one packet from current position in the file and read the packet header. This includes reading the first byte of the packet data. The file pointer will be positioned at the first byte of the packet data. values[FLV_CHUNKHEADER_ITYPE] will be set to 0x7f if failure or start of file.

Parameters:

is - RandomAccessFile

buffer - temporary buffer byte[FLV_CHUNKHEADER_BUFFERSIZE]

values - header values long[FLV_CHUNKHEADER_VALUESIZE]

readChunkHeader

```
public static void readChunkHeader(java.io.RandomAccessFile is,  
    byte[] buffer,  
    int[] values)
```

Read packet header. This includes reading the first byte of the packet data. The file pointer will be positioned at the first byte of the packet data. values[FLV_CHUNKHEADER_ITYPE] will be set to 0x7f if failure or end of file.

(continued from last page)

Parameters:

is - RandomAccessFile
buffer - temporary buffer byte[FLV_CHUNKHEADER_BUFFERSIZE]
values - header values long[FLV_CHUNKHEADER_VALUESIZE]

readHeader

```
public static boolean readHeader(java.io.InputStream is)
```

Read file header. Return true if successful. Basically this just skips first 13 bytes in file.

readHeader

```
public static boolean readHeader(java.io.DataInput is)
```

Read file header. Return true if successful. Basically this just skips first 13 bytes in file.

writeShortHeader

```
public static void writeShortHeader(java.io.OutputStream ds)
```

Write just the FLV file header (without the metadata packet)

Parameters:

ds - OutputStream

writeShortHeader

```
public static void writeShortHeader(java.io.DataOutput ds)
```

writeHeader

```
public static void writeHeader(java.io.OutputStream ds,  
    double duration,  
    java.util.Map extraMetadata)
```

Write file header including onMetaData packet.

Parameters:

ds - OutputStream
duration - duration of .flv file in seconds
extraMetadata - Map of name/value pairs of metadata that will be appended to the onMetaData block

writeHeader

```
public static void writeHeader(java.io.OutputStream ds,  
    double duration,  
    int audiocodecid,  
    int videocodecid,  
    String createdBy,  
    java.util.Map extraMetadata)
```

(continued from last page)

Write file header including onMetaData packet.

With this method you can provide a Map of metadata to write to the file. This map can include a mixture of simple types like: int, long, String, boolean. These types will be wrapped in AMFData classes before they are written to the file. This map can also contain AMFData items. For example if you wanted to insert an array of **cuePoints** the code would look like:

```
Map extraMetadata = new HashMap();

AMFDataArray amfArray = new AMFDataArray();
for(int i=0;i
```

Parameters:

ds - OutputStream
 duration - duration of .flv file in seconds
 audiocodecid - audio codec ID see IVHost.CODEC_AUDIO_* (-1 for now audio)
 videocodecid - video codec ID see IVHost.CODEC_VIDEO_* (-1 for now video)
 createdBy - created by string (null for empty)
 extraMetadata - Map of name/value pairs of metadata that will be appended to the onMetaData block

writeChunk

```
public static com.wowza.wms.mediawriter.MediaWriterStatus
writeChunk(java.io.OutputStream ds,
           java.nio.ByteBuffer data,
           int size,
           long timecode,
           byte type)
```

Write a packets worth of data.

Parameters:

ds - OutputStream
 data - packet data
 size - size of the packet
 timecode - timecode (milliseconds)
 type - type of packet IVHost.CONTENTTYPE_*

writeChunk

```
public static com.wowza.wms.mediawriter.MediaWriterStatus
writeChunk(java.io.DataOutput ds,
           java.nio.ByteBuffer data,
           int size,
           long timecode,
           byte type)
```

(continued from last page)

interleavePackets

```
public static java.util.List interleavePackets(java.util.List audioPackets,
        java.util.List videoPackets,
        java.util.List dataPackets,
        java.util.List audioTCs,
        java.util.List videoTCs,
        java.util.List dataTCs,
        long[] currentTCs)
```

This is a utility function primarily used for IMediaWriters. It is a way of taking the discrete audio/video/data packets and timecodes and interleaving them into a single list of AMFPacket objects. The currentTCs array is an array of longs containing the last timecode for each of the packet types FLV_TCINDEX* (long[3]). This array will be used to calculate the absolute timecode for a given packet. For example if the relative timecode for an audio packet is 26 and the currentTCs value for the audio channel is 1003 then the absolute timecode for that packet will be 1029. The currentTCs array will be updated after this call to reflect the absolute timecode of the last packet processed of each type.

Parameters:

audioPackets - list of audio packets (ByteBuffer)
 videoPackets - list of video packets (ByteBuffer)
 dataPackets - list of data packets (ByteBuffer)
 audioTCs - list of relative audio timecodes (Long)
 videoTCs - list of relative video timecodes (Long)
 dataTCs - list of relative data timecodes (Long)
 currentTCs - array of longs containing last TCs written for each packet type FLV_TCINDEX* (long[3]). Last timecode written for each packet type will be returned in this same array.

Returns:

List of AMFPacket objects sorted by timecodes

interleavePackets

```
public static java.util.List interleavePackets(java.util.List audioPackets,
        java.util.List videoPackets,
        java.util.List dataPackets,
        java.util.List audioTCs,
        java.util.List videoTCs,
        java.util.List dataTCs,
        java.util.List dataTypes,
        long[] currentTCs)
```

This is a utility function primarily used for IMediaWriters. It is a way of taking the discrete audio/video/data packets and timecodes and interleaving them into a single list of AMFPacket objects. The currentTCs array is an array of longs containing the last timecode for each of the packet types FLV_TCINDEX* (long[3]). This array will be used to calculate the absolute timecode for a given packet. For example if the relative timecode for an audio packet is 26 and the currentTCs value for the audio channel is 1003 then the absolute timecode for that packet will be 1029. The currentTCs array will be updated after this call to reflect the absolute timecode of the last packet processed of each type.

Parameters:

audioPackets - list of audio packets (ByteBuffer)
 videoPackets - list of video packets (ByteBuffer)
 dataPackets - list of data packets (ByteBuffer)
 audioTCs - list of relative audio timecodes (Long)
 videoTCs - list of relative video timecodes (Long)
 dataTCs - list of relative data timecodes (Long)
 dataTypes - list of integer packets types (IVHost.CONTENTTYPE_DATA0, IVHost.CONTENTTYPE_DATA3) - if null assumed to be IVHost.CONTENTTYPE_DATA0
 currentTCs - array of longs containing last TCs written for each packet type FLV_TCINDEX* (long[3]). Last timecode written for each packet type will be returned in this same array.

Returns:

List of AMFPacket objects sorted by timecodes

writePackets

```
public static void writePackets(java.io.OutputStream ds,
    java.util.List packetList,
    long tcOffset)
```

Write audio/video/data packets to an .flv file. They will be written in the packetList order.

Parameters:

ds - OutputStream
 packetList - List of AMFPacket objects
 tcOffset - timecode offset

writePackets

```
public static void writePackets(java.io.OutputStream ds,
    java.util.List audioPackets,
    java.util.List videoPackets,
    java.util.List dataPackets,
    java.util.List audioTCs,
    java.util.List videoTCs,
    java.util.List dataTCs,
    long[] currentTCs)
```

Write a bunch of packets to .flv file all at once. The packets will be sorted by timecode as written

Parameters:

ds - OutputStream
 audioPackets - list of audio packets (ByteBuffer)
 videoPackets - list of video packets (ByteBuffer)
 dataPackets - list of data packets (ByteBuffer)
 audioTCs - list of relative audio timecodes (Long)
 videoTCs - list of relative video timecodes (Long)
 dataTCs - list of relative data timecodes (Long)
 currentTCs - array of longs containing last TCs written for each packet type FLV_TCINDEX* (long[3]). Last timecode written for each packet type will be returned in this same array.

writePackets

```
public static void writePackets(java.io.OutputStream ds,
    java.util.List audioPackets,
    java.util.List videoPackets,
    java.util.List dataPackets,
    java.util.List audioTCs,
    java.util.List videoTCs,
    java.util.List dataTCs,
    java.util.List dataTypes,
    long[] currentTCs)
```

Write a bunch of packets to .flv file all at once. The packets will be sorted by timecode as written

Parameters:

ds - OutputStream
 audioPackets - list of audio packets (ByteBuffer)
 videoPackets - list of video packets (ByteBuffer)
 dataPackets - list of data packets (ByteBuffer)
 audioTCs - list of relative audio timecodes (Long)
 videoTCs - list of relative video timecodes (Long)
 dataTCs - list of relative data timecodes (Long)
 dataTypes - list of integer packets types (IVHost.CONTENTTYPE_DATA0, IVHost.CONTENTTYPE_DATA3) - if null assumed to be IVHost.CONTENTTYPE_DATA0

(continued from last page)

currentTCs - array of longs containing last TCs written for each packet type FLV_TCINDEX* (long[3]). Last timecode written for each packet type will be returned in this same array.

writePackets

```
public static void writePackets(java.io.OutputStream ds,
    java.util.List audioPackets,
    java.util.List videoPackets,
    java.util.List dataPackets,
    java.util.List audioTCs,
    java.util.List videoTCs,
    java.util.List dataTCs,
    java.util.List dataTypes,
    long[] currentTCs,
    IFLVWriterAdjustTimecode dataPacketTimecodeAdjuster)
```

Write a bunch of packets to .flv file all at once. The packets will be sorted by timecode as written

Parameters:

ds - OutputStream
audioPackets - list of audio packets (ByteBuffer)
videoPackets - list of video packets (ByteBuffer)
dataPackets - list of data packets (ByteBuffer)
audioTCs - list of relative audio timecodes (Long)
videoTCs - list of relative video timecodes (Long)
dataTCs - list of relative data timecodes (Long)
dataTypes - list of integer packets types (IVHost.CONTENTTYPE_DATA0, IVHost.CONTENTTYPE_DATA3) - if null assumed to be IVHost.CONTENTTYPE_DATA0
currentTCs - array of longs containing last TCs written for each packet type FLV_TCINDEX* (long[3]). Last timecode written for each packet type will be returned
dataPacketTimecodeAdjuster - class that implements the IFLVWriterAdjustTimecode interface for adjusting timecodes in this same array.

adjustFirstPacketTCs

```
public static long adjustFirstPacketTCs(java.util.List audioTCs,
    java.util.List videoTCs,
    java.util.List dataTCs)
```

Align list of timecode for each data type. Assume first entry in each list is absolute timecode. When done lowest entry in three lists will be zero and other lists will be offset accordingly.

Parameters:

audioTCs - list of audio timecodes
videoTCs - list of video timecodes
dataTCs - list of data timecodes

Returns:

lowest of three absolute timecodes

writeDuration

```
public static void writeDuration(java.io.File file,
    double duration)
```

Write the duration to an existing .flv file. This routine will hunt through the .flv file for the onMetaData packet and the duration metadata. It will rewrite the value if found. If not found it will do nothing.

Parameters:

file - .flv file
duration - new duration value (seconds)

getLastTC

```
public static long getLastTC(java.io.File file)
```

Get the duration of an .flv file. This routine will find the onMetaData packet and the duration metadata and return the value. If not found it will read the last packet in the file and return the timecode of that packet.

Parameters:

file

Returns:

duration (milliseconds)

isVideoKeyFrame

```
public static boolean isVideoKeyFrame(AMFPacket packet)
```

Returns true if the packet is a video key frame

Parameters:

packet - packet

Returns:

true if is video key frame

isVideoCodecConfig

```
public static boolean isVideoCodecConfig(int firstByte,  
int secondByte)
```

Returns true if the packet is a video codec config packet

Parameters:

firstByte - first byte of packet

secondByte - second byte of packet

Returns:

true if video codec

isVideoCodecConfig

```
public static boolean isVideoCodecConfig(AMFPacket packet)
```

Returns true if the packet is a video codec config packet

Parameters:

packet

Returns:

true if the packet is a video codec config packet

isAudioCodecConfig

```
public static boolean isAudioCodecConfig(AMFPacket packet)
```

Returns true if the packet is a video codec config packet

Parameters:

(continued from last page)

packet

Returns:

true if the packet is a video codec config packet

getAudioMP3Layer

```
public static int getAudioMP3Layer(AMFPacket packet)
```

Audio marked as MP3 is really MPEG1 Layer 1-3. Only MPEG1 Layer 3 is truly MP3. This function will return the layer number for this packet.

Parameters:

packet - amf packet

Returns:

layer number

getAudioCodec

```
public static int getAudioCodec(AMFPacket packet)
```

Get the codec id for this audio packet. Returns IVHost.CODEC_AUDIO_UNKNOWN is unknown or not audio packet

Parameters:

packet - packet

Returns:

codec id IVHost.CODEC_AUDIO_*

getVideoCodec

```
public static int getVideoCodec(AMFPacket packet)
```

Get the codec id for this video packet. Returns IVHost.CODEC_VIDEO_UNKNOWN is unknown or not video packet

Parameters:

packet - packet

Returns:

codec id IVHost.CODEC_VIDEO_*

getVideoFrameType

```
public static int getVideoFrameType(AMFPacket packet)
```

getVideoTimecodeOffset

```
public static int getVideoTimecodeOffset(AMFPacket packet)
```

Get the timecode offset in milliseconds between the PTS and DTS for this frame.

Parameters:

packet - AMFPacket

Returns:

timecode offset in milliseconds (can be negative)

getVideoTimecodeOffset

```
public static int getVideoTimecodeOffset(byte[] buffer)
```

Get the timecode offset in milliseconds between the PTS and DTS for this frame.

Parameters:

buffer - video packet buffer

Returns:

timecode offset in milliseconds (can be negative)

isVideoKeyFrame

```
public static boolean isVideoKeyFrame(java.nio.ByteBuffer buffer)
```

Returns true if the packet is a video key frame

Parameters:

buffer - packet data (only need first two bytes of data)

Returns:

true if is video key frame

isVideoKeyFrame

```
public static boolean isVideoKeyFrame(byte[] buffer)
```

Returns true if the packet is a video key frame

Parameters:

buffer - packet data (only need first two bytes of data)

Returns:

true if is video key frame

isVideoKeyFrame

```
public static boolean isVideoKeyFrame(int[] chunkHeaderValues)
```

Returns true if the packet is a video key frame

Parameters:

chunkHeaderValues - chunk header values returned by `FLVUtils.readChunkHeader`

Returns:

true if is video key frame

updateOnCuePointTimecode

```
public static AMFPacket updateOnCuePointTimecode(AMFPacket packet,  
long timecode)
```

(continued from last page)

updateOnCuePointTimecode

```
public static java.nio.ByteBuffer updateOnCuePointTimecode(java.nio.ByteBuffer data,  
    int dataType,  
    long timecode)
```

isOnMetadataPacket

```
public static boolean isOnMetadataPacket(AMFPacket packet)
```

Returns true if packet is onMetaData or [@setDataFrame, onMetaData] data packet.

Parameters:

packet - packet

Returns:

true if onMetaData packet

getOnMetadataData

```
public static OnMetadataBasic getOnMetadataData(AMFPacket metaDataPacket)
```

getHeaderFlags

```
public static byte getHeaderFlags(java.io.File file)
```

writeHeaderFlags

```
public static void writeHeaderFlags(java.io.File file,  
    byte newFlags)
```

com.wowza.util Class HTTPUtils

java.lang.Object

└─com.wowza.util.HTTPUtils

public class **HTTPUtils**
extends Object

HTTPUtils: utility class for making http requests.

Field Summary

public static final	COOKIEFORMAT
---------------------	------------------------------

Constructor Summary

public	HTTPUtils()
--------	-----------------------------

Method Summary

static String	assembleQueryStr (java.util.Map queryMap) Assemble a map of name value pairs into a single query string.
---------------	---

static java.util.Map	expandHTTPHeadersStr (String httpUserHTTPHeadersStr)
----------------------	--

static String	formatDeleteCookie (String name, String path, String domain) Formats a cookie header value that is in the past to delete a cookie
---------------	--

static String	formatSetCookie (String name, String value, int timeoffset, String path, String domain, boolean isSecure) Format a HTTP header Set-Cookie value
---------------	--

static byte[]	HTTPRequestToByteArray (String inUrl, String method, String data, java.util.Map headers) Make a HTTP request and return the result as a byte array
---------------	---

static byte[]	HTTPRequestToByteArray (String inUrl, String method, String data, java.util.Map inHeaders, java.util.Map outHeaders, long dataLimit) Make a HTTP request and return the result as a byte array
---------------	---

static boolean	HTTPRequestToFile (java.io.File file, String inUrl, String method, String data, java.util.List headers) Make a HTTP request and have the result saved to a file.
----------------	---

static java.util.List	splitCookie (String str) Breaks Cookies header value into a list of name/value pairs.
-----------------------	--

static String[]	splitPragmas (String str) Split HTTP Pragma values at commas that separate values.
-----------------	---

static java.util.Map	<code>splitQueryStr</code> (String queryStr) Split a query string into a map and URL decode the values
static String	<code>statusCodeToStr</code> (int statusCode) Convert an HTTP status code to a string

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

COOKIEDATEFORMAT

public static final org.apache.commons.lang.time.FastDateFormat **COOKIEDATEFORMAT**

Constructors

HTTPUtils

public **HTTPUtils**()

Methods

HTTPRequestToFile

```
public static boolean HTTPRequestToFile(java.io.File file,
    String inUrl,
    String method,
    String data,
    java.util.List headers)
```

Make a HTTP request and have the result saved to a file.

Parameters:

file - destination
inUrl - url (will work with http and https)
method - method (POST, GET)
data - post data
headers - map of headers (Content-type...)

Returns:

true if successful

HTTPRequestToByteArray

```
public static byte[] HTTPRequestToByteArray(String inUrl,
    String method,
    String data,
    java.util.Map headers)
```

Make a HTTP request and return the result as a byte array

(continued from last page)

Parameters:

inUrl - url (will work with http and https)
method - method (POST, GET)
data - post data
headers - map of headers (Content-type...)

Returns:

byte array of result or null if not successful

HTTPRequestToByteArray

```
public static byte[] HTTPRequestToByteArray(String inUrl,  
      String method,  
      String data,  
      java.util.Map inHeaders,  
      java.util.Map outHeaders,  
      long dataLimit)
```

Make a HTTP request and return the result as a byte array

Parameters:

inUrl - url (will work with http and https)
method - method (POST, GET)
data - data
inHeaders - map of headers (Content-type...)
outHeaders - response headers
dataLimit - maximum number of bytes to read, zero for no limit

Returns:

byte array of result or null if not successful

assembleQueryStr

```
public static String assembleQueryStr(java.util.Map queryMap)
```

Assemble a map of name value pairs into a single query string. URL encode query string values.

Parameters:

queryMap - name value pairs

Returns:

query string

splitQueryStr

```
public static java.util.Map splitQueryStr(String queryStr)
```

Split a query string into a map and URL decode the values

Parameters:

queryStr - query string

Returns:

map of name value pairs

splitPragmas

```
public static String[] splitPragmas(String str)
```

(continued from last page)

Split HTTP Pragma values at commas that separate values. It deals with internal commas in strings. Example:

```
no-cache, client-id=1485578017, features="seekable, stridable", timeout=6
```

Result:

```
no-cache  
client-id=1485578017  
features="seekable, stridable"  
timeout=6
```

Parameters:

str - Pragma value

Returns:

array of strings broken at commas

splitCookie

```
public static java.util.List splitCookie(String str)
```

Breaks Cookies header value into a list of name/value pairs. The Cookie string: "name1=value1;name2;name1=value3" is returned as: List(item({"name1", "value1"}), item("name2", null), item("name1", "value3"))

Parameters:

str - input string

Returns:

List of name value pairs

formatDeleteCookie

```
public static String formatDeleteCookie(String name,  
    String path,  
    String domain)
```

Formats a cookie header value that is in the past to delete a cookie

Parameters:

name - variable name
path - path
domain - domain

Returns:

(continued from last page)

formatted cookie string

formatSetCookie

```
public static String formatSetCookie(String name,  
    String value,  
    int timeoffset,  
    String path,  
    String domain,  
    boolean isSecure)
```

Format a HTTP header Set-Cookie value

Parameters:

name - variable name
value - variable value - null if no value
timeoffset - expiration time in seconds
path - cookie path
domain - cookie domain
isSecure - is cookie secure

Returns:

formatted cookie string

statusCodeToStr

```
public static String statusCodeToStr(int statusCode)
```

Convert an HTTP status code to a string

Parameters:

statusCode - HTTP status code

Returns:

string

expandHTTPHeadersStr

```
public static java.util.Map expandHTTPHeadersStr(String httpUserHTTPHeadersStr)
```

com.wowza.util Interface IBandwidthThrottler

public interface **IBandwidthThrottler**
extends

Method Summary

long	<code>getBytesAllocation</code> (long request) Requests bytes from the bandwidth throttler interface.
------	--

Methods

getBytesAllocation

public long **getBytesAllocation**(long request)

Requests bytes from the bandwidth throttler interface. Return value is the number of bytes allocated

Parameters:

request - request number of bytes

Returns:

allocated number of bytes

com.wowza.util Interface IFileProcess

public interface **IFileProcess**
extends

Used by FileUtils.traverseDirectory

Method Summary

void	onFile (java.io.File file) Triggered for each file encountered in FileUtils.traverseDirectory
------	--

Methods

onFile

public void **onFile**(java.io.File file)

Triggered for each file encountered in FileUtils.traverseDirectory

Parameters:

file - file descriptor

com.wowza.util Class IOPerformanceCounter

java.lang.Object

└-com.wowza.util.IOPerformanceCounter

public class **IOPerformanceCounter**
extends Object

IOPerformanceCounter: data object that tracks the server performance of a particular component (client, vHost, server, stream). It tracks bytes and messages sent to and from the object being measured.

Constructor Summary

public	<u>IOPerformanceCounter</u> () Create an empty performance counter.
--------	---

Method Summary

void	<u>add</u> (<u>IOPerformanceCounter</u> value) Add (value) to this counter.
void	<u>addDifference</u> (<u>IOPerformanceCounter</u> current, <u>IOPerformanceCounter</u> last) Add the result of (current-last) to this counter.
void	<u>clear</u> () Clear or reset this counter back to zero.
<u>IOPerformanceCounter</u>	<u>clone</u> () Create a deep clone (copy) if this object.
void	<u>doSet</u> (<u>IOPerformanceCounter</u> value) Set this object to value.
void	<u>dummy</u> ()
long	<u>getFileInBytes</u> () Get file in bytes
double	<u>getFileInBytesRate</u> () Get estimate of file byte-in byte rate.
long	<u>getFileOutBytes</u> () Get file byte-out bytes (not implemented)
double	<u>getFileOutBytesRate</u> () Get estimate of file byte-out message byte rate (not implemented).
long	<u>getMessagesInBytes</u> () Get byte-in bytes
double	<u>getMessagesInBytesRate</u> () Get estimate of byte-in message byte rate.

long	<u>getMessagesInCount</u> () Get byte-in message count
long	<u>getMessagesInCountRate</u> () Get estimate of byte-in message count rate.
long	<u>getMessagesLossBytes</u> () Get byte-loss bytes
double	<u>getMessagesLossBytesRate</u> () Get estimate of byte-loss message byte rate.
long	<u>getMessagesLossCount</u> () Get byte-loss message count
long	<u>getMessagesLossCountRate</u> () Get estimate of byte-loss message count rate.
long	<u>getMessagesOutBytes</u> () Get byte-out bytes
double	<u>getMessagesOutBytesRate</u> () Get estimate of byte-out message byte rate.
long	<u>getMessagesOutCount</u> () Get byte-out message count
long	<u>getMessagesOutCountRate</u> () Get estimate of byte-out message count rate.
long	<u>incrementBytesIn</u> (long bytes) Increment bytes-in by bytes and increment message count by 0.
long	<u>incrementBytesLoss</u> (long bytes) Increment bytes-loss by bytes and increment message count by 0.
long	<u>incrementBytesOut</u> (long bytes) Increment bytes-out by bytes and increment message count by 0.
long	<u>incrementFileIn</u> (long bytes) Increment file-bytes-in by bytes.
long	<u>incrementFileOut</u> (long bytes) Increment file bytes-out by bytes.
long	<u>incrementMessagesIn</u> () Increment byte-in message count by 1.
long	<u>incrementMessagesIn</u> (long bytes) Increment bytes-in by bytes and increment message count by 1.
long	<u>incrementMessagesIn</u> (long bytes, long count) Increment bytes-in by bytes and message count by count.
long	<u>incrementMessagesLoss</u> () Increment byte-loss message count by 1.
long	<u>incrementMessagesLoss</u> (long bytes) Increment bytes-loss by bytes and increment message count by 1.

long	<code>incrementMessagesLoss</code> (long bytes, long count) Increment bytes-loss by bytes and message count by count.
long	<code>incrementMessagesOut</code> () Increment byte-out message count by 1.
long	<code>incrementMessagesOut</code> (long bytes) Increment bytes-out by bytes and increment message count by 1.
long	<code>incrementMessagesOut</code> (long bytes, long count) Increment bytes-out by bytes and message count by count.
boolean	<code>isDebugLog</code> ()
void	<code>setDebugLog</code> (boolean debugLog)

Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Constructors

IOPerformanceCounter

```
public IOPerformanceCounter()
```

Create an empty performance counter.

Methods

clone

```
public IOPerformanceCounter clone()
```

Create a deep clone (copy) if this object.

addDifference

```
public void addDifference(IOPerformanceCounter current,
    IOPerformanceCounter last)
```

Add the result of (current-last) to this counter. Used internally to efficiently track object performance on a timer.

Parameters:

current - current counter

last - last counter

clear

```
public void clear()
```

Clear or reset this counter back to zero.

(continued from last page)

add

```
public void add(IOPerformanceCounter value)
```

Add (value) to this counter.

Parameters:

value - value to add

doSet

```
public void doSet(IOPerformanceCounter value)
```

Set this object to value.

Parameters:

value - value to set

incrementMessagesIn

```
public long incrementMessagesIn(long bytes)
```

Increment bytes-in by bytes and increment message count by 1.

Parameters:

bytes - number of bytes

Returns:

bytes-in bytes

incrementFileIn

```
public long incrementFileIn(long bytes)
```

Increment file-bytes-in by bytes.

Parameters:

bytes - number of bytes

Returns:

bytes-in bytes

incrementMessagesIn

```
public long incrementMessagesIn(long bytes,  
    long count)
```

Increment bytes-in by bytes and message count by count.

Parameters:

bytes - number of bytes
count - number of messages

Returns:

bytes-in bytes

(continued from last page)

incrementBytesIn

```
public long incrementBytesIn(long bytes)
```

Increment bytes-in by bytes and increment message count by 0.

Parameters:

bytes

Returns:

bytes-in bytes

incrementMessagesIn

```
public long incrementMessagesIn()
```

Increment byte-in message count by 1.

Returns:

bytes-in message count

incrementMessagesOut

```
public long incrementMessagesOut(long bytes)
```

Increment bytes-out by bytes and increment message count by 1.

Parameters:

bytes - number of bytes

Returns:

bytes-out bytes

incrementFileOut

```
public long incrementFileOut(long bytes)
```

Increment file bytes-out by bytes.

Parameters:

bytes - number of bytes

Returns:

bytes-out bytes

incrementMessagesLoss

```
public long incrementMessagesLoss(long bytes)
```

Increment bytes-loss by bytes and increment message count by 1.

Parameters:

bytes - number of bytes

Returns:

bytes-loss bytes

(continued from last page)

incrementMessagesOut

```
public long incrementMessagesOut(long bytes,  
    long count)
```

Increment bytes-out by bytes and message count by count.

Parameters:

bytes - number of bytes
count - number of messages

Returns:

bytes-out bytes

incrementMessagesLoss

```
public long incrementMessagesLoss(long bytes,  
    long count)
```

Increment bytes-loss by bytes and message count by count.

Parameters:

bytes - number of bytes
count - number of messages

Returns:

bytes-loss bytes

incrementBytesOut

```
public long incrementBytesOut(long bytes)
```

Increment bytes-out by bytes and increment message count by 0.

Parameters:

bytes

Returns:

bytes-out bytes

incrementBytesLoss

```
public long incrementBytesLoss(long bytes)
```

Increment bytes-loss by bytes and increment message count by 0.

Parameters:

bytes

Returns:

bytes-loss bytes

incrementMessagesOut

```
public long incrementMessagesOut()
```

Increment byte-out message count by 1.

Returns:

(continued from last page)

bytes-out message count

incrementMessagesLoss

```
public long incrementMessagesLoss()
```

Increment byte-loss message count by 1.

Returns:

bytes-loss message count

getMessagesInCount

```
public long getMessagesInCount()
```

Get byte-in message count

Returns:

byte-in message count

getMessagesOutCount

```
public long getMessagesOutCount()
```

Get byte-out message count

Returns:

byte-out message count

getMessagesLossCount

```
public long getMessagesLossCount()
```

Get byte-loss message count

Returns:

byte-loss message count

getMessagesInBytes

```
public long getMessagesInBytes()
```

Get byte-in bytes

Returns:

byte-in bytes

getFileInBytes

```
public long getFileInBytes()
```

Get file in bytes

Returns:

file in bytes

(continued from last page)

getMessagesOutBytes

```
public long getMessagesOutBytes()
```

Get byte-out bytes

Returns:

byte-out bytes

getFileOutBytes

```
public long getFileOutBytes()
```

Get file byte-out bytes (not implemented)

Returns:

byte-out bytes

getMessagesLossBytes

```
public long getMessagesLossBytes()
```

Get byte-loss bytes

Returns:

byte-loss bytes

getMessagesInCountRate

```
public long getMessagesInCountRate()
```

Get estimate of byte-in message count rate.

Returns:

estimate of byte-in message count rate (messages per second)

getMessagesOutCountRate

```
public long getMessagesOutCountRate()
```

Get estimate of byte-out message count rate.

Returns:

estimate of byte-out message count rate (messages per second)

getMessagesLossCountRate

```
public long getMessagesLossCountRate()
```

Get estimate of byte-loss message count rate.

Returns:

estimate of byte-loss message count rate (messages per second)

getMessagesInBytesRate

```
public double getMessagesInBytesRate()
```

(continued from last page)

Get estimate of byte-in message byte rate.

Returns:

estimate of byte-in message byte rate (bytes per second)

getFileInBytesRate

```
public double getFileInBytesRate()
```

Get estimate of file byte-in byte rate.

Returns:

estimate of file byte-in byte rate (bytes per second)

getMessagesOutBytesRate

```
public double getMessagesOutBytesRate()
```

Get estimate of byte-out message byte rate.

Returns:

estimate of byte-out message byte rate (bytes per second)

getFileOutBytesRate

```
public double getFileOutBytesRate()
```

Get estimate of file byte-out message byte rate (not implemented).

Returns:

estimate of file byte-out message byte rate (bytes per second)

getMessagesLossBytesRate

```
public double getMessagesLossBytesRate()
```

Get estimate of byte-loss message byte rate.

Returns:

estimate of byte-loss message byte rate (bytes per second)

dummy

```
public void dummy()
```

isDebugLog

```
public boolean isDebugLog()
```

setDebugLog

```
public void setDebugLog(boolean debugLog)
```

com.wowza.util Class MD5DigestUtils

java.lang.Object

└─com.wowza.util.MD5DigestUtils

public class **MD5DigestUtils**
extends Object

MD5DigestUtils: MD5 hash utilities.

Field Summary

protected static	md5Digest
protected static	md5Lock

Constructor Summary

public	MD5DigestUtils()
--------	----------------------------------

Method Summary

static String	generateAuth (String method, String uri, String username, String password, String realm, String nonce) Generate an HTTP authorization response
static String	generateAuth (String a2Hash, String alHash, String realm, String nonce, String qop, String nonceCount, String cnonce) Generate an HTTP authorization response
static String	generateAuth (String method, String uri, String alHash, String realm, String nonce, String qop, String nonceCount, String cnonce) Generate an HTTP authorization response
static String	generateAuth (String method, String uri, String username, String password, String realm, String nonce, String qop, String nonceCount, String cnonce) Generate an HTTP authorization response
static String	generateAuthHTTP (String method, String uri, String username, String password, String realm, String nonce, String qop, String nonceCount, String cnonce) Generate an HTTP authorization response
static String	generateHash (String value) Generate MD5 hash
static byte[]	generateHashBytes (byte[] value) Generate MD5 hash
static byte[]	generateHashBytes (String value) Generate MD5 hash

Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Fields**md5Digest**

protected static `java.security.MessageDigest` **md5Digest**

md5Lock

protected static `java.lang.Object` **md5Lock**

Constructors**MD5DigestUtils**

public **MD5DigestUtils**()

Methods**generateAuth**

```
public static String generateAuth(String method,
    String uri,
    String username,
    String password,
    String realm,
    String nonce)
```

Generate an HTTP authorization response

Parameters:

method - method
uri - URI
username - username
password - password
realm - realm
nonce - nonce

Returns:

auth response

(continued from last page)

generateAuth

```
public static String generateAuth(String method,  
    String uri,  
    String username,  
    String password,  
    String realm,  
    String nonce,  
    String qop,  
    String nonceCount,  
    String cnonce)
```

Generate an HTTP authorization response

Parameters:

method - method
uri - URI
username - username
password - password
realm - realm
nonce - nonce
qop - qop
nonceCount - nonceCount
cnonce - cnonce

Returns:

auth response

generateAuthHTTP

```
public static String generateAuthHTTP(String method,  
    String uri,  
    String username,  
    String password,  
    String realm,  
    String nonce,  
    String qop,  
    String nonceCount,  
    String cnonce)
```

Generate an HTTP authorization response

Parameters:

method - method
uri - URI
username - username
password - password
realm - realm
nonce - nonce
qop - qop
nonceCount - nonceCount
cnonce - cnonce

Returns:

response

(continued from last page)

generateAuth

```
public static String generateAuth(String method,  
    String uri,  
    String alHash,  
    String realm,  
    String nonce,  
    String qop,  
    String nonceCount,  
    String cnonce)
```

Generate an HTTP authorization response

Parameters:

method - method
uri - URI
alHash - alHash
realm - realm
nonce - nonce
qop - qop
nonceCount - nonceCount
cnonce - cnonce

Returns:

auth

generateAuth

```
public static String generateAuth(String a2Hash,  
    String alHash,  
    String realm,  
    String nonce,  
    String qop,  
    String nonceCount,  
    String cnonce)
```

Generate an HTTP authorization response

Parameters:

a2Hash - a2Hash
alHash - alHash
realm - realm
nonce - nonce
qop - qop
nonceCount - nonceCount
cnonce - cnonce

Returns:

auth

generateHashBytes

```
public static byte[] generateHashBytes(byte[] value)
```

Generate MD5 hash

Parameters:

value - byte array to hash

Returns:

byte array result

generateHashBytes

```
public static byte[] generateHashBytes(String value)
```

Generate MD5 hash

Parameters:

value - in string converted to byte array (UTF-8)

Returns:

byte array result

generateHash

```
public static String generateHash(String value)
```

Generate MD5 hash

Parameters:

value - in string converted to byte array (UTF-8)

Returns:

hash as binhex string

com.wowza.util Class MediaUtils

java.lang.Object

└─com.wowza.util.MediaUtils

public class **MediaUtils**
extends Object

MediaUtils: Media utilities.

Field Summary

public static final	CODECSTR_FORMAT_VIDEO_POSTIOS4 Value: 2
public static final	CODECSTR_FORMAT_VIDEO_PREIOS4 Value: 1
public static final	CODECSTR_FORMAT_VIDEO_UNKNOWN Value: 0

Constructor Summary

public	MediaUtils()
--------	------------------------------

Method Summary

static String	audioCodecTypeToString (int codecType) Audio codec ID to string.
static String	videoCodecTypeToString (int codecType) Video codec ID to string.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

CODECSTR_FORMAT_VIDEO_UNKNOWN

public static final int **CODECSTR_FORMAT_VIDEO_UNKNOWN**

Constant value: **0**

CODECSTR_FORMAT_VIDEO_PREIOS4

```
public static final int CODECSTR_FORMAT_VIDEO_PREIOS4
```

Constant value: **1**

CODECSTR_FORMAT_VIDEO_POSTIOS4

```
public static final int CODECSTR_FORMAT_VIDEO_POSTIOS4
```

Constant value: **2**

Constructors

MediaUtils

```
public MediaUtils()
```

Methods

audioCodecTypeToString

```
public static String audioCodecTypeToString(int codecType)
```

Audio codec ID to string. See IVHost.CODEC_AUDIO_*

Parameters:

codecType - codec id. See IVHost.CODEC_AUDIO_*

Returns:

codec string

videoCodecTypeToString

```
public static String videoCodecTypeToString(int codecType)
```

Video codec ID to string. See IVHost.CODEC_VIDEO_*

Parameters:

codecType - codec id. See IVHost.CODEC_VIDEO_*

Returns:

codec string

com.wowza.util Class NetworkUtils

java.lang.Object

└─com.wowza.util.NetworkUtils

public class **NetworkUtils**
extends Object

NetworkUtils: Networking utilities.

Constructor Summary

public	NetworkUtils()
--------	--------------------------------

Method Summary

static boolean	isAddressMulticast (String IpAddress) Returns true if IP address is multicast address
static String	socketAddressToUniqueStr (java.net.InetSocketAddress socketAddress)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

NetworkUtils

public **NetworkUtils**()

Methods

isAddressMulticast

public static boolean **isAddressMulticast**(String IpAddress)

Returns true if IP address is multicast address

Parameters:

IpAddress - IP address

Returns:

true if IP address is multicast address

socketAddressToUniqueStr

```
public static String socketAddressToUniqueStr( java.net.InetSocketAddress  
socketAddress)
```

com.wowza.util Interface NoMBean

public interface **NoMBean**
extends `Annotation`

Annotation for excluding a method from the JMX interface. Below is an example of how it would be used

```
import com.wowza.util.NoMBean;
import com.wowza.wms.module.*;

class MyClass extends ModuleBase
{
    @NoMBean public void myMethod()
    {
    }
}
```

Methods inherited from interface <code>java.lang.annotation.Annotation</code>

<code>annotationType</code> , <code>equals</code> , <code>hashCode</code> , <code>toString</code>

com.wowza.util Class StringUtils

java.lang.Object

└─com.wowza.util.StringUtils

public class **StringUtils**
extends Object

StringUtils: utility class of String utilities.

Constructor Summary

public	StringUtils()
--------	-------------------------------

Method Summary

static boolean	equals (String s1, String s2) Checks if 2 strings are equals, accounting for null cases.
static String	intToHexStr (int value, int strLen)
static String	intToStr (int value, int strLen)
static boolean	isEmpty (String s) Checks if string is empty, handling null String case.
static int	length (String s) Returns string length, handling null String case as length of 0.
static String	longToHexStr (long value, int strLen)
static String	longToStr (long value, int strLen)
static String	stampToString (long stamp) Convert a duration (milliseconds) to a formatted string.
static String	stampToStringNoUnits (long stamp) Convert a duration (milliseconds) to a formatted string.
static long	stringToMilliseconds (String stamp) Convert a hh:mm:ss.SSS string to milliseconds No validation that hh:mm:ss are valid ranges
static String	toStringList (java.util.List strings) Convert list of Strings to comma-separated string
static String	toStringList (String[] strings) Convert array of Strings to comma-separated string
static String	toStringList (WMSProperties properties) Convert properties to comma-separated string

static String	valueOf (String s) Returns string, handling null String case as "".
---------------	--

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

StringUtils

```
public StringUtils()
```

Methods

intToStr

```
public static String intToStr(int value,  
                               int strLen)
```

intToHexStr

```
public static String intToHexStr(int value,  
                                  int strLen)
```

longToStr

```
public static String longToStr(long value,  
                                int strLen)
```

longToHexStr

```
public static String longToHexStr(long value,  
                                   int strLen)
```

isEmpty

```
public static boolean isEmpty(String s)
```

Checks if string is empty, handling null String case.

Parameters:

s

Returns:

(continued from last page)

false if string is null or "", true otherwise

length

```
public static int length(String s)
```

Returns string length, handling null String case as length of 0.

Parameters:

s

Returns:

0 if string is null, or length of string.

valueOf

```
public static String valueOf(String s)
```

Returns string, handling null String case as "".

Parameters:

s

Returns:

"" if string is null, or string value.

stampToString

```
public static String stampToString(long stamp)
```

Convert a duration (milliseconds) to a formatted string.

Parameters:

stamp - duration (milliseconds)

Returns:

formatted string (example: 3 days 4 minutes 1 seconds)

stampToStringNoUnits

```
public static String stampToStringNoUnits(long stamp)
```

Convert a duration (milliseconds) to a formatted string.

Parameters:

stamp - duration (milliseconds)

Returns:

formatted string (example: 03:04:01.123 hours:minutes:seconds.milliseconds)

stringToMilliseconds

```
public static long stringToMilliseconds(String stamp)
```

Convert a hh:mm:ss.SSS string to milliseconds No validation that hh:mm:ss are valid ranges

Parameters:

stamp - string (hours:minutes:seconds.milliseconds)

(continued from last page)

Returns:milliseconds, -1 is parse failed

equals

```
public static boolean equals(String s1,  
                             String s2)
```

Checks if 2 strings are equals, accounting for null cases. If both are null, they are considered equal.

Parameters:

s1 - string one

s2 - string 2

Returns:true is the strings are equal

toStringList

```
public static String toStringList(WMSPProperties properties)
```

Convert properties to comma-separated string

Parameters:

properties

Returns:comma-separated string

toStringList

```
public static String toStringList(java.util.List strings)
```

Convert list of Strings to comma-separated string

Parameters:

strings

Returns:comma-separated string

toStringList

```
public static String toStringList(String[] strings)
```

Convert array of Strings to comma-separated string

Parameters:

strings

Returns:comma-separated string

com.wowza.util Class SystemUtils

java.lang.Object

└─com.wowza.util.SystemUtils

public class **SystemUtils**
extends Object

Nested Class Summary

class	SystemUtils.ReplaceItem SystemUtils.ReplaceItem
-------	--

Field Summary

public static final	defaultLocale
public static final	defaultTimeZone
public static final	gmtTimeZone
protected static final	msb0baseTime Value: 2085978496000
protected static final	msblbaseTime Value: -2208988800000

Constructor Summary

public	SystemUtils()
--------	-------------------------------

Method Summary

static void	addBouncyCastleSecurityProvider() Load bouncy castle providers, Internal use.
static String	expandEnvironmentVariables (String inValue) Expand system level Java properties in a String in the form \${property-name}
static String	expandEnvironmentVariables (String inValue, java.util.Map valueMap) Expand system level Java properties in a String in the form \${property-name}.
static long	getCpuTime () Get CPU time in nanoseconds.
static byte[]	getMACAddress () Get MAC address of localhost interface (only works on Java 6 or greater)

static long	<code>getSystemTime()</code> Get system time in nanoseconds.
static long	<code>getUserTime()</code> Get user time in nanoseconds.
static long	<code>toNTPTime(long t)</code> Convert a timecode value in milliseconds to NTP time

Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Fields

`defaultTimeZone`

`public static final java.util.TimeZone defaultTimeZone`

`gmtTimeZone`

`public static final java.util.TimeZone gmtTimeZone`

`defaultLocale`

`public static final java.util.Locale defaultLocale`

`msb0baseTime`

`protected static final long msb0baseTime`

Constant value: **2085978496000**

`msb1baseTime`

`protected static final long msb1baseTime`

Constant value: **-2208988800000**

Constructors

`SystemUtils`

`public SystemUtils()`

(continued from last page)

Methods

expandEnvironmentVariables

```
public static String expandEnvironmentVariables(String inValue)
```

Expand system level Java properties in a String in the form `${property-name}`

Parameters:

inValue - string with properties

Returns:

expanded string

expandEnvironmentVariables

```
public static String expandEnvironmentVariables(String inValue,  
        java.util.Map valueMap)
```

Expand system level Java properties in a String in the form `${property-name}`. You can also pass in a map of name/value pairs that will expand the list of available properties

Parameters:

inValue - string with properties

valueMap - name/value pair map

Returns:

expanded string

getMACAddress

```
public static byte[] getMACAddress()
```

Get MAC address of localhost interface (only works on Java 6 or greater)

Returns:

MAC address of localhost interface

addBouncyCastleSecurityProvider

```
public static void addBouncyCastleSecurityProvider()
```

Load bouncy castle providers, Internal use.

toNTPTime

```
public static long toNTPTime(long t)
```

Convert a timecode value in milliseconds to NTP time

Parameters:

t - timecode value in milliseconds

Returns:

NTP time

(continued from last page)

getCpuTime

```
public static long getCpuTime()
```

Get CPU time in nanoseconds.

getUserTime

```
public static long getUserTime()
```

Get user time in nanoseconds.

getSystemTime

```
public static long getSystemTime()
```

Get system time in nanoseconds.

com.wowza.util Class SystemUtils.ReplaceItem

java.lang.Object

└─com.wowza.util.SystemUtils.ReplaceItem

public static class **SystemUtils.ReplaceItem**
extends Object

Field Summary

public	end
public	newValue
public	start

Constructor Summary

public	SystemUtils.ReplaceItem (int start, int stop, String newValue)
--------	--

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

start

public int **start**

end

public int **end**

newValue

public java.lang.String **newValue**

Constructors

(continued from last page)

SystemUtils.ReplaceItem

```
public SystemUtils.ReplaceItem(int start,  
                               int stop,  
                               String newValue)
```


com.wowza.util Class URLUtils

java.lang.Object

└─com.wowza.util.URLUtils

public final class **URLUtils**
extends Object

Class to deal with URLs

Constructor Summary

public	URLUtils()
--------	----------------------------

Method Summary

static String	appendParamsToUrl (String url, String params) Convenience call to add parameters to a url.
static String	decodeValue (String val) Wrapper for URLDecoder.decode(val, "UTF-8");
static String	encodeValue (String val) Wrapper for URLEncoder.encode(val, "UTF-8");
static String	getParamValue (java.util.Map params, String key) Helper funtion to get single value from multiple value parameter Map
static java.util.Map	parseQueryStr (String queryString, boolean doDecode) Parse query string part of url into Map of Lists (to support multiple values) of query parameters
static String	pathToFileURL (String basePath) Convert a path to a url (file://[path])
static String	urlToId (java.net.URL url) Create a MD5 message digest hash of a url

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

URLUtils

public **URLUtils**()

(continued from last page)

Methods

pathToFileURL

```
public static String pathToFileURL(String basePath)
```

Convert a path to a url (file://[path])

Parameters:

basePath - path

Returns:

file url

urlToId

```
public static String urlToId(java.net.URL url)
```

Create a MD5 message digest hash of a url

Parameters:

url - url to hash

Returns:

hashed url

appendParamsToUrl

```
public static String appendParamsToUrl(String url,  
    String params)
```

Convenience call to add parameters to a url. If already has ? add &.

Parameters:

url - input url

params - parameters in the form param1=val1¶m2=val2

Returns:

appended url

encodeValue

```
public static String encodeValue(String val)
```

Wrapper for URLEncoder.encode(val, "UTF-8");

Parameters:

val - value to encode

Returns:

encoded value

decodeValue

```
public static String decodeValue(String val)
```

Wrapper for URLDecoder.decode(val, "UTF-8");

(continued from last page)

Parameters:

val - value to decode

Returns:decoded value

parseQueryStr

```
public static java.util.Map parseQueryStr(String queryString,  
                                           boolean doDecode)
```

Parse query string part of url into Map of Lists (to support multiple values) of query parameters

Parameters:

queryString - query string

doDecode - true to use URLDecoder.decode() to decode parameters

Returns:Map of Lists

getParamValue

```
public static String getParamValue(java.util.Map params,  
                                     String key)
```

Helper funtion to get single value from multiple value parameter Map

Parameters:

params - params Map

key - key string

Returns:first value

com.wowza.util Class XMLUtils

java.lang.Object

└─com.wowza.util.XMLUtils

public class **XMLUtils**
extends Object

XMLUtils: utility class for parsing XML files.

Constructor Summary

public	XMLUtils()
--------	----------------------------

Method Summary

static org.w3c.dom.Node	getNodeByTagName (org.w3c.dom.Element node, String name) Return a child Node by tag name.
static String	getNodeValue (org.w3c.dom.Node node) Return the text value of a node.
static String	getNodeValueByTagName (org.w3c.dom.Element node, String name) Get a child Node value by tag name.
static String	getVersion (org.w3c.dom.Element root) Given a Wowza XML file's root element, parse the version attribute
static boolean	getXMLPropertyBool (javax.xml.xpath.XPath xpath, String xpathStr, org.w3c.dom.Element root, boolean defaultVal) Get a boolean int value pointed to by xpath or defaultVal if not found.
static double	getXMLPropertyDouble (javax.xml.xpath.XPath xpath, String xpathStr, org.w3c.dom.Element root, double defaultVal) Get a single double value pointed to by xpath or defaultVal if not found.
static boolean	getXMLPropertyExists (javax.xml.xpath.XPath xpath, String xpathStr, org.w3c.dom.Element root)
static int	getXMLPropertyInt (javax.xml.xpath.XPath xpath, String xpathStr, org.w3c.dom.Element root, int defaultVal) Get a single int value pointed to by xpath or defaultVal if not found.
static int	getXMLPropertyIntSize (javax.xml.xpath.XPath xpath, String xpathStr, org.w3c.dom.Element root, int defaultVal)
static long	getXMLPropertyLong (javax.xml.xpath.XPath xpath, String xpathStr, org.w3c.dom.Element root, long defaultVal) Get a single long value pointed to by xpath or defaultVal if not found.
static long	getXMLPropertyLongSize (javax.xml.xpath.XPath xpath, String xpathStr, org.w3c.dom.Element root, long defaultVal)

static String	getXMLPropertyStr (javax.xml.xpath.XPath xpath, String xpathStr, org.w3c.dom.Element root) Get a single string value pointed to by xpath or null if not found.
static String	getXMLPropertyStr (javax.xml.xpath.XPath xpath, String xpathStr, org.w3c.dom.Element root, String defaultVal) Get a single string value pointed to by xpath or defaultVal if not found.
static void	loadConfigProperties (org.w3c.dom.Element root, String propertiesXPath, WMSProperties properties) Loads <Properties> elemnt by xpath into properties object.
static void	loadConfigProperties (org.w3c.dom.NodeList resultList, WMSProperties properties) Given a nodeList load children as properties.
static javax.xml.xpath.XPath Factory	newXPathFactory () Get a new XPath factory object.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

XMLUtils

public **XMLUtils**()

Methods

getNodeValueByTagName

public static String **getNodeValueByTagName**(org.w3c.dom.Element node, String name)

Get a child Node value by tag name. Return null if does not exist.

Parameters:

node - parent node
name - tag name

Returns:

node value or null if does not exist

getNodeByTagName

public static org.w3c.dom.Node **getNodeByTagName**(org.w3c.dom.Element node, String name)

Return a child Node by tag name. Return null if does not exist.

Parameters:

node - parent node

(continued from last page)

name - tag name

Returns:

Node or null if does not exist

getNodeValue

```
public static String getNodeValue(org.w3c.dom.Node node)
```

Return the text value of a node. Return concatenated value of all children nodes that are text nodes.

Parameters:

node - parent node

Returns:

concatenated text nodes or empty string if not found or no children

loadConfigProperties

```
public static void loadConfigProperties(org.w3c.dom.Element root,  
    String propertiesXPath,  
    WMSProperties properties)
```

Loads <Properties> elemnt by xpath into properties object.

Parameters:

root - root node

propertiesXPath - xpath to search for <Properties> element

properties - properties to add values to

loadConfigProperties

```
public static void loadConfigProperties(org.w3c.dom.NodeList resultList,  
    WMSProperties properties)
```

Given a nodeList load children as properties.

Parameters:

resultList - node list

properties - properties to add values to

getXMLPropertyStr

```
public static String getXMLPropertyStr(javax.xml.xpath.XPath xpath,  
    String xpathStr,  
    org.w3c.dom.Element root)
```

Get a single string value pointed to by xpath or null if not found.

Parameters:

xpath - XPath interface

xpathStr - xpath string

root - root node to start search

Returns:

single string value pointed to by xpath or null if not found

(continued from last page)

getXMLPropertyStr

```
public static String getXMLPropertyStr( javax.xml.xpath.XPath xpath,
    String xpathStr,
    org.w3c.dom.Element root,
    String defaultVal)
```

Get a single string value pointed to by xpath or defaultVal if not found.

Parameters:

xpath - XPath interface
xpathStr - xpath string
root - node to start search
defaultVal - default value

Returns:

single string value pointed to by xpath or defaultVal if not found

getXMLPropertyExists

```
public static boolean getXMLPropertyExists( javax.xml.xpath.XPath xpath,
    String xpathStr,
    org.w3c.dom.Element root)
```

getXMLPropertyInt

```
public static int getXMLPropertyInt( javax.xml.xpath.XPath xpath,
    String xpathStr,
    org.w3c.dom.Element root,
    int defaultVal)
```

Get a single int value pointed to by xpath or defaultVal if not found.

Parameters:

xpath - XPath interface
xpathStr - xpath string
root - node to start search
defaultVal - default value

Returns:

int value or defaultVal if not found

getXMLPropertyIntSize

```
public static int getXMLPropertyIntSize( javax.xml.xpath.XPath xpath,
    String xpathStr,
    org.w3c.dom.Element root,
    int defaultVal)
```

getXMLPropertyLong

```
public static long getXMLPropertyLong( javax.xml.xpath.XPath xpath,
    String xpathStr,
    org.w3c.dom.Element root,
    long defaultVal)
```

Get a single long value pointed to by xpath or defaultVal if not found.

(continued from last page)

Parameters:

xpath - XPath interface
xpathStr - xpath string
root - node to start search
defaultVal - default value

Returns:

long value or defaultVal if not found

getXMLPropertyLongSize

```
public static long getXMLPropertyLongSize( javax.xml.xpath.XPath xpath,
      String xpathStr,
      org.w3c.dom.Element root,
      long defaultVal)
```

getXMLPropertyDouble

```
public static double getXMLPropertyDouble( javax.xml.xpath.XPath xpath,
      String xpathStr,
      org.w3c.dom.Element root,
      double defaultVal)
```

Get a single double value pointed to by xpath or defaultVal if not found.

Parameters:

xpath - XPath interface
xpathStr - xpath string
root - node to start search
defaultVal - default value

Returns:

double value or defaultVal if not found

getXMLPropertyBool

```
public static boolean getXMLPropertyBool( javax.xml.xpath.XPath xpath,
      String xpathStr,
      org.w3c.dom.Element root,
      boolean defaultVal)
```

Get a boolean int value pointed to by xpath or defaultVal if not found.

Parameters:

xpath - XPath interface
xpathStr - xpath string
root - node to start search
defaultVal - default value

Returns:

boolean value or defaultVal if not found

newXPathFactory

```
public static javax.xml.xpath.XPathFactory newXPathFactory()
```


(continued from last page)

Get a new XPath factory object. There is a bug in certain implementations of the Sun VM that forces an explicit creation of the "com.sun.org.apache.xpath.internal.jaxp.XPathFactoryImpl". This method will try the correct method for creation and if fails will directly create "com.sun.org.apache.xpath.internal.jaxp.XPathFactoryImpl" object.

Returns:

XPathFactory or null if not found

getVersion

```
public static String getVersion(org.w3c.dom.Element root)
```

Given a Wowza XML file's root element, parse the version attribute

Parameters:

root

Returns:

version or "0" if not specified

Package

com.wowza.wms.amf

com.wowza.wms.amf

Class AMF3Utils

java.lang.Object

└─com.wowza.wms.amf.AMF3Utils

public class **AMF3Utils**
extends Object

AMF3 utilities

Constructor Summary

public	AMF3Utils()
--------	-----------------------------

Method Summary

static java.util.Date	deserializeDate (java.nio.ByteBuffer data) Deserialize date type
static int	deserializeInt (java.nio.ByteBuffer data) Deserialize int
static String	deserializeString (java.nio.ByteBuffer data) Deserialize string
static String	deserializeString (java.nio.ByteBuffer data, AMFDataContextDeserialize context) Deserialize string
static String	deserializeString (java.nio.ByteBuffer data, int utflen) Deserialize string
static int	serializeDate (java.io.DataOutputStream out, java.util.Date date) Serialize a date object
static int	serializeInt (java.io.DataOutputStream out, int val) Serialize int value
static int	serializeString (java.io.DataOutputStream out, String str) Serialize a string value
static int	serializeStringNoLength (java.io.DataOutputStream out, String str) Serialize string but do not write the length
static void	serializeZeroLengthString (java.io.DataOutputStream out) Serialize empty string

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

AMF3Utils

```
public AMF3Utils()
```

Methods

deserializeDate

```
public static java.util.Date deserializeDate(java.nio.ByteBuffer data)
```

Deserialize date type

Parameters:

data - buffer

Returns:

date

serializeDate

```
public static int serializeDate(java.io.DataOutputStream out,  
    java.util.Date date)
```

Serialize a date object

Parameters:

out - output stream

date - date

Returns:

number of bytes written

deserializeInt

```
public static int deserializeInt(java.nio.ByteBuffer data)
```

Deserialize int

Parameters:

data - buffer

Returns:

int value

serializeInt

```
public static int serializeInt(java.io.DataOutputStream out,  
    int val)
```

Serialize int value

Parameters:

(continued from last page)

out - output stream
val - int value

Returns:
number of bytes written

deserializeString

```
public static String deserializeString(java.nio.ByteBuffer data,  
    AMFDataContextDeserialize context)  
    throws java.io.IOException
```

Deserialize string

Parameters:
data - data
context - AMF context

Returns:
string value

Throws:
IOException

deserializeString

```
public static String deserializeString(java.nio.ByteBuffer data)  
    throws java.io.IOException
```

Deserialize string

Parameters:
data - data

Returns:
string value

Throws:
IOException

deserializeString

```
public static String deserializeString(java.nio.ByteBuffer data,  
    int utflen)  
    throws java.io.IOException
```

Deserialize string

Parameters:
data - data
utflen - len of the string

Returns:
string value

Throws:
IOException

(continued from last page)

serializeZeroLengthString

```
public static void serializeZeroLengthString(java.io.DataOutputStream out)
```

Serialize empty string

Parameters:

out - output stream

serializeStringNoLength

```
public static int serializeStringNoLength(java.io.DataOutputStream out,  
    String str)  
throws java.io.IOException
```

Serialize string but do not write the length

Parameters:

out - output stream

str - string value

Returns:

number of bytes written

Throws:

IOException

serializeString

```
public static int serializeString(java.io.DataOutputStream out,  
    String str)  
throws java.io.IOException
```

Serialize a string value

Parameters:

out - output stream

str - string value

Returns:

number of bytes written

Throws:

IOException

com.wowza.wms.amf

Class AMFData

java.lang.Object

└─com.wowza.wms.amf.AMFData

Direct Known Subclasses:

[AMFDataObj](#), [AMFDataList](#), [AMFDataItem](#), [AMFDataByteArray](#), [AMFDataArray](#)

```
public abstract class AMFData
extends Object
```

Base abstract class for data in Action Message Format (AMF). Data is sent between the Flash client and the Wowza Streaming Engine using the AMF format. This class cannot be instantiated. It serves as the base class for all AMFData objects.

Field Summary

public static final	AMF_LEVEL0 Value: 0
public static final	AMF_LEVEL3 Value: 3
public static final	DATA_TYPE_AMF3 Value: 17
public static final	DATA_TYPE_AMF3_ARRAY Value: 9
public static final	DATA_TYPE_AMF3_BOOLEAN_FALSE Value: 2
public static final	DATA_TYPE_AMF3_BOOLEAN_TRUE Value: 3
public static final	DATA_TYPE_AMF3_BYTEARRAY Value: 12
public static final	DATA_TYPE_AMF3_DATE Value: 8
public static final	DATA_TYPE_AMF3_INTEGER Value: 4
public static final	DATA_TYPE_AMF3_NULL Value: 1

public static final	DATA_TYPE_AMF3_NUMBER Value: 5
public static final	DATA_TYPE_AMF3_OBJECT Value: 10
public static final	DATA_TYPE_AMF3_STRING Value: 6
public static final	DATA_TYPE_AMF3_UNDEFINED Value: 0
public static final	DATA_TYPE_AMF3_XML_LEGACY Value: 7
public static final	DATA_TYPE_AMF3_XML_TOP Value: 11
public static final	DATA_TYPE_ARRAY Value: 10
public static final	DATA_TYPE_AS_OBJECT Value: 13
public static final	DATA_TYPE_BOOLEAN Value: 1
public static final	DATA_TYPE_BYTEARRAY Value: 33
public static final	DATA_TYPE_CUSTOM_CLASS Value: 16
public static final	DATA_TYPE_DATE Value: 11
public static final	DATA_TYPE_INTEGER Value: 32
public static final	DATA_TYPE_LONG_STRING Value: 12
public static final	DATA_TYPE_MIXED_ARRAY Value: 8
public static final	DATA_TYPE_MOVIE_CLIP Value: 4

public static final	DATA_TYPE_NULL Value: 5
public static final	DATA_TYPE_NUMBER Value: 0
public static final	DATA_TYPE_OBJECT Value: 3
public static final	DATA_TYPE_OBJECT_END Value: 9
public static final	DATA_TYPE_RECORDSET Value: 14
public static final	DATA_TYPE_REFERENCE_OBJECT Value: 7
public static final	DATA_TYPE_STRING Value: 2
public static final	DATA_TYPE_UNDEFINED Value: 6
public static final	DATA_TYPE_UNKNOWN Value: -1
public static final	DATA_TYPE_XML Value: 15
public static final	DATA_TYPE_XML_TOP Value: 34
public static final	MILLS_PER_HOUR Number of milliseconds in an hour Value: 60000
protected	type

Constructor Summary

public	AMFData()
--------	---------------------------

Method Summary

static AMFDataContextDeserialize	createContextDeserialize() Create an AMF3 deserialization context
---	--

static AMFDataContextDeserialize	createContextDeserialize (int objectEncoding) Create an AMF3 deserialization context
static AMFDataContextSerialize	createContextSerialize () Create an AMF3 serialization context
static AMFDataContextSerialize	createContextSerialize (int objectEncoding) Create an AMF3 serialization context
abstract void	deserialize (java.nio.ByteBuffer data) Deserialize data in byte buffer
abstract void	deserialize (java.nio.ByteBuffer data, AMFDataContextDeserialize context) Deserialize data in byte buffer
static AMFData	deserializeInnerObject (java.nio.ByteBuffer data, AMFDataContextDeserialize context) Deserialize next item
static AMFData	getReference (java.nio.ByteBuffer data, AMFDataContextDeserialize context) Get AMF3 object reference
int	getType () Returns the data type for this object
abstract Object	getValue () Convert object to Java native class
static boolean	isAMF3Start (java.nio.ByteBuffer data, AMFDataContextDeserialize context) Returns true if next byte in data is AMF3 start
static boolean	isArrayStart (java.nio.ByteBuffer data, AMFDataContextDeserialize context) Returns true if next byte in data is an array start
static boolean	isByteArrayStart (java.nio.ByteBuffer data, AMFDataContextDeserialize context) Returns true if next byte is ByteArray start
static boolean	isMixedArrayStart (java.nio.ByteBuffer data, AMFDataContextDeserialize context) Returns true if next byte in data is mixed array start
static boolean	isObjEnd (java.nio.ByteBuffer data, AMFDataContextDeserialize context) Returns true if next byte is object end
static boolean	isObjStart (java.nio.ByteBuffer data, AMFDataContextDeserialize context) Returns true if next byte in data is object start
static int	peekByte (java.nio.ByteBuffer data) Return the next byte in the buffer without incrementing the data position
abstract byte[]	serialize () Serial object to byte array
abstract byte[]	serialize (AMFDataContextSerialize context) Serial object to byte array

abstract void	serialize (java.io.DataOutputStream out) Serialize object to output stream
abstract void	serialize (java.io.DataOutputStream out, AMFDataContextSerialize context) Serialize object to output stream
abstract void	serialize (java.io.DataOutputStream out, int objectEncoding) Serialize object to output stream
abstract byte[]	serialize (int objectEncoding) Serial object to byte array
void	setType (int type) Sets the the data type for this object
static int	skipByte (java.nio.ByteBuffer data) Skip forward one byte in the byte buffer
static boolean	testNextByte (java.nio.ByteBuffer data, int test) Peek at the next value in data to see if its the test value
static boolean	triggerAMF3Switch (AMFData data) Return true if the object is serialized differently in AMF3

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

DATA_TYPE_UNKNOWN

```
public static final byte DATA_TYPE_UNKNOWN
```

Constant value: **-1**

DATA_TYPE_NUMBER

```
public static final byte DATA_TYPE_NUMBER
```

Constant value: **0**

DATA_TYPE_BOOLEAN

```
public static final byte DATA_TYPE_BOOLEAN
```

Constant value: **1**

DATA_TYPE_STRING

```
public static final byte DATA_TYPE_STRING
```

(continued from last page)

Constant value: **2**

DATA_TYPE_OBJECT

```
public static final byte DATA_TYPE_OBJECT
```

Constant value: **3**

DATA_TYPE_MOVIE_CLIP

```
public static final byte DATA_TYPE_MOVIE_CLIP
```

Constant value: **4**

DATA_TYPE_NULL

```
public static final byte DATA_TYPE_NULL
```

Constant value: **5**

DATA_TYPE_UNDEFINED

```
public static final byte DATA_TYPE_UNDEFINED
```

Constant value: **6**

DATA_TYPE_REFERENCE_OBJECT

```
public static final byte DATA_TYPE_REFERENCE_OBJECT
```

Constant value: **7**

DATA_TYPE_MIXED_ARRAY

```
public static final byte DATA_TYPE_MIXED_ARRAY
```

Constant value: **8**

DATA_TYPE_OBJECT_END

```
public static final byte DATA_TYPE_OBJECT_END
```

Constant value: **9**

DATA_TYPE_ARRAY

```
public static final byte DATA_TYPE_ARRAY
```

Constant value: **10**

(continued from last page)

DATA_TYPE_DATE

```
public static final byte DATA_TYPE_DATE
```

Constant value: **11**

DATA_TYPE_LONG_STRING

```
public static final byte DATA_TYPE_LONG_STRING
```

Constant value: **12**

DATA_TYPE_AS_OBJECT

```
public static final byte DATA_TYPE_AS_OBJECT
```

Constant value: **13**

DATA_TYPE_RECORDSET

```
public static final byte DATA_TYPE_RECORDSET
```

Constant value: **14**

DATA_TYPE_XML

```
public static final byte DATA_TYPE_XML
```

Constant value: **15**

DATA_TYPE_CUSTOM_CLASS

```
public static final byte DATA_TYPE_CUSTOM_CLASS
```

Constant value: **16**

DATA_TYPE_AMF3

```
public static final byte DATA_TYPE_AMF3
```

Constant value: **17**

DATA_TYPE_INTEGER

```
public static final byte DATA_TYPE_INTEGER
```

Constant value: **32**

DATA_TYPE_BYTEARRAY

```
public static final byte DATA_TYPE_BYTEARRAY
```

(continued from last page)

Constant value: **33**

DATA_TYPE_XML_TOP

```
public static final byte DATA_TYPE_XML_TOP
```

Constant value: **34**

DATA_TYPE_AMF3_UNDEFINED

```
public static final byte DATA_TYPE_AMF3_UNDEFINED
```

Constant value: **0**

DATA_TYPE_AMF3_NULL

```
public static final byte DATA_TYPE_AMF3_NULL
```

Constant value: **1**

DATA_TYPE_AMF3_BOOLEAN_FALSE

```
public static final byte DATA_TYPE_AMF3_BOOLEAN_FALSE
```

Constant value: **2**

DATA_TYPE_AMF3_BOOLEAN_TRUE

```
public static final byte DATA_TYPE_AMF3_BOOLEAN_TRUE
```

Constant value: **3**

DATA_TYPE_AMF3_INTEGER

```
public static final byte DATA_TYPE_AMF3_INTEGER
```

Constant value: **4**

DATA_TYPE_AMF3_NUMBER

```
public static final byte DATA_TYPE_AMF3_NUMBER
```

Constant value: **5**

DATA_TYPE_AMF3_STRING

```
public static final byte DATA_TYPE_AMF3_STRING
```

Constant value: **6**

DATA_TYPE_AMF3_XML_LEGACY

public static final byte **DATA_TYPE_AMF3_XML_LEGACY**

Constant value: **7**

DATA_TYPE_AMF3_DATE

public static final byte **DATA_TYPE_AMF3_DATE**

Constant value: **8**

DATA_TYPE_AMF3_ARRAY

public static final byte **DATA_TYPE_AMF3_ARRAY**

Constant value: **9**

DATA_TYPE_AMF3_OBJECT

public static final byte **DATA_TYPE_AMF3_OBJECT**

Constant value: **10**

DATA_TYPE_AMF3_XML_TOP

public static final byte **DATA_TYPE_AMF3_XML_TOP**

Constant value: **11**

DATA_TYPE_AMF3_BYTEARRAY

public static final byte **DATA_TYPE_AMF3_BYTEARRAY**

Constant value: **12**

AMF_LEVEL0

public static final byte **AMF_LEVEL0**

Constant value: **0**

AMF_LEVEL3

public static final byte **AMF_LEVEL3**

Constant value: **3**

(continued from last page)

MILLS_PER_HOUR

```
public static final int MILLS_PER_HOUR
```

Number of milliseconds in an hour
Constant value: **60000**

type

```
protected int type
```

Constructors

AMFData

```
public AMFData()
```

Methods

getType

```
public int getType()
```

Returns the data type for this object

Returns:

object type DATA_TYPE_*

setType

```
public void setType(int type)
```

Sets the the data type for this object

Parameters:

type - type DATA_TYPE_*

testNextByte

```
public static boolean testNextByte(java.nio.ByteBuffer data,  
    int test)
```

Peek at the next value in data to see if its the test value

Parameters:

data - binary data being deserialized
test - value being tested

Returns:

return true if the next byte in the buffer equals the test value

peekByte

```
public static int peekByte(java.nio.ByteBuffer data)
```


(continued from last page)

Return the next byte in the buffer without incrementing the data position

Parameters:

data - binary data being deserialized

Returns:

next byte in buffer

skipByte

```
public static int skipByte(java.nio.ByteBuffer data)
```

Skip forward one byte in the byte buffer

Parameters:

data - binary data being deserialized

Returns:

next byte in buffer

isObjStart

```
public static boolean isObjStart(java.nio.ByteBuffer data,  
    AMFDataContextDeserialize context)
```

Returns true if next byte in data is object start

Parameters:

data - binary data being deserialized
context - deserialization context

Returns:

Returns true if next byte in data is object start

isAMF3Start

```
public static boolean isAMF3Start(java.nio.ByteBuffer data,  
    AMFDataContextDeserialize context)
```

Returns true if next byte in data is AMF3 start

Parameters:

data - binary data being deserialized
context - deserialization context

Returns:

Returns true if next byte in data is object start

isArrayStart

```
public static boolean isArrayStart(java.nio.ByteBuffer data,  
    AMFDataContextDeserialize context)
```

Returns true if next byte in data is an array start

Parameters:

data - binary data being deserialized
context - deserialization context

(continued from last page)

Returns:

Returns true if next byte in data is and array start

isMixedArrayStart

```
public static boolean isMixedArrayStart(java.nio.ByteBuffer data,  
    AMFDataContextDeserialize context)
```

Returns true if next byte in data is mixed array start

Parameters:

data - binary data being deserialized
context - deserialization context

Returns:

Returns true if next byte in data is mixed array start

isObjEnd

```
public static boolean isObjEnd(java.nio.ByteBuffer data,  
    AMFDataContextDeserialize context)
```

Returns true if next byte is object end

Parameters:

data - binary data being deserialized
context - deserialization context

Returns:

Returns true if next byte in object end

isByteArrayStart

```
public static boolean isByteArrayStart(java.nio.ByteBuffer data,  
    AMFDataContextDeserialize context)
```

Returns true if next byte is ByteArray start

Parameters:

data - binary data being deserialized
context - deserialization context

Returns:

true if next byte in object end

getReference

```
public static AMFData getReference(java.nio.ByteBuffer data,  
    AMFDataContextDeserialize context)
```

Get AMF3 object reference

Parameters:

data - binary data being deserialized
context - deserialization context

Returns:

null if not a reference else referenced object

deserializeInnerObject

```
public static AMFData deserializeInnerObject( java.nio.ByteBuffer data,  
        AMFDataContextDeserialize context)  
    throws java.io.IOException
```

Deserialize next item

Parameters:

data - binary data being deserialized
context - deserialization context

Returns:

deserialized object

Throws:

IOException

createContextSerialize

```
public static AMFDataContextSerialize createContextSerialize()
```

Create an AMF3 serialization context

Returns:

AMF3 serialization context

createContextSerialize

```
public static AMFDataContextSerialize createContextSerialize(int objectEncoding)
```

Create an AMF3 serialization context

Parameters:

objectEncoding - object encoding level (see AMF_LEVEL*)

Returns:

AMF3 serialization context

createContextDeserialize

```
public static AMFDataContextDeserialize createContextDeserialize()
```

Create an AMF3 deserialization context

Returns:

AMF3 deserialization context

createContextDeserialize

```
public static AMFDataContextDeserialize createContextDeserialize(int objectEncoding)
```

Create an AMF3 deserialization context

Parameters:

objectEncoding - object encoding level (see AMF_LEVEL*)

Returns:

(continued from last page)

AMF3 deserialization context

serialize

```
public abstract void serialize(java.io.DataOutputStream out)
```

Serialize object to output stream

Parameters:

out - Output stream

serialize

```
public abstract void serialize(java.io.DataOutputStream out,  
    int objectEncoding)
```

Serialize object to output stream

Parameters:

out - Output stream

objectEncoding - object encoding level (see AMF_LEVEL*)

serialize

```
public abstract void serialize(java.io.DataOutputStream out,  
    AMFDataContextSerialize context)
```

Serialize object to output stream

Parameters:

out - Output stream

context - serialization context used by AMF3

serialize

```
public abstract byte[] serialize()
```

Serial object to byte array

Returns:

serialized byte array

serialize

```
public abstract byte[] serialize(int objectEncoding)
```

Serial object to byte array

Parameters:

objectEncoding - object encoding level (see AMF_LEVEL*)

Returns:

serialized byte array

serialize

```
public abstract byte[] serialize(AMFDataContextSerialize context)
```

Serial object to byte array

(continued from last page)

Parameters:

context - serialization context used by AMF3

Returns:

serialized byte array

deserialize

```
public abstract void deserialize(java.nio.ByteBuffer data)
```

Deserialize data in byte buffer

Parameters:

data - binary data

deserialize

```
public abstract void deserialize(java.nio.ByteBuffer data,  
    AMFDataContextDeserialize context)
```

Deserialize data in byte buffer

Parameters:

data - binary data

context - deserialization context used by AMF3

getValue

```
public abstract Object getValue()
```

Convert object to Java native class

Returns:

java native class

triggerAMF3Switch

```
public static boolean triggerAMF3Switch(AMFData data)
```

Return true if the object is serialized differently in AMF3

Parameters:

data - AMF object

Returns:

true if the object is serialized differently in AMF3

com.wowza.wms.amf Class AMFDataArray

```
java.lang.Object
├── com.wowza.wms.amf.AMFData
│   └── com.wowza.wms.amf.AMFDataArray
```

```
public class AMFDataArray
extends AMFData
```

AMFDataArray: class for marshalling data between Wowza Pro server and Flash client. This class is a simple ordered array of items.

Create Array of Strings

```
AMFDataArray amfDataArray = new AMFDataArray();

amfDataArray.add("item1");
amfDataArray.add("item2");
amfDataArray.add("item3");
```

Iterate Items In Array

```
AMFDataArray amfDataArray;

for(int i=0;i<amfDataArray.size();i++)
{
    AMFData amfData = amfDataArray.get(i);
    WMSLoggerFactory.getLogger(null).debug("amfData.getType(): "+amfData.getType());
}
```

NOTE: There is a slight difference between this class and AMFDataList. This class when serialized/deserialized does include the DATA_TYPE_ARRAY header (byte) and array size (int).

NOTE: Simple arrays created in the Flash player client and sent to the Wowza Pro server are of type [AMFDataMixedArray](#).

Fields inherited from class [com.wowza.wms.amf.AMFData](#)

[AMF_LEVEL0](#), [AMF_LEVEL3](#), [DATA_TYPE_AMF3](#), [DATA_TYPE_AMF3_ARRAY](#), [DATA_TYPE_AMF3_BOOLEAN_FALSE](#), [DATA_TYPE_AMF3_BOOLEAN_TRUE](#), [DATA_TYPE_AMF3_BYTEARRAY](#), [DATA_TYPE_AMF3_DATE](#), [DATA_TYPE_AMF3_INTEGER](#), [DATA_TYPE_AMF3_NULL](#), [DATA_TYPE_AMF3_NUMBER](#), [DATA_TYPE_AMF3_OBJECT](#), [DATA_TYPE_AMF3_STRING](#), [DATA_TYPE_AMF3_UNDEFINED](#), [DATA_TYPE_AMF3_XML_LEGACY](#), [DATA_TYPE_AMF3_XML_TOP](#), [DATA_TYPE_ARRAY](#), [DATA_TYPE_AS_OBJECT](#), [DATA_TYPE_BOOLEAN](#), [DATA_TYPE_BYTEARRAY](#), [DATA_TYPE_CUSTOM_CLASS](#), [DATA_TYPE_DATE](#), [DATA_TYPE_INTEGER](#), [DATA_TYPE_LONG_STRING](#), [DATA_TYPE_MIXED_ARRAY](#), [DATA_TYPE_MOVIE_CLIP](#), [DATA_TYPE_NULL](#), [DATA_TYPE_NUMBER](#), [DATA_TYPE_OBJECT](#), [DATA_TYPE_OBJECT_END](#), [DATA_TYPE_RECORDSET](#), [DATA_TYPE_REFERENCE_OBJECT](#), [DATA_TYPE_STRING](#), [DATA_TYPE_UNDEFINED](#), [DATA_TYPE_UNKNOWN](#), [DATA_TYPE_XML](#), [DATA_TYPE_XML_TOP](#), [MILLS_PER_HOUR](#), [type](#)

Constructor Summary

public	AMFDataArray () Create empty AMFDataArray object
public	AMFDataArray (byte[] data) Deserialize entire data array and create AMFDataArray object
public	AMFDataArray (byte[] data, int offset, int size) Deserialize data array starting at offset for size bytes and create AMFDataArray object
public	AMFDataArray (java.nio.ByteBuffer data) Deserialize entire data array and create AMFDataArray object
public	AMFDataArray (java.nio.ByteBuffer data, AMFDataContextDeserialize context)

Method Summary

void	add (AMFData data) Append a new item onto the array
void	add (boolean data) Append a boolean (will be wrapped in an AMFDataItem object)
void	add (java.util.Date data) Append a date (will be wrapped in an AMFDataItem object)
void	add (double data) Append a double (will be wrapped in an AMFDataItem object)
void	add (int data) Append a int (will be wrapped in an AMFDataItem object)
void	add (int index, AMFData data) Insert an item into the array
void	add (int index, boolean data) Insert a boolean value (will be wrapped in an AMFDataItem object)
void	add (int index, java.util.Date data) Insert a date value (will be wrapped in an AMFDataItem object)
void	add (int index, double data) Insert a double value (will be wrapped in an AMFDataItem object)

void	add (int index, int data) Insert a int value (will be wrapped in an AMFDataItem object)
void	add (int index, long data) Insert a long value (will be wrapped in an AMFDataItem object)
void	add (int index, String data) Insert a string value (will be wrapped in an AMFDataItem object)
void	add (long data) Append a long (will be wrapped in an AMFDataItem object)
void	add (String data) Append a string (will be wrapped in an AMFDataItem object)
void	deserialize (java.nio.ByteBuffer data)
void	deserialize (java.nio.ByteBuffer data, AMFDataContextDeserialize context)
AMFData	get (int index) Get item at index
boolean	getBoolean (int index) Get item at index return as boolean
byte	getBytes (int index) Get item at index return as byte
java.util.Date	getDate (int index) Get item at index return as Date
double	getDouble (int index) Get item at index return as double
float	getFloat (int index) Get item at index return as float
int	getInt (int index) Get item at index return as int
long	getLong (int index) Get item at index return as long
AMFDataObj	getObject (int index) Get item at index return as AMFDataObj
short	getShort (int index) Get item at index return as short
String	getString (int index) Get item at index return as String
Object	getValue ()
AMFData	remove (int index) Remove an item from the array

byte[]	<u>serialize()</u>
byte[]	<u>serialize()</u> (<u>AMFDataContextSerialize</u> context)
void	<u>serialize()</u> (java.io.DataOutputStream out)
void	<u>serialize()</u> (java.io.DataOutputStream out, <u>AMFDataContextSerialize</u> context)
void	<u>serialize()</u> (java.io.DataOutputStream out, int objectEncoding)
byte[]	<u>serialize()</u> (int objectEncoding)
void	<u>set()</u> (int index, <u>AMFData</u> data) Set an array item
void	<u>set()</u> (int index, boolean data) Set an boolean value (will be wrapped in an AMFDataItem object)
void	<u>set()</u> (int index, java.util.Date data) Set an date value (will be wrapped in an AMFDataItem object)
void	<u>set()</u> (int index, double data) Set an double value (will be wrapped in an AMFDataItem object)
void	<u>set()</u> (int index, int data) Set an int value (will be wrapped in an AMFDataItem object)
void	<u>set()</u> (int index, long data) Set an long value (will be wrapped in an AMFDataItem object)
void	<u>set()</u> (int index, String data) Set an string value (will be wrapped in an AMFDataItem object)
int	<u>size()</u> Returns the number of items in array
String	<u>toString()</u> Return object as formatted string

Methods inherited from class [com.wowza.wms.amf.AMFData](#)

[createContextDeserialize](#), [createContextDeserialize](#), [createContextSerialize](#), [createContextSerialize](#), [deserialize](#), [deserialize](#), [deserializeInnerObject](#), [getReference](#), [getType](#), [getValue](#), [isAMF3Start](#), [isArrayStart](#), [isByteArrayStart](#), [isMixedArrayStart](#), [isObjEnd](#), [isObjStart](#), [peekByte](#), [serialize](#), [serialize](#), [serialize](#), [serialize](#), [serialize](#), [serialize](#), [setType](#), [skipByte](#), [testNextByte](#), [triggerAMF3Switch](#)

Methods inherited from class java.lang.Object

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

Constructors

(continued from last page)

AMFDataArray

```
public AMFDataArray()
```

Create empty AMFDataArray object

AMFDataArray

```
public AMFDataArray(byte[] data)
```

Deserialize entire data array and create AMFDataArray object

Parameters:

data - binary data

AMFDataArray

```
public AMFDataArray(byte[] data,  
                    int offset,  
                    int size)
```

Deserialize data array starting at offset for size bytes and create AMFDataArray object

Parameters:

data - binary data

offset - starting offset into data

size - size of data to deserialize

AMFDataArray

```
public AMFDataArray(java.nio.ByteBuffer data)
```

Deserialize entire data array and create AMFDataArray object

Parameters:

data - binary data

AMFDataArray

```
public AMFDataArray(java.nio.ByteBuffer data,  
                   AMFDataContextDeserialize context)
```

Methods

remove

```
public AMFData remove(int index)
```

Remove an item from the array

Parameters:

index - index

Returns:

delete item or null if not found

size

```
public int size()
```

Returns the number of items in array

Returns:

number of items in array

add

```
public void add(AMFData data)
```

Append a new item onto the array

Parameters:

data - AMFData object

add

```
public void add(String data)
```

Append a string (will be wrapped in an AMFDataItem object)

Parameters:

data - string value

add

```
public void add(double data)
```

Append a double (will be wrapped in an AMFDataItem object)

Parameters:

data - double value

add

```
public void add(int data)
```

Append a int (will be wrapped in an AMFDataItem object)

Parameters:

data - int value

add

```
public void add(long data)
```

Append a long (will be wrapped in an AMFDataItem object)

Parameters:

data - long value

add

```
public void add(java.util.Date data)
```

(continued from last page)

Append a date (will be wrapped in an AMFDataItem object)

Parameters:

data - date value

add

```
public void add(boolean data)
```

Append a boolean (will be wrapped in an AMFDataItem object)

Parameters:

data - boolean value

add

```
public void add(int index,  
    AMFData data)
```

Insert an item into the array

Parameters:

index - index

data - AMFData object

add

```
public void add(int index,  
    String data)
```

Insert a string value (will be wrapped in an AMFDataItem object)

Parameters:

index - index

data - string value

add

```
public void add(int index,  
    double data)
```

Insert a double value (will be wrapped in an AMFDataItem object)

Parameters:

index - index

data - double value

add

```
public void add(int index,  
    int data)
```

Insert a int value (will be wrapped in an AMFDataItem object)

Parameters:

index - index

data - int value

add

```
public void add(int index,  
                long data)
```

Insert a long value (will be wrapped in an AMFDataItem object)

Parameters:

index - index
data - long value

add

```
public void add(int index,  
                java.util.Date data)
```

Insert a date value (will be wrapped in an AMFDataItem object)

Parameters:

index - index
data - date value

add

```
public void add(int index,  
                boolean data)
```

Insert a boolean value (will be wrapped in an AMFDataItem object)

Parameters:

index - index
data - boolean value

set

```
public void set(int index,  
                AMFData data)
```

Set an array item

Parameters:

index - index
data - AMFData object

set

```
public void set(int index,  
                String data)
```

Set an string value (will be wrapped in an AMFDataItem object)

Parameters:

index - index
data - string value

set

```
public void set(int index,  
                double data)
```

(continued from last page)

Set an double value (will be wrapped in an AMFDataItem object)

Parameters:

index - index
data - double value

set

```
public void set(int index,  
                int data)
```

Set an int value (will be wrapped in an AMFDataItem object)

Parameters:

index - index
data - int value

set

```
public void set(int index,  
                long data)
```

Set an long value (will be wrapped in an AMFDataItem object)

Parameters:

index - index
data - long value

set

```
public void set(int index,  
                java.util.Date data)
```

Set an date value (will be wrapped in an AMFDataItem object)

Parameters:

index - index
data - date value

set

```
public void set(int index,  
                boolean data)
```

Set an boolean value (will be wrapped in an AMFDataItem object)

Parameters:

index - index
data - boolean value

get

```
public AMFData get(int index)
```

Get item at index

Parameters:

index

(continued from last page)

Returns:

Returns AMFData object or null if out of bounds

getString

```
public String getString(int index)
```

Get item at index return as String

Parameters:

index

Returns:

Return item as String or null if out of bounds

getInt

```
public int getInt(int index)
```

Get item at index return as int

Parameters:

index

Returns:

Return item as int or 0 if out of bounds

getLong

```
public long getLong(int index)
```

Get item at index return as long

Parameters:

index

Returns:

Return item as long or 0 if out of bounds

getShort

```
public short getShort(int index)
```

Get item at index return as short

Parameters:

index

Returns:

Return item as short or 0 if out of bounds

getByte

```
public byte getByte(int index)
```

Get item at index return as byte

Parameters:

(continued from last page)

index

Returns:

Return item as byte or 0 if out of bounds

getBoolean

```
public boolean getBoolean(int index)
```

Get item at index return as boolean

Parameters:

index

Returns:

Return item as boolean or false if out of bounds

getDate

```
public java.util.Date getDate(int index)
```

Get item at index return as Date

Parameters:

index

Returns:

Return item as Date or null if out of bounds

getObject

```
public AMFDataObj getObject(int index)
```

Get item at index return as AMFDataObj

Parameters:

index

Returns:

Return item as AMFDataObj or null if out of bounds

getDouble

```
public double getDouble(int index)
```

Get item at index return as double

Parameters:

index

Returns:

Return item as double or 0 if out of bounds

getFloat

```
public float getFloat(int index)
```

Get item at index return as float

(continued from last page)

Parameters:

index

Returns:

Return item as float or 0 if out of bounds

deserialize

```
public void deserialize(java.nio.ByteBuffer data)
```

Deserialize data in byte buffer

deserialize

```
public void deserialize(java.nio.ByteBuffer data,  
    AMFDataContextDeserialize context)
```

Deserialize data in byte buffer

serialize

```
public void serialize(java.io.DataOutputStream out)
```

Serialize object to output stream

serialize

```
public void serialize(java.io.DataOutputStream out,  
    int objectEncoding)
```

Serialize object to output stream

serialize

```
public void serialize(java.io.DataOutputStream out,  
    AMFDataContextSerialize context)
```

Serialize object to output stream

serialize

```
public byte[] serialize()
```

Serial object to byte array

serialize

```
public byte[] serialize(int objectEncoding)
```

Serial object to byte array

serialize

```
public byte[] serialize(AMFDataContextSerialize context)
```

Serial object to byte array

(continued from last page)

getValue

```
public Object getValue()
```

Convert object to Java native class

toString

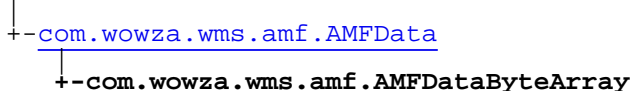
```
public String toString()
```

Return object as formatted string

com.wowza.wms.amf

Class AMFDataByteArray

java.lang.Object



public class **AMFDataByteArray**
 extends [AMFData](#)

AMFDataByteArray: class for marshalling data between Wowza Pro server and Flash client. This class is a simple byte array.

Fields inherited from class [com.wowza.wms.amf.AMFData](#)

[AMF_LEVEL0](#), [AMF_LEVEL3](#), [DATA_TYPE_AMF3](#), [DATA_TYPE_AMF3_ARRAY](#), [DATA_TYPE_AMF3_BOOLEAN_FALSE](#), [DATA_TYPE_AMF3_BOOLEAN_TRUE](#), [DATA_TYPE_AMF3_BYTEARRAY](#), [DATA_TYPE_AMF3_DATE](#), [DATA_TYPE_AMF3_INTEGER](#), [DATA_TYPE_AMF3_NULL](#), [DATA_TYPE_AMF3_NUMBER](#), [DATA_TYPE_AMF3_OBJECT](#), [DATA_TYPE_AMF3_STRING](#), [DATA_TYPE_AMF3_UNDEFINED](#), [DATA_TYPE_AMF3_XML_LEGACY](#), [DATA_TYPE_AMF3_XML_TOP](#), [DATA_TYPE_ARRAY](#), [DATA_TYPE_AS_OBJECT](#), [DATA_TYPE_BOOLEAN](#), [DATA_TYPE_BYTEARRAY](#), [DATA_TYPE_CUSTOM_CLASS](#), [DATA_TYPE_DATE](#), [DATA_TYPE_INTEGER](#), [DATA_TYPE_LONG_STRING](#), [DATA_TYPE_MIXED_ARRAY](#), [DATA_TYPE_MOVIE_CLIP](#), [DATA_TYPE_NULL](#), [DATA_TYPE_NUMBER](#), [DATA_TYPE_OBJECT](#), [DATA_TYPE_OBJECT_END](#), [DATA_TYPE_RECORDSET](#), [DATA_TYPE_REFERENCE_OBJECT](#), [DATA_TYPE_STRING](#), [DATA_TYPE_UNDEFINED](#), [DATA_TYPE_UNKNOWN](#), [DATA_TYPE_XML](#), [DATA_TYPE_XML_TOP](#), [MILLS_PER_HOUR](#), [type](#)

Constructor Summary

public	AMFDataByteArray() Create empty AMFDataByteArray object
public	AMFDataByteArray(byte[] data) Deserialize entire byte array and create AMFDataByteArray object.
public	AMFDataByteArray(byte[] data, int offset, int size) Deserialize data array starting at offset for size bytes and create AMFDataByteArray object.
public	AMFDataByteArray(java.nio.ByteBuffer data) Deserialize entire data array and create AMFDataByteArray object.
public	AMFDataByteArray(java.nio.ByteBuffer data, AMFDataContextDeserialize context) Deserialize entire data array and create AMFDataByteArray object.

Method Summary

int	compress() Compress the internal buffer using the ZLIB compression library
int	decompress() Decompress the internal buffer using the ZLIB compression library
void	deserialize(java.nio.ByteBuffer data)

void	deserialize (java.nio.ByteBuffer data, AMFDataContextDeserialize context)
Object	getValue () Returns the underlying byte[] data buffer
byte[]	serialize ()
byte[]	serialize (AMFDataContextSerialize context)
void	serialize (java.io.DataOutputStream out)
void	serialize (java.io.DataOutputStream out, AMFDataContextSerialize context)
void	serialize (java.io.DataOutputStream out, int objectEncoding)
byte[]	serialize (int objectEncoding)
int	size () Returns the number of bytes in the byte array
byte[]	toArray () Returns the underlying data buffer (not a copy)
java.nio.ByteBuffer	toByteBuffer () Wraps the underlying data buffer with a ByteBuffer object.
String	toString () Return object as formatted string
static AMFDataByteArray	wrap (byte[] data) Wraps a byte[] into a AMFDataByteArray.
static AMFDataByteArray	wrap (java.nio.ByteBuffer data) Wraps a ByteBuffer into a AMFDataByteArray.

Methods inherited from class [com.wowza.wms.amf.AMFData](#)

[createContextDeserialize](#), [createContextDeserialize](#), [createContextSerialize](#), [createContextSerialize](#), [deserialize](#), [deserialize](#), [deserializeInnerObject](#), [getReference](#), [getType](#), [getValue](#), [isAMF3Start](#), [isArrayStart](#), [isByteArrayStart](#), [isMixedArrayStart](#), [isObjEnd](#), [isObjStart](#), [peekByte](#), [serialize](#), [serialize](#), [serialize](#), [serialize](#), [setType](#), [skipByte](#), [testNextByte](#), [triggerAMF3Switch](#)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

(continued from last page)

AMFDataByteArray

```
public AMFDataByteArray( )
```

Create empty AMFDataByteArray object

AMFDataByteArray

```
public AMFDataByteArray(byte[] data)
```

Deserialize entire byte array and create AMFDataByteArray object. Note: This is for AMF3 deserialization and cannot be used to create an AMFDataByteArray with the passed in data. To create a new AMFDataByteArray from either a byte[] or ByteBuffer, use the static wrap method.

Parameters:

data - binary data

AMFDataByteArray

```
public AMFDataByteArray(byte[] data,  
                        int offset,  
                        int size)
```

Deserialize data array starting at offset for size bytes and create AMFDataByteArray object. Note: This is for AMF3 deserialization and cannot be used to create an AMFDataByteArray with the passed in data. To create a new AMFDataByteArray from either a byte[] or ByteBuffer, use the static wrap method.

Parameters:

data - binary data

offset - starting offset into data

size - size of data to deserialize

AMFDataByteArray

```
public AMFDataByteArray( java.nio.ByteBuffer data)
```

Deserialize entire data array and create AMFDataByteArray object. Note: This is for AMF3 deserialization and cannot be used to create an AMFDataByteArray with the passed in data. To create a new AMFDataByteArray from either a byte[] or ByteBuffer, use the static wrap method.

Parameters:

data - binary data

AMFDataByteArray

```
public AMFDataByteArray( java.nio.ByteBuffer data,  
                        AMFDataContextDeserialize context)
```

Deserialize entire data array and create AMFDataByteArray object. Note: This is for AMF3 deserialization and cannot be used to create an AMFDataByteArray with the passed in data. To create a new AMFDataByteArray from either a byte[] or ByteBuffer, use the static wrap method.

Parameters:

data - binary data

context - deserialization context (used for AMF3 decoding)

Methods

(continued from last page)

size

```
public int size()
```

Returns the number of bytes in the byte array

Returns:

number of bytes in the array

toArray

```
public byte[] toArray()
```

Returns the underlying data buffer (not a copy)

Returns:

data buffer (not a copy)

toByteBuffer

```
public java.nio.ByteBuffer toByteBuffer()
```

Wraps the underlying data buffer with a ByteBuffer object.

Returns:

byte[] wrapped as ByteBuffer

wrap

```
public static AMFDataByteArray wrap(byte[] data)
```

Wraps a byte[] into a AMFDataByteArray. Note: This method does not copy the array.

Parameters:

data - byte[] data

Returns:

wrapped byte[]

wrap

```
public static AMFDataByteArray wrap(java.nio.ByteBuffer data)
```

Wraps a ByteBuffer into a AMFDataByteArray. Note: This method only copies the ByteBuffer data if the ByteBuffer.array() method fails.

Parameters:

data - ByteBuffer data

Returns:

wrapped ByteBuffer

getValue

```
public Object getValue()
```

Returns the underlying byte[] data buffer

deserialize

```
public void deserialize(java.nio.ByteBuffer data)
```

Deserialize data in byte buffer

deserialize

```
public void deserialize(java.nio.ByteBuffer data,  
    AMFDataContextDeserialize context)
```

Deserialize data in byte buffer

serialize

```
public void serialize(java.io.DataOutputStream out)
```

Serialize object to output stream

serialize

```
public void serialize(java.io.DataOutputStream out,  
    int objectEncoding)
```

Serialize object to output stream

serialize

```
public void serialize(java.io.DataOutputStream out,  
    AMFDataContextSerialize context)
```

Serialize object to output stream

serialize

```
public byte[] serialize()
```

Serial object to byte array

serialize

```
public byte[] serialize(int objectEncoding)
```

Serial object to byte array

serialize

```
public byte[] serialize(AMFDataContextSerialize context)
```

Serial object to byte array

compress

```
public int compress()
```

Compress the internal buffer using the ZLIB compression library

Returns:

(continued from last page)

size of compressed buffer

decompress

```
public int decompress()
```

Decompress the internal buffer using the ZLIB compression library

Returns:

size of decompressed buffer

toString

```
public String toString()
```

Return object as formatted string

com.wowza.wms.amf

Class AMFDataContextDeserialize

java.lang.Object

└─com.wowza.wms.amf.AMFDataContextDeserialize

public class **AMFDataContextDeserialize**
extends Object

AMF context used for deserialization

Constructor Summary

public	AMFDataContextDeserialize() Constructor
public	AMFDataContextDeserialize(int objectEncoding) Constructor with encoding

Method Summary

void	addObject(Object obj) Add an object to the object cache
void	addString(String str) Add a string to the string cache
void	addTrait(AMFDataTrait obj) Add a trait to the trait cache
int	clearIntData() Internal use, get and clear int data
int	getIntData() Internal use, get int data
Object	getObject(int index) Get an object from the object cache
int	getObjectEncoding() Get object encoding, see AMFData.AMF_LEVEL*
String	getString(int index) Get a string from the string cache
AMFDataTrait	getTrait(int index) Get a trait from the trait cache
boolean	isAMF0() Is context AMF0
boolean	isAMF3() Is context AMF3

boolean	<code>isIntData()</code> Internal use, get int data
void	<code>setIntData(int intData)</code> Internal use, set int data
void	<code>setObjectEncoding(int objectEncoding)</code> Set object encoding, see <code>AMFData.AMF_LEVEL*</code>

Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Constructors

AMFDataContextDeserialize

```
public AMFDataContextDeserialize()
```

Constructor

AMFDataContextDeserialize

```
public AMFDataContextDeserialize(int objectEncoding)
```

Constructor with encoding

Parameters:

`objectEncoding` - object encoding, see `AMFData.AMF_LEVEL*`

Methods

isIntData

```
public boolean isIntData()
```

Internal use, get int data

Returns:

returns true if pending int data

setIntData

```
public void setIntData(int intData)
```

Internal use, set int data

Parameters:

`intData` - int data

getIntData

```
public int getIntData()
```

Internal use, get int data

(continued from last page)

Returns:

int data

clearIntData

```
public int clearIntData()
```

Internal use, get and clear int data

Returns:

int data

getObjectEncoding

```
public int getObjectEncoding()
```

Get object encoding, see AMFData.AMF_LEVEL*

Returns:

object encoding, see AMFData.AMF_LEVEL*

setObjectEncoding

```
public void setObjectEncoding(int objectEncoding)
```

Set object encoding, see AMFData.AMF_LEVEL*

Parameters:

objectEncoding - object encoding, see AMFData.AMF_LEVEL*

isAMF3

```
public boolean isAMF3()
```

Is context AMF3

Returns:

true, if AMF3

isAMF0

```
public boolean isAMF0()
```

Is context AMF0

Returns:

true, if AMF0

addString

```
public void addString(String str)
```

Add a string to the string cache

Parameters:

str - string value

getString

```
public String getString(int index)  
    throws IndexOutOfBoundsException
```

Get a string from the string cache

Parameters:

index - index

Returns:

string value

Throws:

IndexOutOfBoundsException

addObject

```
public void addObject(Object obj)
```

Add an object to the object cache

Parameters:

obj - object value

getObject

```
public Object getObject(int index)  
    throws IndexOutOfBoundsException
```

Get an object from the object cache

Parameters:

index - index

Returns:

object value

Throws:

IndexOutOfBoundsException

addTrait

```
public void addTrait(AMFDataTrait obj)
```

Add a trait to the trait cache

Parameters:

obj - trait object

getTrait

```
public AMFDataTrait getTrait(int index)  
    throws IndexOutOfBoundsException
```

Get a trait from the trait cache

Parameters:

(continued from last page)

index - index

Returns:

trait object

Throws:

IndexOutOfBoundsException

com.wowza.wms.amf

Class AMFDataContextSerialize

java.lang.Object

└─com.wowza.wms.amf.AMFDataContextSerialize

public class **AMFDataContextSerialize**
extends Object

AMF context used for serialization

Constructor Summary

public	AMFDataContextSerialize() Constructor
public	AMFDataContextSerialize(int objectEncoding) Constructor with object encoding, see AMFData.AMF_LEVEL*

Method Summary

int	getObjectEncoding() Get object encoding, see AMFData.AMF_LEVEL*
int	getObjectReference(Object obj) Get index of object in object cache
int	getStringReference(String str) Get index of string item in string cache
int	getTargetEncoding() Get target encoding, see AMFData.AMF_LEVEL*
int	getTraitReference(AMFDataTrait obj) Get index of trait object in trait cache
boolean	isAMF0() Is context AMF0
boolean	isAMF3() Is context AMF3
void	setObjectEncoding(int objectEncoding) Set object encoding, see AMFData.AMF_LEVEL*
void	setTargetEncoding(int targetEncoding) Set target encoding, , see AMFData.AMF_LEVEL*
void	writeString(java.io.DataOutputStream out, String str) Write a stream to the output buffer

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

AMFDataContextSerialize

```
public AMFDataContextSerialize()
```

Constructor

AMFDataContextSerialize

```
public AMFDataContextSerialize(int objectEncoding)
```

Constructor with object encoding, see AMFData.AMF_LEVEL*

Parameters:

objectEncoding - object encoding, see AMFData.AMF_LEVEL*

Methods

getObjectEncoding

```
public int getObjectEncoding()
```

Get object encoding, see AMFData.AMF_LEVEL*

Returns:

object encoding, see AMFData.AMF_LEVEL*

setObjectEncoding

```
public void setObjectEncoding(int objectEncoding)
```

Set object encoding, see AMFData.AMF_LEVEL*

Parameters:

objectEncoding - object encoding, see AMFData.AMF_LEVEL*

getTargetEncoding

```
public int getTargetEncoding()
```

Get target encoding, see AMFData.AMF_LEVEL*

Returns:

target encoding, see AMFData.AMF_LEVEL*

setTargetEncoding

```
public void setTargetEncoding(int targetEncoding)
```

Set target encoding, , see AMFData.AMF_LEVEL*

(continued from last page)

Parameters:

targetEncoding - target encoding, , see AMFData.AMF_LEVEL*

isAMF3

```
public boolean isAMF3()
```

Is context AMF3

Returns:

true, if AMF3

isAMF0

```
public boolean isAMF0()
```

Is context AMF0

Returns:

true, if AMF0

getStringReference

```
public int getStringReference(String str)
```

Get index of string item in string cache

Parameters:

str - string value

Returns:

index

getObjectReference

```
public int getObjectReference(Object obj)
```

Get index of object in object cache

Parameters:

obj - object value

Returns:

index

getTraitReference

```
public int getTraitReference(AMFDataTrait obj)
```

Get index of trait object in trait cache

Parameters:

obj - trait object

Returns:

index

(continued from last page)

writeString

```
public void writeString(java.io.DataOutputStream out,  
    String str)
```

Write a stream to the output buffer

Parameters:

out - output buffer

str - string value

com.wowza.wms.amf

Class AMFDataItem

```
java.lang.Object
├── com.wowza.wms.amf.AMFData
│   └── com.wowza.wms.amf.AMFDataItem
```

```
public class AMFDataItem
    extends AMFData
```

AMFDataItem: class for marshalling data between Wowza Pro server and Flash client. The type wraps native Java data types.

- DATA_TYPE_NUMBER = int, long, short, double, float
- DATA_TYPE_STRING = String
- DATA_TYPE_BOOLEAN = boolean
- DATA_TYPE_DATE = Date
- DATA_TYPE_NULL = 'null'

Create Native Java Types

```
AMFDataItem amfDataString = new AMFDataItem("here is my string"); // String
AMFDataItem amfDataLong = new AMFDataItem(1234L); // long
AMFDataItem amfDataDouble = new AMFDataItem(1.234); // double
AMFDataItem amfDataBoolean = new AMFDataItem(true); // boolean
AMFDataItem amfDataNull = new AMFDataItem(); // null
```

Get Native Java Types

```
String dataString = amfDataString.getType()==AMFData.DATA_TYPE_LONG_STRING?
    amfDataString.toString():"";

long dataLong = amfDataLong.getType()==AMFData.DATA_TYPE_NUMBER?
    amfDataLong.longValue():0;

double dataDouble = amfDataDouble.getType()==AMFData.DATA_TYPE_NUMBER?
    amfDataLong.doubleValue():0.0;

boolean dataBoolean = amfDataBoolean.getType()==AMFData.DATA_TYPE_BOOLEAN?
    amfDataLong.booleanValue():false;

Object dataNull = amfDataNull.getType()==AMFData.DATA_TYPE_NULL?
    null:null;
```

Field Summary

public static final	DATEFORMAT Value: EEE, dd MMM yyyy HH:mm:ss
protected	fastDateFormat

Fields inherited from class [com.wowza.wms.amf.AMFData](#)

[AMF_LEVEL0](#), [AMF_LEVEL3](#), [DATA_TYPE_AMF3](#), [DATA_TYPE_AMF3_ARRAY](#), [DATA_TYPE_AMF3_BOOLEAN_FALSE](#), [DATA_TYPE_AMF3_BOOLEAN_TRUE](#), [DATA_TYPE_AMF3_BYTEARRAY](#), [DATA_TYPE_AMF3_DATE](#), [DATA_TYPE_AMF3_INTEGER](#), [DATA_TYPE_AMF3_NULL](#), [DATA_TYPE_AMF3_NUMBER](#), [DATA_TYPE_AMF3_OBJECT](#), [DATA_TYPE_AMF3_STRING](#), [DATA_TYPE_AMF3_UNDEFINED](#), [DATA_TYPE_AMF3_XML_LEGACY](#), [DATA_TYPE_AMF3_XML_TOP](#), [DATA_TYPE_ARRAY](#), [DATA_TYPE_AS_OBJECT](#), [DATA_TYPE_BOOLEAN](#), [DATA_TYPE_BYTEARRAY](#), [DATA_TYPE_CUSTOM_CLASS](#), [DATA_TYPE_DATE](#), [DATA_TYPE_INTEGER](#), [DATA_TYPE_LONG_STRING](#), [DATA_TYPE_MIXED_ARRAY](#), [DATA_TYPE_MOVIE_CLIP](#), [DATA_TYPE_NULL](#), [DATA_TYPE_NUMBER](#), [DATA_TYPE_OBJECT](#), [DATA_TYPE_OBJECT_END](#), [DATA_TYPE_RECORDSET](#), [DATA_TYPE_REFERENCE_OBJECT](#), [DATA_TYPE_STRING](#), [DATA_TYPE_UNDEFINED](#), [DATA_TYPE_UNKNOWN](#), [DATA_TYPE_XML](#), [DATA_TYPE_XML_TOP](#), [MILLS_PER_HOUR](#), [type](#)

Constructor Summary

public	AMFDataItem () Construct AMF type DATA_TYPE_NULL object
public	AMFDataItem (String value) Construct AMF type DATA_TYPE_STRING object
public	AMFDataItem (int value) Construct AMF type DATA_TYPE_NUMBER object
public	AMFDataItem (long value) Construct AMF type DATA_TYPE_NUMBER object
public	AMFDataItem (double value) Construct AMF type DATA_TYPE_NUMBER object
public	AMFDataItem (boolean value) Construct AMF type DATA_TYPE_BOOLEAN
public	AMFDataItem (java.util.Date value) Construct AMF type DATA_TYPE_DATE
public	AMFDataItem (byte[] data) Deserialize entire data array and create AMFDataItem object
public	AMFDataItem (byte[] data, int offset, int size) Deserialize data array starting at offset for size bytes and create AMFDataItem object
public	AMFDataItem (java.nio.ByteBuffer data) Deserialize entire data array and create AMFDataItem object
public	AMFDataItem (java.nio.ByteBuffer data, AMFDataContextDeserialize context)

Method Summary

boolean	booleanValue() Return object as boolean.
byte	byteValue() Return object as byte.
java.util.Date	dateValue() Return object as Date.
void	deserialize() (java.nio.ByteBuffer data)
void	deserialize() (java.nio.ByteBuffer data, AMFDataContextDeserialize context)
double	doubleValue() Return object as double.
float	floatValue() Return object as float.
Object	getValue() Return value as Java class
int	intValue() Return object as int.
long	longValue() Return object as long.
byte[]	serialize()
byte[]	serialize() (AMFDataContextSerialize context)
void	serialize() (java.io.DataOutputStream out)
void	serialize() (java.io.DataOutputStream out, AMFDataContextSerialize context)
void	serialize() (java.io.DataOutputStream out, int objectEncoding)
byte[]	serialize() (int objectEncoding)
short	shortValue() Return object as short.
String	toString() Return object as formatted string

Methods inherited from class [com.wowza.wms.amf.AMFData](#)

[createContextDeserialize](#), [createContextDeserialize](#), [createContextSerialize](#), [createContextSerialize](#), [deserialize](#), [deserialize](#), [deserializeInnerObject](#), [getReference](#), [getType](#), [getValue](#), [isAMF3Start](#), [isArrayStart](#), [isByteArrayStart](#), [isMixedArrayStart](#), [isObjEnd](#), [isObjStart](#), [peekByte](#), [serialize](#), [serialize](#), [serialize](#), [serialize](#), [serialize](#), [setType](#), [skipByte](#), [testNextByte](#), [triggerAMF3Switch](#)

Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Fields

DATEFORMAT

`public static final java.lang.String` **DATEFORMAT**

Constant value: **EEE, dd MMM yyyy HH:mm:ss**

fastDateFormat

`protected org.apache.commons.lang.time.FastDateFormat` **fastDateFormat**

Constructors

AMFDataItem

`public` **AMFDataItem**()

Construct AMF type DATA_TYPE_NULL object

AMFDataItem

`public` **AMFDataItem**(String value)

Construct AMF type DATA_TYPE_STRING object

Parameters:

value - String value

AMFDataItem

`public` **AMFDataItem**(int value)

Construct AMF type DATA_TYPE_NUMBER object

Parameters:

value - int value

AMFDataItem

`public` **AMFDataItem**(long value)

(continued from last page)

Construct AMF type DATA_TYPE_NUMBER object

Parameters:

value - long value

AMFDataItem

```
public AMFDataItem(double value)
```

Construct AMF type DATA_TYPE_NUMBER object

Parameters:

value - double value

AMFDataItem

```
public AMFDataItem(boolean value)
```

Construct AMF type DATA_TYPE_BOOLEAN

Parameters:

value - boolean value

AMFDataItem

```
public AMFDataItem(java.util.Date value)
```

Construct AMF type DATA_TYPE_DATE

Parameters:

value - Date value

AMFDataItem

```
public AMFDataItem(byte[] data)
```

Deserialize entire data array and create AMFDataItem object

Parameters:

data - binary data

AMFDataItem

```
public AMFDataItem(byte[] data,  
                   int offset,  
                   int size)
```

Deserialize data array starting at offset for size bytes and create AMFDataItem object

Parameters:

data - binary data

offset - starting offset into data

size - size of data to deserialize

AMFDataItem

```
public AMFDataItem(java.nio.ByteBuffer data)
```

(continued from last page)

Deserialize entire data array and create AMFDataItem object

Parameters:

data - binary data

AMFDataItem

```
public AMFDataItem(java.nio.ByteBuffer data,  
    AMFDataContextDeserialize context)
```

Methods

longValue

```
public long longValue()
```

Return object as long. Valid object types are DATA_TYPE_NUMBER and DATA_TYPE_STRING.

Returns:

long value or 0 if failure

intValue

```
public int intValue()
```

Return object as int. Valid object types are DATA_TYPE_NUMBER and DATA_TYPE_STRING.

Returns:

int value or 0 if failure

doubleValue

```
public double doubleValue()
```

Return object as double. Valid object types are DATA_TYPE_NUMBER and DATA_TYPE_STRING.

Returns:

double value or 0 if failure

floatValue

```
public float floatValue()
```

Return object as float. Valid object types are DATA_TYPE_NUMBER and DATA_TYPE_STRING.

Returns:

float value or 0 if failure

shortValue

```
public short shortValue()
```

Return object as short. Valid object types are DATA_TYPE_NUMBER and DATA_TYPE_STRING.

Returns:

short value or 0 if failure

byteValue

```
public byte byteValue()
```

Return object as byte. Valid object types are DATA_TYPE_NUMBER and DATA_TYPE_STRING.

Returns:

byte value or 0 if failure

dateValue

```
public java.util.Date dateValue()
```

Return object as Date. Valid object types are DATA_TYPE_DATE.

Returns:

Date value or null if failure

booleanValue

```
public boolean booleanValue()
```

Return object as boolean. Valid object types are DATA_TYPE_BOOLEAN and DATA_TYPE_STRING.

Returns:

boolean value or false if failure

deserialize

```
public void deserialize(java.nio.ByteBuffer data)
```

Deserialize data in byte buffer

deserialize

```
public void deserialize(java.nio.ByteBuffer data,  
    AMFDataContextDeserialize context)
```

Deserialize data in byte buffer

getValue

```
public Object getValue()
```

Return value as Java class

toString

```
public String toString()
```

Return object as formatted string

serialize

```
public void serialize(java.io.DataOutputStream out)
```

Serialize object to output stream

serialize

```
public void serialize(java.io.DataOutputStream out,  
    int objectEncoding)
```

Serialize object to output stream

serialize

```
public void serialize(java.io.DataOutputStream out,  
    AMFDataContextSerialize context)
```

Serialize object to output stream

serialize

```
public byte[] serialize()
```

Serial object to byte array

serialize

```
public byte[] serialize(int objectEncoding)
```

Serial object to byte array

serialize

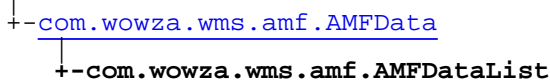
```
public byte[] serialize(AMFDataContextSerialize context)
```

Serial object to byte array

com.wowza.wms.amf

Class AMFDataList

java.lang.Object



public class **AMFDataList**
 extends [AMFData](#)

AMFDataItem: class for marshalling data between Wowza Pro server and Flash client. This class is for internal server marshalling of AMF event messages between the Flash client and Wowza Pro server.

NOTE: There is a slight difference between this class and AMFDataArray. This class when serialized/deserialized does NOT include the DATA_TYPE_ARRAY header (byte) and array size (int). AMF formatted functions use this class since they do not include these elements.

Fields inherited from class [com.wowza.wms.amf.AMFData](#)

[AMF_LEVEL0](#), [AMF_LEVEL3](#), [DATA_TYPE_AMF3](#), [DATA_TYPE_AMF3_ARRAY](#), [DATA_TYPE_AMF3_BOOLEAN_FALSE](#), [DATA_TYPE_AMF3_BOOLEAN_TRUE](#), [DATA_TYPE_AMF3_BYTEARRAY](#), [DATA_TYPE_AMF3_DATE](#), [DATA_TYPE_AMF3_INTEGER](#), [DATA_TYPE_AMF3_NULL](#), [DATA_TYPE_AMF3_NUMBER](#), [DATA_TYPE_AMF3_OBJECT](#), [DATA_TYPE_AMF3_STRING](#), [DATA_TYPE_AMF3_UNDEFINED](#), [DATA_TYPE_AMF3_XML_LEGACY](#), [DATA_TYPE_AMF3_XML_TOP](#), [DATA_TYPE_ARRAY](#), [DATA_TYPE_AS_OBJECT](#), [DATA_TYPE_BOOLEAN](#), [DATA_TYPE_BYTEARRAY](#), [DATA_TYPE_CUSTOM_CLASS](#), [DATA_TYPE_DATE](#), [DATA_TYPE_INTEGER](#), [DATA_TYPE_LONG_STRING](#), [DATA_TYPE_MIXED_ARRAY](#), [DATA_TYPE_MOVIE_CLIP](#), [DATA_TYPE_NULL](#), [DATA_TYPE_NUMBER](#), [DATA_TYPE_OBJECT](#), [DATA_TYPE_OBJECT_END](#), [DATA_TYPE_RECORDSET](#), [DATA_TYPE_REFERENCE_OBJECT](#), [DATA_TYPE_STRING](#), [DATA_TYPE_UNDEFINED](#), [DATA_TYPE_UNKNOWN](#), [DATA_TYPE_XML](#), [DATA_TYPE_XML_TOP](#), [MILLS_PER_HOUR](#), [type](#)

Constructor Summary

public	AMFDataList () Create empty AMFDataList object
public	AMFDataList (byte[] data) Deserialize entire data array and create AMFDataList object
public	AMFDataList (byte[] data, int offset, int size) Deserialize data array starting at offset for size bytes and create AMFDataList object
public	AMFDataList (java.nio.ByteBuffer data) Deserialize entire data array and create AMFDataList object
public	AMFDataList (java.nio.ByteBuffer data, AMFDataContextDeserialize context)

Method Summary

void	add (AMFData data) Append a new item onto the array
------	---

void	add (boolean data) Append a boolean (will be wrapped in an AMFDataItem object)
void	add (java.util.Date data) Append a date (will be wrapped in an AMFDataItem object)
void	add (double data) Append a double (will be wrapped in an AMFDataItem object)
void	add (int data) Append a int (will be wrapped in an AMFDataItem object)
void	add (int index, AMFData data) Insert an item into the array
void	add (int index, boolean data) Insert a boolean value (will be wrapped in an AMFDataItem object)
void	add (int index, java.util.Date data) Insert a date value (will be wrapped in an AMFDataItem object)
void	add (int index, double data) Insert a double value (will be wrapped in an AMFDataItem object)
void	add (int index, int data) Insert a int value (will be wrapped in an AMFDataItem object)
void	add (int index, long data) Insert a long value (will be wrapped in an AMFDataItem object)
void	add (int index, String data) Insert a string value (will be wrapped in an AMFDataItem object)
void	add (long data) Append a long (will be wrapped in an AMFDataItem object)
void	add (String data) Append a string (will be wrapped in an AMFDataItem object)
void	deserialize (java.nio.ByteBuffer data)
void	deserialize (java.nio.ByteBuffer data, AMFDataContextDeserialize context)
AMFData	get (int index) Get item at index
boolean	getBoolean (int index) Get item at index return as boolean
byte	getBytes (int index) Get item at index return as byte
java.util.Date	getDate (int index) Get item at index return as Date
double	getDouble (int index) Get item at index return as double

float	<u>getFloat</u> (int index) Get item at index return as float
int	<u>getInt</u> (int index) Get item at index return as int
long	<u>getLong</u> (int index) Get item at index return as long
<u>AMFDataObj</u>	<u>getObject</u> (int index) Get item at index return as AMFDataObj
short	<u>getShort</u> (int index) Get item at index return as short
String	<u>getString</u> (int index) Get item at index return as String
int	<u>getType</u> (int index) Get type of item at index.
Object	<u>getValue</u> ()
<u>AMFData</u>	<u>remove</u> (int index) Remove an element from the AMFDataList object
byte[]	<u>serialize</u> ()
byte[]	<u>serialize</u> (<u>AMFDataContextSerialize</u> context)
byte[]	<u>serialize</u> (<u>AMFDataContextSerialize</u> context, byte[] prepend)
void	<u>serialize</u> (java.io.DataOutputStream out)
void	<u>serialize</u> (java.io.DataOutputStream out, <u>AMFDataContextSerialize</u> context)
void	<u>serialize</u> (java.io.DataOutputStream out, <u>AMFDataContextSerialize</u> context, byte[] prepend)
void	<u>serialize</u> (java.io.DataOutputStream out, int objectEncoding)
byte[]	<u>serialize</u> (int objectEncoding)
void	<u>set</u> (int index, <u>AMFData</u> data) Set an array item
void	<u>set</u> (int index, boolean data) Set an boolean value (will be wrapped in an AMFDataItem object)
void	<u>set</u> (int index, java.util.Date data) Set an date value (will be wrapped in an AMFDataItem object)
void	<u>set</u> (int index, double data) Set an double value (will be wrapped in an AMFDataItem object)

void	set (int index, int data) Set an int value (will be wrapped in an AMFDataItem object)
void	set (int index, long data) Set an long value (will be wrapped in an AMFDataItem object)
void	set (int index, String data) Set an string value (will be wrapped in an AMFDataItem object)
int	size () Returns the number of items in array
String	toString () Return object as formatted string

Methods inherited from class [com.wowza.wms.amf.AMFData](#)

[createContextDeserialize](#), [createContextDeserialize](#), [createContextSerialize](#), [createContextSerialize](#), [deserialize](#), [deserialize](#), [deserializeInnerObject](#), [getReference](#), [getType](#), [getValue](#), [isAMF3Start](#), [isArrayStart](#), [isByteArrayStart](#), [isMixedArrayStart](#), [isObjEnd](#), [isObjStart](#), [peekByte](#), [serialize](#), [serialize](#), [serialize](#), [serialize](#), [serialize](#), [setType](#), [skipByte](#), [testNextByte](#), [triggerAMF3Switch](#)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

AMFDataList

```
public AMFDataList()
```

Create empty AMFDataList object

AMFDataList

```
public AMFDataList(byte[] data)
```

Deserialize entire data array and create AMFDataList object

Parameters:

data - binary data

AMFDataList

```
public AMFDataList(byte[] data,  
                  int offset,  
                  int size)
```

Deserialize data array starting at offset for size bytes and create AMFDataList object

Parameters:

data - binary data

offset - starting offset into data

(continued from last page)

size - size of data to deserialize

AMFDataList

```
public AMFDataList(java.nio.ByteBuffer data)
```

Deserialize entire data array and create AMFDataList object

Parameters:

data - binary data

AMFDataList

```
public AMFDataList(java.nio.ByteBuffer data,  
    AMFDataContextDeserialize context)
```

Methods

remove

```
public AMFData remove(int index)
```

Remove an element from the AMFDataList object

Parameters:

index - index

Returns:

removed element

size

```
public int size()
```

Returns the number of items in array

Returns:

number of items in array

add

```
public void add(AMFData data)
```

Append a new item onto the array

Parameters:

data - AMFData object

add

```
public void add(String data)
```

Append a string (will be wrapped in an AMFDataItem object)

Parameters:

data - string value

add

```
public void add(double data)
```

Append a double (will be wrapped in an AMFDataItem object)

Parameters:

data - double value

add

```
public void add(int data)
```

Append a int (will be wrapped in an AMFDataItem object)

Parameters:

data - int value

add

```
public void add(long data)
```

Append a long (will be wrapped in an AMFDataItem object)

Parameters:

data - long value

add

```
public void add(java.util.Date data)
```

Append a date (will be wrapped in an AMFDataItem object)

Parameters:

data - date value

add

```
public void add(boolean data)
```

Append a boolean (will be wrapped in an AMFDataItem object)

Parameters:

data - boolean value

add

```
public void add(int index,  
    AMFData data)
```

Insert an item into the array

Parameters:

index - index

data - AMFData object

(continued from last page)

add

```
public void add(int index,  
                String data)
```

Insert a string value (will be wrapped in an AMFDataItem object)

Parameters:

index - index
data - string value

add

```
public void add(int index,  
                double data)
```

Insert a double value (will be wrapped in an AMFDataItem object)

Parameters:

index - index
data - double value

add

```
public void add(int index,  
                int data)
```

Insert a int value (will be wrapped in an AMFDataItem object)

Parameters:

index - index
data - int value

add

```
public void add(int index,  
                long data)
```

Insert a long value (will be wrapped in an AMFDataItem object)

Parameters:

index - index
data - long value

add

```
public void add(int index,  
                java.util.Date data)
```

Insert a date value (will be wrapped in an AMFDataItem object)

Parameters:

index - index
data - date value

add

```
public void add(int index,  
                boolean data)
```

(continued from last page)

Insert a boolean value (will be wrapped in an AMFDataItem object)

Parameters:

index - index
data - boolean value

set

```
public void set(int index,  
    AMFData data)
```

Set an array item

Parameters:

data - AMFData object

set

```
public void set(int index,  
    String data)
```

Set an string value (will be wrapped in an AMFDataItem object)

Parameters:

index - index
data - string value

set

```
public void set(int index,  
    double data)
```

Set an double value (will be wrapped in an AMFDataItem object)

Parameters:

index - index
data - double value

set

```
public void set(int index,  
    int data)
```

Set an int value (will be wrapped in an AMFDataItem object)

Parameters:

index - index
data - int value

set

```
public void set(int index,  
    long data)
```

Set an long value (will be wrapped in an AMFDataItem object)

Parameters:

index - index
data - long value

set

```
public void set(int index,  
                java.util.Date data)
```

Set an date value (will be wrapped in an AMFDataItem object)

Parameters:

index - index
data - date value

set

```
public void set(int index,  
                boolean data)
```

Set an boolean value (will be wrapped in an AMFDataItem object)

Parameters:

index - index
data - boolean value

getType

```
public int getType(int index)
```

Get type of item at index. Return AMFData.DATA_TYPE_UNKNOWN if item does not exist

Parameters:

index

Returns:

type of item at index

get

```
public AMFData get(int index)
```

Get item at index

Parameters:

index

Returns:

Returns AMFData object or null if out of bounds

getString

```
public String getString(int index)
```

Get item at index return as String

Parameters:

index

Returns:

Return item as String or null if out of bounds

getInt

```
public int getInt(int index)
```

Get item at index return as int

Parameters:

index

Returns:

Return item as int or 0 if out of bounds

getLong

```
public long getLong(int index)
```

Get item at index return as long

Parameters:

index

Returns:

Return item as long or 0 if out of bounds

getDouble

```
public double getDouble(int index)
```

Get item at index return as double

Parameters:

index

Returns:

Return item as double or 0 if out of bounds

getFloat

```
public float getFloat(int index)
```

Get item at index return as float

Parameters:

index

Returns:

Return item as float or 0 if out of bounds

getShort

```
public short getShort(int index)
```

Get item at index return as short

Parameters:

index

(continued from last page)

Returns:

Return item as short or 0 if out of bounds

getBytes

```
public byte getBytes(int index)
```

Get item at index return as byte

Parameters:

index

Returns:

Return item as byte or 0 if out of bounds

getBoolean

```
public boolean getBoolean(int index)
```

Get item at index return as boolean

Parameters:

index

Returns:

Return item as boolean or false if out of bounds

getDate

```
public java.util.Date getDate(int index)
```

Get item at index return as Date

Parameters:

index

Returns:

Return item as Date or null if out of bounds

getObject

```
public AMFDataObj getObject(int index)
```

Get item at index return as AMFDataObj

Parameters:

index

Returns:

Return item as AMFDataObj or null if out of bounds

deserialize

```
public void deserialize(java.nio.ByteBuffer data)
```

Deserialize data in byte buffer

(continued from last page)

deserialize

```
public void deserialize(java.nio.ByteBuffer data,  
    AMFDataContextDeserialize context)
```

Deserialize data in byte buffer

serialize

```
public void serialize(java.io.DataOutputStream out)
```

Serialize object to output stream

serialize

```
public void serialize(java.io.DataOutputStream out,  
    int objectEncoding)
```

Serialize object to output stream

serialize

```
public void serialize(java.io.DataOutputStream out,  
    AMFDataContextSerialize context)
```

Serialize object to output stream

serialize

```
public void serialize(java.io.DataOutputStream out,  
    AMFDataContextSerialize context,  
    byte[] prepend)
```

serialize

```
public byte[] serialize()
```

Serial object to byte array

serialize

```
public byte[] serialize(int objectEncoding)
```

Serial object to byte array

serialize

```
public byte[] serialize(AMFDataContextSerialize context)
```

Serial object to byte array

serialize

```
public byte[] serialize(AMFDataContextSerialize context,  
    byte[] prepend)
```

getValue

```
public Object getValue()
```

Convert object to Java native class

toString

```
public String toString()
```

Return object as formatted string

com.wowza.wms.amf

Class AMFDataMixedArray

```

java.lang.Object
|
+-com.wowza.wms.amf.AMFData
|   |
|   +-com.wowza.wms.amf.AMFDataObj
|       |
|       +-com.wowza.wms.amf.AMFDataMixedArray

```

All Implemented Interfaces:

[IAMFDataObj](#)

```

public class AMFDataMixedArray
extends AMFDataObj

```

AMFDataMixedArray: class for marshalling data between Wowza Pro server and Flash client. Array of mixed data types. An Array object created in the Flash client is wrapped in this data type when sent to the Wowza Pro server.

Create Array of Strings

```

AMFDataMixedArray amfDataMixedArray = new AMFDataMixedArray();

amfDataMixedArray.put("0", "item1");
amfDataMixedArray.put("1", "item2");
amfDataMixedArray.put("2", "item3");

```

Iterate Mixed Array

```

AMFDataMixedArray amfDataMixedArray;

int len = amfDataMixedArray.size();
for(int i=0;i<len;i++)
{
    String value = amfDataMixedArray.getString(i);
    int itemType = amfDataMixedArray.getType(i);
    WMSLoggerFactory.getLogger(null).debug("item:
["+i+": "+amfDataMixedArray.getKey(i)+"]="+value+" type:"+itemType);
}

```

NOTE: A AMFDataMixedArray is exactly the same as a AMFDataObj except its type is DATA_TYPE_MIXED_ARRAY.

This objects acts like a Map and a List at the same time. As items are added by key the order and position of each object is recorded. Objects can be retrieved either by key or by index.

Fields inherited from class [com.wowza.wms.amf.AMFDataObj](#)

[DECODE_OBJ_REF](#), [DECODE_TRAITS](#), [DECODE_TRAITS_EXT](#), [DECODE_TRAITS_REF](#), [DECODE_UNDEFINED](#), [members](#), [order](#), [serializer](#), [trait](#)

Fields inherited from class [com.wowza.wms.amf.AMFData](#)

[AMF_LEVEL0](#), [AMF_LEVEL3](#), [DATA_TYPE_AMF3](#), [DATA_TYPE_AMF3_ARRAY](#), [DATA_TYPE_AMF3_BOOLEAN_FALSE](#), [DATA_TYPE_AMF3_BOOLEAN_TRUE](#), [DATA_TYPE_AMF3_BYTEARRAY](#), [DATA_TYPE_AMF3_DATE](#), [DATA_TYPE_AMF3_INTEGER](#), [DATA_TYPE_AMF3_NULL](#), [DATA_TYPE_AMF3_NUMBER](#), [DATA_TYPE_AMF3_OBJECT](#), [DATA_TYPE_AMF3_STRING](#), [DATA_TYPE_AMF3_UNDEFINED](#), [DATA_TYPE_AMF3_XML_LEGACY](#), [DATA_TYPE_AMF3_XML_TOP](#), [DATA_TYPE_ARRAY](#), [DATA_TYPE_AS_OBJECT](#), [DATA_TYPE_BOOLEAN](#), [DATA_TYPE_BYTEARRAY](#), [DATA_TYPE_CUSTOM_CLASS](#), [DATA_TYPE_DATE](#), [DATA_TYPE_INTEGER](#), [DATA_TYPE_LONG_STRING](#), [DATA_TYPE_MIXED_ARRAY](#), [DATA_TYPE_MOVIE_CLIP](#), [DATA_TYPE_NULL](#), [DATA_TYPE_NUMBER](#), [DATA_TYPE_OBJECT](#), [DATA_TYPE_OBJECT_END](#), [DATA_TYPE_RECORDSET](#), [DATA_TYPE_REFERENCE_OBJECT](#), [DATA_TYPE_STRING](#), [DATA_TYPE_UNDEFINED](#), [DATA_TYPE_UNKNOWN](#), [DATA_TYPE_XML](#), [DATA_TYPE_XML_TOP](#), [MILLS_PER_HOUR](#), [type](#)

Constructor Summary

public	AMFDataMixedArray() Create empty AMFDataMixedArray object
public	AMFDataMixedArray (byte[] data) Deserialize entire data array and create AMFDataMixedArray object
public	AMFDataMixedArray (byte[] data, int offset, int size) Deserialize data array starting at offset for size bytes and create AMFDataMixedArray object
public	AMFDataMixedArray (java.nio.ByteBuffer data) Deserialize entire data array and create AMFDataMixedArray object
public	AMFDataMixedArray (java.nio.ByteBuffer data, AMFDataContextDeserialize context)

Method Summary

void	deserialize (java.nio.ByteBuffer data)
void	deserialize (java.nio.ByteBuffer data, AMFDataContextDeserialize context)
void	serialize (java.io.DataOutputStream out)
void	serialize (java.io.DataOutputStream out, AMFDataContextSerialize context)
void	serialize (java.io.DataOutputStream out, int objectEncoding)
String	toString () Return object as formatted string

Methods inherited from class [com.wowza.wms.amf.AMFDataObj](#)

[containsKey](#), [deserialize](#), [deserialize](#), [get](#), [get](#), [getBoolean](#), [getBoolean](#), [getBytes](#), [getBytes](#), [getClassName](#), [getDate](#), [getDate](#), [getDouble](#), [getDouble](#), [getFloat](#), [getFloat](#), [getInt](#), [getInt](#), [getKey](#), [getKeys](#), [getLong](#), [getLong](#), [getObject](#), [getObject](#), [getSerializer](#), [getShort](#), [getShort](#), [getString](#), [getString](#), [getTrait](#), [getValue](#), [put](#), [put](#), [put](#), [put](#), [put](#), [put](#), [remove](#), [remove](#), [serialize](#), [serialize](#), [serialize](#), [serialize](#), [serialize](#), [setClassName](#), [setSerializer](#), [size](#), [toString](#)

Methods inherited from class [com.wowza.wms.amf.AMFData](#)

[createContextDeserialize](#), [createContextDeserialize](#), [createContextSerialize](#), [createContextSerialize](#), [deserialize](#), [deserialize](#), [deserializeInnerObject](#), [getReference](#), [getType](#), [getValue](#), [isAMF3Start](#), [isArrayStart](#), [isByteArrayStart](#), [isMixedArrayStart](#), [isObjEnd](#), [isObjStart](#), [peekByte](#), [serialize](#), [serialize](#), [serialize](#), [serialize](#), [serialize](#), [set](#), [setType](#), [skipByte](#), [testNextByte](#), [triggerAMF3Switch](#)

Methods inherited from class [java.lang.Object](#)

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

Methods inherited from interface [com.wowza.wms.amf.IAMFDataObj](#)

[containsKey](#), [get](#), [get](#), [getBoolean](#), [getBoolean](#), [getBytes](#), [getBytes](#), [getDate](#), [getDate](#), [getDouble](#), [getDouble](#), [getFloat](#), [getFloat](#), [getInt](#), [getInt](#), [getKey](#), [getKeys](#), [getLong](#), [getLong](#), [getObject](#), [getObject](#), [getShort](#), [getShort](#), [getString](#), [getString](#), [put](#), [put](#), [put](#), [put](#), [put](#), [put](#), [remove](#), [remove](#)

Constructors

AMFDataMixedArray

```
public AMFDataMixedArray()
```

Create empty AMFDataMixedArray object

AMFDataMixedArray

```
public AMFDataMixedArray(byte[] data)
```

Deserialize entire data array and create AMFDataMixedArray object

Parameters:

data - binary data

AMFDataMixedArray

```
public AMFDataMixedArray(byte[] data,  
                           int offset,  
                           int size)
```

Deserialize data array starting at offset for size bytes and create AMFDataMixedArray object

Parameters:

data - binary data

(continued from last page)

offset - starting offset into data
size - size of data to deserialize

AMFDataMixedArray

```
public AMFDataMixedArray( java.nio.ByteBuffer data)
```

Deserialize entire data array and create AMFDataMixedArray object

Parameters:

data - binary data

AMFDataMixedArray

```
public AMFDataMixedArray( java.nio.ByteBuffer data,  
                           AMFDataContextDeserialize context)
```

Methods

deserialize

```
public void deserialize( java.nio.ByteBuffer data)
```

Deserialize data in byte buffer

deserialize

```
public void deserialize( java.nio.ByteBuffer data,  
                           AMFDataContextDeserialize context)
```

Deserialize data in byte buffer

serialize

```
public void serialize( java.io.DataOutputStream out)
```

Serialize object to output stream

serialize

```
public void serialize( java.io.DataOutputStream out,  
                       int objectEncoding)
```

Serialize object to output stream

serialize

```
public void serialize( java.io.DataOutputStream out,  
                           AMFDataContextSerialize context)
```

Serialize object to output stream

toString

```
public String toString()
```

Return object as formatted string

com.wowza.wms.amf

Class AMFDataObj

```
java.lang.Object
|
+-com.wowza.wms.amf.AMFData
|
+-com.wowza.wms.amf.AMFDataObj
```

All Implemented Interfaces:

[IAMFDataObj](#)

Direct Known Subclasses:

[AMFDataMixedArray](#)

```
public class AMFDataObj
extends AMFData
implements IAMFDataObj
```

AMFDataObj: class for marshalling data between Wowza Pro server and Flash client. Object with attributes. Implementation is very similar to a java.util.Map. Each parameter is an item in the map.

Create AMFDataObj

```
AMFDataObj amfDataObj = new AMFDataObj();

amfDataObj.put("key1", "item1");
amfDataObj.put("key2", "item2");
amfDataObj.put("key3", "item3");
```

Iterate AMFDataObj

```
AMFDataObj amfDataObj;

List keys = amfDataObj.getKeys();
Iterator iter = keys.iterator();
while(iter.hasNext())
{
    String key = (String)iter.next();
    AMFData value = amfDataObj.get(key);
    int itemType = value.getType();
    WMSLoggerFactory.getLogger(null).debug(key+"="+value.toString()+"
(type:"+itemType+")");
}
```

Direct Access To Attributes

```
AMFDataObj amfDataObj;

// If you know the type you can access it directly
String dataString = amfDataObj.getString("stringData");
long dataLong = amfDataObj.getLong("longData");
double dataDouble = amfDataObj.getDouble("doubleData");
boolean dataBoolean = amfDataObj.getBoolean("booleanData");

// This illustrate how to decode the value if
// you don't know the type
AMFData myItemKey1 = amfDataObj.get("theData");
switch (myItemKey1.getType())
{
default:
case AMFDataItem.DATA_TYPE_UNDEFINED:
case AMFDataItem.DATA_TYPE_UNKNOWN:
case AMFDataItem.DATA_TYPE_NULL:
    // the value is null or undefined
    break;
case AMFDataItem.DATA_TYPE_NUMBER:
    double amfDataDouble = ((AMFDataItem)myItemKey1).doubleValue();
    break;
case AMFDataItem.DATA_TYPE_BOOLEAN:
    boolean amfDataBoolean = ((AMFDataItem)myItemKey1).booleanValue();
    break;
case AMFDataItem.DATA_TYPE_STRING:
    String amfDataString = ((AMFDataItem)myItemKey1).toString();
    break;
case AMFDataItem.DATA_TYPE_DATE:
    Date amfDataDate = ((AMFDataItem)myItemKey1).dateValue();
    break;
case AMFDataItem.DATA_TYPE_OBJECT:
    AMFDataObj amfDataValObj = (AMFDataObj)myItemKey1;
    break;
case AMFDataItem.DATA_TYPE_MIXED_ARRAY:
    AMFDataMixedArray amfDataMixedArray = (AMFDataMixedArray)myItemKey1;
    break;
case AMFDataItem.DATA_TYPE_ARRAY:
    AMFDataArray amfDataArray = (AMFDataArray)myItemKey1;
    break;
}
```

NOTE: A AMFDataObj is exactly the same as a AMFDataMixedArray except its type is DATA_TYPE_OBJECT.

Field Summary

public static final	DECODE_OBJ_REF Value: 1
public static final	DECODE_TRAITS Value: 4
public static final	DECODE_TRAITS_EXT Value: 3
public static final	DECODE_TRAITS_REF Value: 2
public static final	DECODE_UNDEFINED Value: 0
protected	members
protected	order
protected	serializer
protected	trait

Fields inherited from class [com.wowza.wms.amf.AMFData](#)

[AMF_LEVEL0](#), [AMF_LEVEL3](#), [DATA_TYPE_AMF3](#), [DATA_TYPE_AMF3_ARRAY](#), [DATA_TYPE_AMF3_BOOLEAN_FALSE](#),
[DATA_TYPE_AMF3_BOOLEAN_TRUE](#), [DATA_TYPE_AMF3_BYTEARRAY](#), [DATA_TYPE_AMF3_DATE](#),
[DATA_TYPE_AMF3_INTEGER](#), [DATA_TYPE_AMF3_NULL](#), [DATA_TYPE_AMF3_NUMBER](#), [DATA_TYPE_AMF3_OBJECT](#),
[DATA_TYPE_AMF3_STRING](#), [DATA_TYPE_AMF3_UNDEFINED](#), [DATA_TYPE_AMF3_XML_LEGACY](#),
[DATA_TYPE_AMF3_XML_TOP](#), [DATA_TYPE_ARRAY](#), [DATA_TYPE_AS_OBJECT](#), [DATA_TYPE_BOOLEAN](#),
[DATA_TYPE_BYTEARRAY](#), [DATA_TYPE_CUSTOM_CLASS](#), [DATA_TYPE_DATE](#), [DATA_TYPE_INTEGER](#),
[DATA_TYPE_LONG_STRING](#), [DATA_TYPE_MIXED_ARRAY](#), [DATA_TYPE_MOVIE_CLIP](#), [DATA_TYPE_NULL](#),
[DATA_TYPE_NUMBER](#), [DATA_TYPE_OBJECT](#), [DATA_TYPE_OBJECT_END](#), [DATA_TYPE_RECORDSET](#),
[DATA_TYPE_REFERENCE_OBJECT](#), [DATA_TYPE_STRING](#), [DATA_TYPE_UNDEFINED](#), [DATA_TYPE_UNKNOWN](#),
[DATA_TYPE_XML](#), [DATA_TYPE_XML_TOP](#), [MILLS_PER_HOUR](#), [type](#)

Constructor Summary

public	AMFDataObj() Create empty AMFDataObj object
public	AMFDataObj(byte[] data) Deserialize entire data array and create AMFDataObj object
public	AMFDataObj(byte[] data, int offset, int size) Deserialize data array starting at offset for size bytes and create AMFDataObj object
public	AMFDataObj(java.nio.ByteBuffer data) Deserialize entire data array and create AMFDataObj object

public	AMFDataObj (java.nio.ByteBuffer data, AMFDataContextDeserialize context)
--------	--

Method Summary

boolean	containsKey (String name)
void	deserialize (java.nio.ByteBuffer data)
void	deserialize (java.nio.ByteBuffer data, AMFDataContextDeserialize context)
AMFData	get (int index)
AMFData	get (String name)
boolean	getBoolean (int index)
boolean	getBoolean (String name)
byte	getBytes (int index)
byte	getBytes (String name)
String	getClassName ()
java.util.Date	getDate (int index)
java.util.Date	getDate (String name)
double	getDouble (int index)
double	getDouble (String name)
float	getFloat (int index)
float	getFloat (String name)
int	getInt (int index)
int	getInt (String name)
String	getKey (int index)
java.util.List	getKeys ()
long	getLong (int index)
long	getLong (String name)

AMFDataObj	getObject (int index)
AMFDataObj	getObject (String name)
IAMFExternalizable	getSerializer ()
short	getShort (int index)
short	getShort (String name)
String	getString (int index)
String	getString (String name)
AMFDataTrait	getTrait ()
Object	getValue ()
void	put (String name, AMFData data)
void	put (String name, boolean data)
void	put (String name, java.util.Date data)
void	put (String name, double data)
void	put (String name, int data)
void	put (String name, long data)
void	put (String name, String data)
AMFData	remove (int index)
AMFData	remove (String name)
byte[]	serialize ()
byte[]	serialize (AMFDataContextSerialize context)
void	serialize (java.io.DataOutputStream out)
void	serialize (java.io.DataOutputStream out, AMFDataContextSerialize context)
void	serialize (java.io.DataOutputStream out, int objectEncoding)
byte[]	serialize (int objectEncoding)

void	setClassName (String className)
void	setSerializer (IAMFExternalizable serializer)
int	size () Return the number of members of this object/array
String	toString () Return object as formatted string

Methods inherited from class com.wowza.wms.amf.AMFData	
createContextDeserialize , createContextDeserialize , createContextSerialize , createContextSerialize , deserialize , deserialize , deserializeInnerObject , getReference , getType , getValue , isAMF3Start , isArrayStart , isByteArrayStart , isMixedArrayStart , isObjEnd , isObjStart , peekByte , serialize , serialize , serialize , serialize , serialize , setType , skipByte , testNextByte , triggerAMF3Switch	

Methods inherited from class java.lang.Object	
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait	

Methods inherited from interface com.wowza.wms.amf.IAMFDataObj	
containsKey , get , get , getBoolean , getBoolean , getByte , getByte , getDate , getDate , getDouble , getDouble , getFloat , getFloat , getInt , getInt , getKey , getKeys , getLong , getLong , getObject , getObject , getShort , getShort , getString , getString , put , put , put , put , put , put , remove , remove	

Fields

DECODE_UNDEFINED

```
public static final int DECODE_UNDEFINED
```

Constant value: 0

DECODE_OBJ_REF

```
public static final int DECODE_OBJ_REF
```

Constant value: 1

DECODE_TRAITS_REF

```
public static final int DECODE_TRAITS_REF
```

Constant value: 2

(continued from last page)

DECODE_TRAITS_EXT

```
public static final int DECODE_TRAITS_EXT
```

Constant value: **3**

DECODE_TRAITS

```
public static final int DECODE_TRAITS
```

Constant value: **4**

members

```
protected java.util.Map members
```

order

```
protected java.util.List order
```

trait

```
protected com.wowza.wms.amf.AMFDataTrait trait
```

serializer

```
protected com.wowza.wms.amf.IAMFExternalizable serializer
```

Constructors

AMFDataObj

```
public AMFDataObj()
```

Create empty AMFDataObj object

AMFDataObj

```
public AMFDataObj(byte[] data)
```

Deserialize entire data array and create AMFDataObj object

Parameters:

data - binary data

(continued from last page)

AMFDataObj

```
public AMFDataObj(byte[] data,  
                  int offset,  
                  int size)
```

Deserialize data array starting at offset for size bytes and create AMFDataObj object

Parameters:

data - binary data
offset - starting offset into data
size - size of data to deserialize

AMFDataObj

```
public AMFDataObj(java.nio.ByteBuffer data)
```

Deserialize entire data array and create AMFDataObj object

Parameters:

data - binary data

AMFDataObj

```
public AMFDataObj(java.nio.ByteBuffer data,  
                  AMFDataContextDeserialize context)
```

Methods

size

```
public int size()
```

Return the number of members of this object/array

Returns:

number of members

containsKey

```
public boolean containsKey(String name)
```

put

```
public void put(String name,  
               AMFData data)
```

put

```
public void put(String name,  
               String data)
```

put

```
public void put(String name,  
                double data)
```

put

```
public void put(String name,  
                int data)
```

put

```
public void put(String name,  
                long data)
```

put

```
public void put(String name,  
                java.util.Date data)
```

put

```
public void put(String name,  
                boolean data)
```

getKeys

```
public java.util.List getKeys()
```

getKey

```
public String getKey(int index)
```

get

```
public AMFData get(String name)
```

get

```
public AMFData get(int index)
```

remove

```
public AMFData remove(String name)
```

remove

```
public AMFData remove(int index)
```

getString

```
public String getString(String name)
```

getInt

```
public int getInt(String name)
```

getLong

```
public long getLong(String name)
```

getShort

```
public short getShort(String name)
```

getDouble

```
public double getDouble(String name)
```

getFloat

```
public float getFloat(String name)
```

getByte

```
public byte getByte(String name)
```

getBoolean

```
public boolean getBoolean(String name)
```

(continued from last page)

getDate

```
public java.util.Date getDate(String name)
```

getObject

```
public AMFDataObj getObject(String name)
```

getString

```
public String getString(int index)
```

getInt

```
public int getInt(int index)
```

getLong

```
public long getLong(int index)
```

getShort

```
public short getShort(int index)
```

getBytes

```
public byte getBytes(int index)
```

getDouble

```
public double getDouble(int index)
```

getFloat

```
public float getFloat(int index)
```

(continued from last page)

getBoolean

```
public boolean getBoolean(int index)
```

getDate

```
public java.util.Date getDate(int index)
```

getObject

```
public AMFDataObj getObject(int index)
```

deserialize

```
public void deserialize(java.nio.ByteBuffer data)
```

Deserialize data in byte buffer

deserialize

```
public void deserialize(java.nio.ByteBuffer data,  
    AMFDataContextDeserialize context)
```

Deserialize data in byte buffer

serialize

```
public void serialize(java.io.DataOutputStream out)
```

Serialize object to output stream

serialize

```
public void serialize(java.io.DataOutputStream out,  
    int objectEncoding)
```

Serialize object to output stream

serialize

```
public void serialize(java.io.DataOutputStream out,  
    AMFDataContextSerialize context)
```

Serialize object to output stream

serialize

```
public byte[] serialize()
```

Serial object to byte array

(continued from last page)

serialize

```
public byte[] serialize(int objectEncoding)
```

Serial object to byte array

serialize

```
public byte[] serialize(AMFDataContextSerialize context)
```

Serial object to byte array

getValue

```
public Object getValue()
```

Convert object to Java native class

toString

```
public String toString()
```

Return object as formatted string

getClassName

```
public String getClassName()
```

setClassName

```
public void setClassName(String className)
```

getTrait

```
public AMFDataTrait getTrait()
```

getSerializer

```
public IAMFExternalizable getSerializer()
```

setSerializer

```
public void setSerializer(IAMFExternalizable serializer)
```

com.wowza.wms.amf

Class AMFDataTrait

java.lang.Object

└─com.wowza.wms.amf.AMFDataTrait

public class **AMFDataTrait**
extends Object

AMF trait used to describe an AMF class in AMF3

Constructor Summary

public	AMFDataTrait () Constructor
--------	---

Method Summary

void	addMember (String member) Add a member
AMFDataTrait	clone () clone the trait
String	getClassName () Get class name
AMFData	getInnerObj () Get inner object
String	getMember (int i) Get member by index
int	getMemberCount () Get the number of members
java.util.List	getMembers () Get a list of trait members
boolean	isDynamic () Is class dynamic
boolean	isMember (String member) Return true if member of this trait
void	setClassName (String className) Set class name
void	setDynamic (boolean isDynamic) Set isDynamic
void	setInnerObj (AMFData innerObj) Set inner object

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

AMFDataTrait

```
public AMFDataTrait()
```

Constructor

Methods

clone

```
public AMFDataTrait clone()
```

clone the trait

addMember

```
public void addMember(String member)
```

Add a member

Parameters:

member - member name

isMember

```
public boolean isMember(String member)
```

Return true if member of this trait

Parameters:

member - member name

Returns:

true, if member

getMembers

```
public java.util.List getMembers()
```

Get a list of trait members

Returns:

list of trait members

getMemberCount

```
public int getMemberCount()
```

(continued from last page)

Get the number of members

Returns:

number of members

getMember

```
public String getMember(int i)
```

Get member by index

Parameters:

i - index

Returns:

member name

getClassName

```
public String getClassName()
```

Get class name

Returns:

class name

setClassName

```
public void setClassName(String className)
```

Set class name

Parameters:

className - class name

isDynamic

```
public boolean isDynamic()
```

Is class dynamic

Returns:

true, if dynamic

setDynamic

```
public void setDynamic(boolean isDynamic)
```

Set isDynamic

Parameters:

isDynamic - true, if dynamic

getInnerObj

```
public AMFData getInnerObj()
```

Get inner object

(continued from last page)

Returns:

inner object

setInnerObj

```
public void setInnerObj(AMFData innerObj)
```

Set inner object

Parameters:

innerObj - inner object

com.wowza.wms.amf

Class AMFObj

java.lang.Object

└─com.wowza.wms.amf.AMFObj

public class **AMFObj**
extends Object

AMFObj: class that stores the state of channel between the client and the server.

Field Summary

public static final	AMFDEBUGHEADERSIZE Value: false
public static final	WOWZDEBUGHEADERSIZE Value: false

Constructor Summary

public	AMFObj (int id) Create new AMFObj for a given channel (id)
public	AMFObj (int id, int objectEncoding) Create new AMFObj for a given channel (id)

Method Summary

void	addChunk (byte[] buffer, int offset, int len) Add a chunk to the chunk list
void	clearByteContainer () Clear the byte container
long	getAbsTimecode () Get the absolute time code
int	getByteContainerLevel () Fake container for processing
long	getChunkCounter ()
java.util.List	getChunks () Get the chunks that make up this packet
int	getId () Get channel id
int	getObjectEncoding ()

int	<u>getSize()</u> Get packet size
int	<u>getSrc()</u> Get stream id (0 if not stream data)
long	<u>getTimecode()</u> Get timecode (milliseconds) sometimes relative
int	<u>getType()</u> Get content type IVHost.CONTENTTYPE_*
long	<u>incAbsTimecode(long absTimecode)</u> Increment the absolute timecode
void	<u>incByteContainerLevel(int byteContainerLevel)</u> Fake container for processing
boolean	<u>isByteContainerEmpty()</u> Fake container for processing
boolean	<u>isByteContainerFull()</u> Fake container for processing
boolean	<u>isLastSentAbsTimecode()</u>
boolean	<u>isLongTimecode()</u> Get is a 32 bit timecode
boolean	<u>isNew()</u> Is this a new packet.
boolean	<u>isObjectEncodingAMF0()</u>
boolean	<u>isObjectEncodingAMF3()</u>
long	<u>setAbsTimecodeLong(long absTimecode)</u> Set the absolute timecode
long	<u>setAbsTimecodeShort(long absTimecode)</u> Set the absolute timecode
void	<u>setByteContainerLevel(int byteContainerLevel)</u> Fake container for processing
void	<u>setChunkCounter(long chunkCounter)</u>
void	<u>setId(int id)</u> Set channel id
void	<u>setLastSentAbsTimecode(boolean isLastSentAbsTimecode)</u>
void	<u>setLongTimecode(boolean isLongTimecode)</u> Set is a 32 bit timecode
void	<u>setNew(boolean isNew)</u> Set is new packet

void	<code>setObjectEncoding</code> (int objectEncoding)
void	<code>setSize</code> (int size) Set packet size
void	<code>setSrc</code> (int src) Set stream id (0 if not stream data)
void	<code>setTimecode</code> (long timecode) Set timecode (milliseconds) sometimes relative
void	<code>setType</code> (int type) Set content type IVHost.CONTENT_TYPE_*
String	<code>toString</code> () Return object as formatted string

Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Fields

AMFDEBUGHEADERSIZE

public static final boolean **AMFDEBUGHEADERSIZE**

Constant value: **false**

WOWZDEBUGHEADERSIZE

public static final boolean **WOWZDEBUGHEADERSIZE**

Constant value: **false**

Constructors

AMFObj

public **AMFObj**(int id)

Create new AMFObj for a given channel (id)

Parameters:

id - channel id

AMFObj

public **AMFObj**(int id,
int objectEncoding)

Create new AMFObj for a given channel (id)

(continued from last page)

Parameters:

id - channel id
objectEncoding - object encoding level (AMF0 or AMF3)

Methods

getChunks

```
public java.util.List getChunks()
```

Get the chunks that make up this packet

Returns:

chunks that make up this packet

addChunk

```
public void addChunk(byte[] buffer,  
    int offset,  
    int len)
```

Add a chunk to the chunk list

Parameters:

buffer - buffer
offset - offset
len - length

getId

```
public int getId()
```

Get channel id

Returns:

channel id

setId

```
public void setId(int id)
```

Set channel id

Parameters:

id - channel id

getSize

```
public int getSize()
```

Get packet size

Returns:

packet size

setSize

```
public void setSize(int size)
```

(continued from last page)

Set packet size

Parameters:

size - packet size

getType

```
public int getType()
```

Get content type IVHost.CONTENTTYPE_*

Returns:

content type

setType

```
public void setType(int type)
```

Set content type IVHost.CONTENTTYPE_*

Parameters:

type - content type

getSrc

```
public int getSrc()
```

Get stream id (0 if not stream data)

Returns:

stream id

setSrc

```
public void setSrc(int src)
```

Set stream id (0 if not stream data)

Parameters:

src - stream id

toString

```
public String toString()
```

Return object as formatted string

getTimecode

```
public long getTimecode()
```

Get timecode (milliseconds) sometimes relative

Returns:

timecode (milliseconds)

(continued from last page)

setTimecode

```
public void setTimecode(long timecode)
```

Set timecode (milliseconds) sometimes relative

Parameters:

timecode - timecode (milliseconds)

isNew

```
public boolean isNew()
```

Is this a new packet. If new entire packet header needs to be sent

Returns:

is new packet

setNew

```
public void setNew(boolean isNew)
```

Set is new packet

Parameters:

isNew - is new packet

getAbsTimecode

```
public long getAbsTimecode()
```

Get the absolute time code

Returns:

absolute timecode

setAbsTimecodeLong

```
public long setAbsTimecodeLong(long absTimecode)
```

Set the absolute timecode

Parameters:

absTimecode - absolute timecode

Returns:

absolute timecode

setAbsTimecodeShort

```
public long setAbsTimecodeShort(long absTimecode)
```

Set the absolute timecode

Parameters:

absTimecode

Returns:

(continued from last page)

absolute timecode

incAbsTimecode

```
public long incAbsTimecode(long absTimecode)
```

Increment the absolute timecode

Parameters:

absTimecode - absolute timecode

Returns:

absolute timecode

getByteContainerLevel

```
public int getByteContainerLevel()
```

Fake container for processing

Returns:

current container level

clearByteContainer

```
public void clearByteContainer()
```

Clear the byte container

setByteContainerLevel

```
public void setByteContainerLevel(int byteContainerLevel)
```

Fake container for processing

Parameters:

byteContainerLevel - current container level

incByteContainerLevel

```
public void incByteContainerLevel(int byteContainerLevel)
```

Fake container for processing

Parameters:

byteContainerLevel - current container level

isByteContainerEmpty

```
public boolean isByteContainerEmpty()
```

Fake container for processing

Returns:

is container full

(continued from last page)

isByteContainerFull

```
public boolean isByteContainerFull()
```

Fake container for processing

Returns:

is container full

isLongTimecode

```
public boolean isLongTimecode()
```

Get is a 32 bit timecode

Returns:

true if 32 bit timecode

setLongTimecode

```
public void setLongTimecode(boolean isLongTimecode)
```

Set is a 32 bit timecode

Parameters:

isLongTimecode - is a 32 bit timecode

isLastSentAbsTimecode

```
public boolean isLastSentAbsTimecode()
```

setLastSentAbsTimecode

```
public void setLastSentAbsTimecode(boolean isLastSentAbsTimecode)
```

isObjectEncodingAMF3

```
public boolean isObjectEncodingAMF3()
```

isObjectEncodingAMF0

```
public boolean isObjectEncodingAMF0()
```

setObjectEncoding

```
public void setObjectEncoding(int objectEncoding)
```

(continued from last page)

getObjectEncoding

```
public int getObjectEncoding()
```

getChunkCounter

```
public long getChunkCounter()
```

setChunkCounter

```
public void setChunkCounter(long chunkCounter)
```

com.wowza.wms.amf Class AMFObjChunk

java.lang.Object

└─com.wowza.wms.amf.AMFObjChunk

```
public class AMFObjChunk
extends Object
```

Field Summary

public	buffer
public	len
public	offset

Constructor Summary

public	AMFObjChunk (byte[] buffer, int offset, int len)
--------	--

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

buffer

```
public byte buffer
```

offset

```
public int offset
```

len

```
public int len
```

Constructors

(continued from last page)

AMFObjChunk

```
public AMFObjChunk(byte[] buffer,  
                  int offset,  
                  int len)
```

com.wowza.wms.amf

Class AMPacket

java.lang.Object

└─com.wowza.wms.amf.AMPacket

```
public class AMPacket
extends Object
```

AMFPacket: data container for data being transferred to and from the server from the Flash client. AMPacket is also used to store data read/written to/from an flv file.

Constructor Summary

public	AMPacket() Create new empty packet
public	AMPacket(int type, int src, int size) Create new packet with given values
public	AMPacket(int type, int src, byte[] data) Create new packet with given values

Method Summary

int	addData(byte[] data, int offset, int size) Add data to the packet
int	addDataEx(byte[] srcData, int srcOffset, int destOffset, int srcBytes) Add data to the packet
static int	calcTotalPacketSize(int packetSize, int headerSize, int chunkSize, int amfNumber, boolean isLongTimecode) Calculate the total packet size for given packet parameters
AMPacket	clone()
AMPacket	clone(boolean doCopy)
boolean	containsExtraDataKey(String key) Return true if extra data contains the key
static AMPacket	deserializePacket(byte[] buffer)
java.util.Map	extraDataClone() Deep copy the extra data and return as a Map
static java.util.Map	extraDataClone(java.util.Map extraData)
long	getAbsTimecode() Get absolute timecode (milliseconds)

byte[]	<code>getData()</code> Get data as byte[]
java.nio.ByteBuffer	<code>getDataBuffer()</code> Get data as ByteBuffer
<code>IAMFPacketExtraData</code>	<code>getExtraData(String key)</code> Get extra data value
java.util.Set	<code>getExtraDataKeys()</code> Get a set of the extra data keys
int	<code>getFirstByte()</code> Get first byte of data (used to peek into packet)
Object	<code>getLock()</code> Get lock (mostly used for extra data access)
int	<code>getMissing()</code> Get the number of bytes remaining unfilled in the packet
int	<code>getSecondByte()</code> Get second byte of data (used to peek into packet)
long	<code>getSeq()</code> Get packet sequence number.
int	<code>getSize()</code> Get packet size
int	<code>getSrc()</code> Get stream id (0 if not stream data)
long	<code>getTimecode()</code> Get timecode (milliseconds) relative to the <i>previous</i> packet.
int	<code>getType()</code> Get content type IVHost.CONTENTTYPE_*
boolean	<code>isAudio()</code> Is this an audio packet IVHost.CONTENTTYPE_AUDIO
boolean	<code>isExtraData()</code> Returns true if this packet contains any extra data
boolean	<code>isVideo()</code> Is this an audio packet IVHost.CONTENTTYPE_VIDEO
void	<code>putAllExtraData(java.util.Map entries)</code> Add extra data key value pairs
<code>IAMFPacketExtraData</code>	<code>putExtraData(String key, <code>IAMFPacketExtraData</code> value)</code> Add extra data value
byte[]	<code>serializePacket()</code>
void	<code>setAbsTimecode(long absTimecode)</code> Set absolute timecode (milliseconds)

void	<code>setDataBuffer</code> (byte[] data) Set the data buffer to a byte array
void	<code>setDataBuffer</code> (java.nio.ByteBuffer data) Set the data for this packet
void	<code>setSeq</code> (long seq) Set packet sequence.
void	<code>setSize</code> (int size) Set packet size
void	<code>setSrc</code> (int src) Set stream id (0 if not stream data)
void	<code>setTimecode</code> (long timecode) Set timecode (milliseconds) relative,
void	<code>setTimecodes</code> (long timecode, long absTimecode) Set both relative and absolute timecode in one call (milliseconds)
void	<code>setType</code> (int type) Set content type IVHost.CONTENTTYPE_*
String	<code>toString</code> () Return object as formatted string
void	<code>truncatePacket</code> (int newSize)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

AMFPacket

```
public AMFPacket()
```

Create new empty packet

AMFPacket

```
public AMFPacket(int type,  
                 int src,  
                 int size)
```

Create new packet with given values

Parameters:

type - packet content type: IVHost.CONTENTTYPE_*
src - stream id
size - packet size (bytes)

(continued from last page)

AMFPacket

```
public AMFPacket(int type,  
                 int src,  
                 byte[] data)
```

Create new packet with given values

Parameters:

`type` - packet content type: IVHost.CONTENTTYPE_*
`src` - stream id
`data` - data buffer

Methods

clone

```
public AMFPacket clone()
```

clone

```
public AMFPacket clone(boolean doCopy)
```

getSize

```
public int getSize()
```

Get packet size

Returns:

packet size

setSize

```
public void setSize(int size)
```

Set packet size

Parameters:

`size` - packet size

truncatePacket

```
public void truncatePacket(int newSize)
```

getMissing

```
public int getMissing()
```

Get the number of bytes remaining unfilled in the packet

Returns:

number of bytes unfilled in packet

setDataBuffer

```
public void setDataBuffer(java.nio.ByteBuffer data)
```

Set the data for this packet

Parameters:

data - byte buffer that holds packet data

setDataBuffer

```
public void setDataBuffer(byte[] data)
```

Set the data buffer to a byte array

Parameters:

data - data buffer byte array

addData

```
public int addData(byte[] data,  
                  int offset,  
                  int size)
```

Add data to the packet

Parameters:

data - byte buffer with data
offset - offset in byte buffer
size - size of data

Returns:

number of bytes unfilled in packet

getType

```
public int getType()
```

Get content type IVHost.CONTENTTYPE_*

Returns:

content type

setType

```
public void setType(int type)
```

Set content type IVHost.CONTENTTYPE_*

Parameters:

type - content type

getSrc

```
public int getSrc()
```

Get stream id (0 if not stream data)

(continued from last page)

Returns:

stream id

setSrc

```
public void setSrc(int src)
```

Set stream id (0 if not stream data)

Parameters:

src - stream id

toString

```
public String toString()
```

Return object as formatted string

getTimecode

```
public long getTimecode()
```

Get timecode (milliseconds) relative to the *previous* packet.

Returns:

timecode (milliseconds)

setTimecodes

```
public void setTimecodes(long timecode,  
    long absTimecode)
```

Set both relative and absolute timecode in one call (milliseconds)

Note this time is relative to the *previous* packet.

Parameters:

timecode - relative timecode (milliseconds)

absTimecode - absolute timecode (milliseconds)

setTimecode

```
public void setTimecode(long timecode)
```

Set timecode (milliseconds) relative,

Note this time is relative to the *previous* packet.

Parameters:

timecode - timecode (milliseconds)

getAbsTimecode

```
public long getAbsTimecode()
```

Get absolute timecode (milliseconds)

Returns:

absolute timecode

setAbsTimecode

```
public void setAbsTimecode(long absTimecode)
```

Set absolute timecode (milliseconds)

Parameters:

absTimecode - absolute timecode

getSeq

```
public long getSeq()
```

Get packet sequence number. Used for live streams to keep track of packet ordering.

Returns:

packet sequence

setSeq

```
public void setSeq(long seq)
```

Set packet sequence. Used for live streams to keep track of packet ordering.

Parameters:

seq - packet sequence

isAudio

```
public boolean isAudio()
```

Is this an audio packet IVHost.CONTENTTYPE_AUDIO

Returns:

true if audio packet

isVideo

```
public boolean isVideo()
```

Is this an audio packet IVHost.CONTENTTYPE_VIDEO

Returns:

true if video packet

getDataBuffer

```
public java.nio.ByteBuffer getDataBuffer()
```

Get data as ByteBuffer

Returns:

data as ByteBuffer

getData

```
public byte[] getData()
```

(continued from last page)

Get data as byte[]

Returns:

data as byte[] null if no data

getFirstByte

```
public int getFirstByte()
```

Get first byte of data (used to peek into packet)

Returns:

first byte of data in packet

getSecondByte

```
public int getSecondByte()
```

Get second byte of data (used to peek into packet)

Returns:

second byte of data in packet

calcTotalPacketSize

```
public static int calcTotalPacketSize(int packetSize,  
    int headerSize,  
    int chunkSize,  
    int amfNumber,  
    boolean isLongTimecode)
```

Calculate the total packet size for given packet parameters

Parameters:

packetSize - data size
headerSize - header size
chunkSize - chunk size
amfNumber - amf number
isLongTimecode - is long timecode

Returns:

total bytes

addDataEx

```
public int addDataEx(byte[] srcData,  
    int srcOffset,  
    int destOffset,  
    int srcBytes)
```

Add data to the packet

Parameters:

srcData -- source byte buffer with data
srcOffset -- start copying from source buffer at this offset
destOffset -- copy into destination buffer from this offset
srcBytes -- size of data to copy

Returns:

(continued from last page)

number of bytes unfilled in packet

serializePacket

```
public byte[] serializePacket()
```

deserializePacket

```
public static AMFPacket deserializePacket(byte[] buffer)
```

getLock

```
public Object getLock()
```

Get lock (mostly used for extra data access)

Returns:

lock

putExtraData

```
public IAMFPacketExtraData putExtraData(String key,  
    IAMFPacketExtraData value)
```

Add extra data value

Parameters:

key - key

value - extra data

Returns:

extra data currently at that key

putAllExtraData

```
public void putAllExtraData(java.util.Map entries)
```

Add extra data key value pairs

Parameters:

entries - extra data key value pairs

getExtraData

```
public IAMFPacketExtraData getExtraData(String key)
```

Get extra data value

Parameters:

key - key

Returns:

extra data value

getExtraDataKeys

```
public java.util.Set getExtraDataKeys()
```

Get a set of the extra data keys

Returns:

extra data keys

containsExtraDataKey

```
public boolean containsExtraDataKey(String key)
```

Return true if extra data contains the key

Parameters:

key - key

Returns:

true if extra data contains the key

isExtraData

```
public boolean isExtraData()
```

Returns true if this packet contains any extra data

Returns:

true if this packet contains any extra data

extraDataClone

```
public static java.util.Map extraDataClone(java.util.Map extraData)
```

extraDataClone

```
public java.util.Map extraDataClone()
```

Deep copy the extra data and return as a Map

Returns:

deep copy of extra data

com.wowza.wms.amf Interface IAMFDataObj

All Known Implementing Classes:

[AMFDataObj](#)

public interface **IAMFDataObj**
extends

Method Summary

boolean	<code>containsKey</code> (String name) Return true if the object/array contains key
AMFData	<code>get</code> (int index) Return the object at a particular index.
AMFData	<code>get</code> (String name) Return the object at a particular key.
boolean	<code>getBoolean</code> (int index) Get item at index return as boolean
boolean	<code>getBoolean</code> (String name) Get item at key return as boolean
byte	<code>getByte</code> (int index) Get item at index return as byte
byte	<code>getByte</code> (String name) Get item at key return as byte
java.util.Date	<code>getDate</code> (int index) Get item at index return as Date
java.util.Date	<code>getDate</code> (String name) Get item at key return as Date
double	<code>getDouble</code> (int index) Get item at index return as double
double	<code>getDouble</code> (String name) Get item at key return as double
float	<code>getFloat</code> (int index) Get item at index return as float
float	<code>getFloat</code> (String name) Get item at key return as float
int	<code>getInt</code> (int index) Get item at index return as int
int	<code>getInt</code> (String name) Get item at key return as int

String	<u>getKey</u> (int index) Return the key at a particular index.
java.util.List	<u>getKeys</u> () Return a list of all the keys (the list is a copy)
long	<u>getLong</u> (int index) Get item at index return as long
long	<u>getLong</u> (String name) Get item at key return as long
<u>AMFDataObj</u>	<u>getObject</u> (int index) Get item at index return as AMFDataObj
<u>AMFDataObj</u>	<u>getObject</u> (String name) Get item at key return as AMFDataObj
short	<u>getShort</u> (int index) Get item at index return as short
short	<u>getShort</u> (String name) Get item at key return as short
String	<u>getString</u> (int index) Get item at index return as String
String	<u>getString</u> (String name) Get item at key return as String
void	<u>put</u> (String name, <u>AMFData</u> data) Put or replace object at key
void	<u>put</u> (String name, boolean data) Put or replace boolean value at key (data will be wrapped in an AMFDataItem object)
void	<u>put</u> (String name, java.util.Date data) Put or replace date value at key (data will be wrapped in an AMFDataItem object)
void	<u>put</u> (String name, double data) Put or replace double value at key (data will be wrapped in an AMFDataItem object)
void	<u>put</u> (String name, int data) Put or replace int value at key (data will be wrapped in an AMFDataItem object)
void	<u>put</u> (String name, long data) Put or replace long value at key (data will be wrapped in an AMFDataItem object)
void	<u>put</u> (String name, String data) Put or replace string value at key (data will be wrapped in an AMFDataItem object)
<u>AMFData</u>	<u>remove</u> (int index) Remove element by index
<u>AMFData</u>	<u>remove</u> (String name) Remove element by key

(continued from last page)

Methods

containsKey

```
public boolean containsKey(String name)
```

Return true if the object/array contains key

Parameters:

name - key

Returns:

Return true the object/array contains key

put

```
public void put(String name,  
    AMFData data)
```

Put or replace object at key

Parameters:

name - key

data - object

put

```
public void put(String name,  
    String data)
```

Put or replace string value at key (data will be wrapped in an AMFDataItem object)

Parameters:

name - key

data - string value

put

```
public void put(String name,  
    double data)
```

Put or replace double value at key (data will be wrapped in an AMFDataItem object)

Parameters:

name - key

data - double value

put

```
public void put(String name,  
    int data)
```

Put or replace int value at key (data will be wrapped in an AMFDataItem object)

Parameters:

name - key

data - int value

put

```
public void put(String name,  
               long data)
```

Put or replace long value at key (data will be wrapped in an AMFDataItem object)

Parameters:

name - key
data - long value

put

```
public void put(String name,  
               java.util.Date data)
```

Put or replace date value at key (data will be wrapped in an AMFDataItem object)

Parameters:

name - key
data - date value

put

```
public void put(String name,  
               boolean data)
```

Put or replace boolean value at key (data will be wrapped in an AMFDataItem object)

Parameters:

name - key
data - boolean value

getKeys

```
public java.util.List getKeys()
```

Return a list of all the keys (the list is a copy)

Returns:

new list that contains one entry for each key

getKey

```
public String getKey(int index)
```

Return the key at a particular index.

Parameters:

index

Returns:

Return key at index or null if out of bounds

get

```
public AMFData get(String name)
```

(continued from last page)

Return the object at a particular key.

Parameters:

name - key

Returns:

Return object or null if out of bounds

get

```
public AMFData get(int index)
```

Return the object at a particular index.

Parameters:

index - index

Returns:

Return object or null if out of bounds

remove

```
public AMFData remove(String name)
```

Remove element by key

Parameters:

name - key

Returns:

removed object or null if not found

remove

```
public AMFData remove(int index)
```

Remove element by index

Parameters:

index - index

Returns:

removed object or null if not found

getString

```
public String getString(String name)
```

Get item at key return as String

Parameters:

name - key

Returns:

Return item as String or null if out of bounds

(continued from last page)

getInt

```
public int getInt(String name)
```

Get item at key return as int

Parameters:

name - key

Returns:

Return item as int or 0 if out of bounds

getLong

```
public long getLong(String name)
```

Get item at key return as long

Parameters:

name - key

Returns:

Return item as long or 0 if out of bounds

getShort

```
public short getShort(String name)
```

Get item at key return as short

Parameters:

name - key

Returns:

Return item as short or 0 if out of bounds

getDouble

```
public double getDouble(String name)
```

Get item at key return as double

Parameters:

name - key

Returns:

Return item as double or 0 if out of bounds

getFloat

```
public float getFloat(String name)
```

Get item at key return as float

Parameters:

name - key

Returns:

Return item as float or 0 if out of bounds

getBytes

```
public byte getBytes(String name)
```

Get item at key return as byte

Parameters:

name - key

Returns:

Return item as byte or 0 if out of bounds

getBoolean

```
public boolean getBoolean(String name)
```

Get item at key return as boolean

Parameters:

name - key

Returns:

Return item as boolean or false if out of bounds

getDate

```
public java.util.Date getDate(String name)
```

Get item at key return as Date

Parameters:

name - key

Returns:

Return item as Date or null if out of bounds

getObject

```
public AMFDataObj getObject(String name)
```

Get item at key return as AMFDataObj

Parameters:

name - key

Returns:

Return item as AMFDataObj or null if out of bounds

getString

```
public String getString(int index)
```

Get item at index return as String

Parameters:

index - index

(continued from last page)

Returns:

Return item as String or null if out of bounds

getInt

```
public int getInt(int index)
```

Get item at index return as int

Parameters:

index - index

Returns:

Return item as int or 0 if out of bounds

getLong

```
public long getLong(int index)
```

Get item at index return as long

Parameters:

index - index

Returns:

Return item as long or 0 if out of bounds

getShort

```
public short getShort(int index)
```

Get item at index return as short

Parameters:

index - index

Returns:

Return item as short or 0 if out of bounds

getByte

```
public byte getByte(int index)
```

Get item at index return as byte

Parameters:

index - index

Returns:

Return item as byte or 0 if out of bounds

getDouble

```
public double getDouble(int index)
```

Get item at index return as double

Parameters:

(continued from last page)

index - index

Returns:

Return item as double or 0 if out of bounds

getFloat

```
public float getFloat(int index)
```

Get item at index return as float

Parameters:

index - index

Returns:

Return item as float or 0 if out of bounds

getBoolean

```
public boolean getBoolean(int index)
```

Get item at index return as boolean

Parameters:

index - index

Returns:

Return item as boolean or false if out of bounds

getDate

```
public java.util.Date getDate(int index)
```

Get item at index return as Date

Parameters:

index - index

Returns:

Return item as Date or null if out of bounds

getObject

```
public AMFDataObj getObject(int index)
```

Get item at index return as AMFDataObj

Parameters:

index - index

Returns:

Return item as AMFDataObj or null if out of bounds

com.wowza.wms.amf Interface IAMFExternalizable

public interface **IAMFExternalizable**
extends

Method Summary

void	deserialize (AMFDataObj amfObj, java.nio.ByteBuffer data, AMFDataContextDeserialize context)
String	getClassName ()
void	serialize (AMFDataObj amfObj, java.io.DataOutputStream out, AMFDataContextSerialize context)
void	setClassName (String className)

Methods

deserialize

```
public void deserialize(AMFDataObj amfObj,  
    java.nio.ByteBuffer data,  
    AMFDataContextDeserialize context)
```

serialize

```
public void serialize(AMFDataObj amfObj,  
    java.io.DataOutputStream out,  
    AMFDataContextSerialize context)
```

getClassName

```
public String getClassName()
```

setClassName

```
public void setClassName(String className)
```

com.wowza.wms.amf Interface IAMFPacketExtraData

public interface **IAMFPacketExtraData**
extends

Method Summary

IAMFPacketExtraData	clone() Return a deep copy of the extra data item
-------------------------------------	--

Methods

clone

public [IAMFPacketExtraData](#) **clone()**

Return a deep copy of the extra data item

Returns:

deep copy of the extra data item

Package

com.wowza.wms.application

com.wowza.wms.application Interface IApplication

public interface **IApplication**
extends

IApplication: public interface to Application object

Field Summary

public static final	<u>DEFAULT_APPLICATION_NAME</u> Value: _defapp_
---------------------	---

Method Summary

void	<u>addApplicationInstanceListener</u> (<u>IApplicationInstanceNotify</u> applicationInstanceListener) Add applicationInstance listener.
<u>IApplicationInstance</u>	<u>getAppInstance</u> (String name) Get applicationInstance object by name
java.util.List	<u>getAppInstanceNames</u> () Get a list of application instance names
String	<u>getApplicationPath</u> () Get the root path for application
String	<u>getConfigPath</u> () Get full path to Application.xml file
<u>ConnectionCounter</u>	<u>getConnectionCounter</u> () Get the connectionCounter for application
ConnectionCounterSimple	<u>getConnectionCounter</u> (int counterIndex) Get the connectionCounter for application for application for a specific technology (see IVHost.COUNTER_*)
String	<u>getDateStarted</u> () Get date application started
<u>IOPerformanceCounter</u>	<u>getIoPerformanceCounter</u> () Get the performance counter for application
<u>IOPerformanceCounter</u>	<u>getIoPerformanceCounter</u> (int counterIndex) Get the performance counter for application for a specific technology (see IVHost.COUNTER_*)
String	<u>getName</u> () Get the name of application
<u>WMSProperties</u>	<u>getProperties</u> () Get application properties

void	<u>getProtocolUsage</u> (boolean[] protocolsInUse) Get the protocols in use by this application instance (see IApplicationInstance.PROTOCOLUSAGE_*)
String	<u>getTimeRunning</u> () Get time application running
double	<u>getTimeRunningSeconds</u> () Get time running in seconds
<u>IVHost</u>	<u>getVHost</u> () Get the parent vHost object
boolean	<u>isAppInstanceLoaded</u> (String name) Return true if application instance is loaded
boolean	<u>isRunning</u> () Returns true if the application is in running state
String	<u>readAppConfig</u> (String sName) Method to read xml config file..
void	<u>removeAppInstance</u> (<u>IApplicationInstance</u> appInstance) Disconnect all clients connected to an application instance and remove it from the IApplication application list.
void	<u>removeApplicationInstanceListener</u> (<u>IApplicationInstanceNotify</u> applicationInstanceListener) Remove applicationInstance listener.
void	<u>setName</u> (String name) Set name of application
void	<u>shutdown</u> (boolean isServerShutdown) shutdown application
void	<u>shutdownAppInstance</u> (String appInstanceName) Shutdown an application instance by name.
boolean	<u>writeAppConfig</u> (String sName, String data) Method to write xml config file..

Fields

DEFAULT_APPLICATION_NAME

public static final java.lang.String **DEFAULT_APPLICATION_NAME**

Constant value: **_defapp_**

Methods

shutdown

public void **shutdown**(boolean isServerShutdown)

shutdown application

(continued from last page)

Parameters:

isServerShutdown - true if due to shutdown of server

getApplicationPath

```
public String getApplicationPath()
```

Get the root path for application

Returns:

root path for application

getConfigPath

```
public String getConfigPath()
```

Get full path to Application.xml file

Returns:

full path to Application.xml file

getAppInstance

```
public IApplicationInstance getAppInstance(String name)
```

Get applicationInstance object by name

Parameters:

name - applicationInstance name

Returns:

appliationInstance object

isAppInstanceLoaded

```
public boolean isAppInstanceLoaded(String name)
```

Return true if application instance is loaded

Parameters:

name - applicationInstance name

Returns:

true if application instance is loaded

getName

```
public String getName()
```

Get the name of application

Returns:

name of application

setName

```
public void setName(String name)
```

(continued from last page)

Set name of application

Parameters:

name - name of application

getVHost

```
public IVHost getVHost()
```

Get the parent vHost object

Returns:

parent vHost

getProperties

```
public WMSProperties getProperties()
```

Get application properties

Returns:

application properties

addApplicationInstanceListener

```
public void addApplicationInstanceListener(IApplicationInstanceNotify  
applicationInstanceListener)
```

Add applicationInstance listener. Will be invoked each time applicationInstance created/deleted

Parameters:

applicationInstanceListener - applicationInstance listener

removeApplicationInstanceListener

```
public void removeApplicationInstanceListener(IApplicationInstanceNotify  
applicationInstanceListener)
```

Remove applicationInstance listener. Will be invoked each time applicationInstance created/deleted

Parameters:

applicationInstanceListener - applicationInstance listener

getConnectionCounter

```
public ConnectionCounter getConnectionCounter()
```

Get the connectionCounter for application

Returns:

connectionCounter for application

getConnectionCounter

```
public ConnectionCounterSimple getConnectionCounter(int counterIndex)
```

Get the connectionCounter for application for application for a specific technology (see IVHost.COUNTER_*)

(continued from last page)

Parameters:

counterIndex - counter index (see IVHost.COUNTER_*)

Returns:

connection ocunter

getIoPerformanceCounter

```
public IoPerformanceCounter getIoPerformanceCounter()
```

Get the performance counter for application

Returns:

performance counter for application

getIoPerformanceCounter

```
public IoPerformanceCounter getIoPerformanceCounter(int counterIndex)
```

Get the performance counter for application for a specific technology (see IVHost.COUNTER_*)

Parameters:

counterIndex - counter index (see IVHost.COUNTER_*)

Returns:

performance counter

getDateStarted

```
public String getDateStarted()
```

Get date application started

Returns:

date application started

getTimeRunning

```
public String getTimeRunning()
```

Get time application running

Returns:

time application running

getTimeRunningSeconds

```
public double getTimeRunningSeconds()
```

Get time running in seconds

Returns:

time running in seconds

getAppInstanceNames

```
public java.util.List getAppInstanceNames()
```

(continued from last page)

Get a list of application instance names

Returns:

list of application instance names

removeAppInstance

public void **removeAppInstance**([IApplicationInstance](#) appInstance)

Disconnect all clients connected to an application instance and remove it from the IApplication application list. The proper way to call this is:

```
public void shutdownAppInstance(IApplicationInstance appInstance)
{
    IVHost vhost = appInstance.getVHost();
    IApplication app = appInstance.getApplication();

    WMSReadWriteLock appLock = vhost.getApplicationLock();
    appLock.writeLock().lock();
    try
    {
        app.removeAppInstance(appInstance);
    }
    catch (Exception e)
    {
        WMSLoggerFactory.getLogger(Application.class).error("Application.shutdownAppInstance ",
        e);
    }
    finally
    {
        appLock.writeLock().unlock();
    }
}
```

Parameters:

appInstance - application instance to remove

shutdownAppInstance

public void **shutdownAppInstance**(String appInstanceName)

Shutdown an application instance by name. This will disconnect all clients connected to this application instance.

Parameters:

appInstanceName - application instance name

(continued from last page)

readAppConfig

```
public String readAppConfig(String sName)
```

Method to read xml config file..

writeAppConfig

```
public boolean writeAppConfig(String sName,  
                               String data)
```

Method to write xml config file..

getProtocolUsage

```
public void getProtocolUsage(boolean[] protocolsInUse)
```

Get the protocols in use by this application instance (see IApplicationInstance.PROTOCOLUSAGE_*)

isRunning

```
public boolean isRunning()
```

Returns true if the application is in running state

Returns:

true if the application is in running state

com.wowza.wms.application Interface IApplicationInstance

public interface **IApplicationInstance**
extends

IApplicationInstance: public interface to ApplicationInstance object

Field Summary

public static final	DEFAULT_APPINSTANCE_NAME Value: _definst_
public static final	PROTOCOLUSAGE_CUPERTINO Value: 7
public static final	PROTOCOLUSAGE_MC_LIVEREPEATER Value: 44
public static final	PROTOCOLUSAGE_MC_RTP Value: 45
public static final	PROTOCOLUSAGE_MC_SHOUTCAST Value: 43
public static final	PROTOCOLUSAGE_MEDIACACHE Value: 20
public static final	PROTOCOLUSAGE_MPEGDASH Value: 10
public static final	PROTOCOLUSAGE_ORIGINCUPERTINO Value: 15
public static final	PROTOCOLUSAGE_ORIGINMPEGDASH Value: 18
public static final	PROTOCOLUSAGE_ORIGINSANJOSE Value: 17
public static final	PROTOCOLUSAGE_ORIGINSMOOTH Value: 16
public static final	PROTOCOLUSAGE_PUSHUBLISHING Value: 19

public static final	<u>PROTOCOLUSAGE_RTMP</u> Value: 0
public static final	<u>PROTOCOLUSAGE_RTMP_IN</u> Value: 21
public static final	<u>PROTOCOLUSAGE_RTMP_OUT</u> Value: 22
public static final	<u>PROTOCOLUSAGE_RTMPE</u> Value: 4
public static final	<u>PROTOCOLUSAGE_RTMPE_IN</u> Value: 29
public static final	<u>PROTOCOLUSAGE_RTMPE_OUT</u> Value: 30
public static final	<u>PROTOCOLUSAGE_RTMPS</u> Value: 2
public static final	<u>PROTOCOLUSAGE_RTMPS_IN</u> Value: 25
public static final	<u>PROTOCOLUSAGE_RTMPS_OUT</u> Value: 26
public static final	<u>PROTOCOLUSAGE_RTMPT</u> Value: 1
public static final	<u>PROTOCOLUSAGE_RTMPT_IN</u> Value: 23
public static final	<u>PROTOCOLUSAGE_RTMPT_OUT</u> Value: 24
public static final	<u>PROTOCOLUSAGE_RTMPTE</u> Value: 5
public static final	<u>PROTOCOLUSAGE_RTMPTE_IN</u> Value: 31
public static final	<u>PROTOCOLUSAGE_RTMPTE_OUT</u> Value: 32
public static final	<u>PROTOCOLUSAGE_RTMPTS</u> Value: 3

public static final	<u>PROTOCOLUSAGE_RTPTS_IN</u> Value: 27
public static final	<u>PROTOCOLUSAGE_RTPTS_OUT</u> Value: 28
public static final	<u>PROTOCOLUSAGE_RTP</u> Value: 6
public static final	<u>PROTOCOLUSAGE_RTP_IN</u> Value: 33
public static final	<u>PROTOCOLUSAGE_RTP_OUT</u> Value: 34
public static final	<u>PROTOCOLUSAGE_RTSP_IN</u> Value: 35
public static final	<u>PROTOCOLUSAGE_RTSP_OUT</u> Value: 36
public static final	<u>PROTOCOLUSAGE_RTSPULL_IN</u> Value: 46
public static final	<u>PROTOCOLUSAGE_RTSPULL_OUT</u> Value: 47
public static final	<u>PROTOCOLUSAGE_SANJOSE</u> Value: 9
public static final	<u>PROTOCOLUSAGE_SMOOTH</u> Value: 8
public static final	<u>PROTOCOLUSAGE_TOTAL</u> Value: 48
public static final	<u>PROTOCOLUSAGE_WEBM</u> Value: 11
public static final	<u>PROTOCOLUSAGE_WOWZ</u> Value: 12
public static final	<u>PROTOCOLUSAGE_WOWZ_IN</u> Value: 37
public static final	<u>PROTOCOLUSAGE_WOWZ_OUT</u> Value: 38

public static final	PROTOCOLUSAGE_WOWZE Value: 13
public static final	PROTOCOLUSAGE_WOWZE_IN Value: 39
public static final	PROTOCOLUSAGE_WOWZE_OUT Value: 40
public static final	PROTOCOLUSAGE_WOWZS Value: 14
public static final	PROTOCOLUSAGE_WOWZS_IN Value: 41
public static final	PROTOCOLUSAGE_WOWZS_OUT Value: 42

Method Summary

void	addClientListener (IClientNotify clientListener) Add client listener.
void	addDvrRecorderListener (ILiveStreamDvrRecorderActionNotify listener) Add a Dvr Recorder listener (see: ILiveStreamDvrRecorderActionNotify)
void	addDvrStreamManagerListener (IDvrStreamManagerActionNotify listener) Add a Dvr Application Store Manager listener (see: IDvrStoreActionNotify)
void	addHTTPListener (IHTTPSessionNotify httpListener) Add listner to be notified of HTTPSession create and destroy.
void	addHTTPStreamerSession (IHTTPStreamerSession httpStreamerSession) Add a HTTPStreamerSession to this application instance
void	addLiveStreamPacketizerListener (ILiveStreamPacketizerActionNotify LiveStreamPacketizerListener) Add a Live Stream Packetizer listener (see: ILiveStreamPacketizerActionNotify)
void	addLiveStreamTranscoderListener (ILiveStreamTranscoderNotify LiveStreamTranscoderListener) Add a live stream transcoder listener
void	addMediaCasterListener (IMediaCasterNotify mediaCasterListener) Add mediaCaster listener.
void	addMediaCasterListener (IMediaCasterNotify2 mediaCasterListener) Add mediaCaster listener.
void	addMediaReaderListener (IMediaReaderActionNotify mediaReaderListener) Add media reader listener.
void	addMediaStreamListener (IMediaStreamNotify mediaStreamListener) Add mediaStream listener.

void	<u>addMediaWriterListener</u> (<u>IMediaWriterActionNotify</u> listener) Add a MediaWriter listener class.
void	<u>addModuleListener</u> (<u>IModuleNotify</u> moduleListener) Add module listener.
void	<u>addPlayStreamByName</u> (<u>IMediaStream</u> stream, String name) Add a media stream to the list of streams that are listening for a published stream
void	<u>addPublisher</u> (<u>Publisher</u> publisher) Add a server side publisher to this application instance
void	<u>addPushPublishSession</u> (<u>IMediaStream</u> stream, <u>IPushPublishSession</u> pushPublishSession) Add a push publish session to the list of active push publish sessions
void	<u>addPushPublishSessionListener</u> (<u>IPushPublishSessionNotify</u> pushPublishSessionListener) Add a push publish session listener (see IPushPublishSessionNotify)
void	<u>addRTPIncomingDatagramPortAll</u> () Allow all incoming RTP UDP ports for this application instance
void	<u>addRTPIncomingDatagramPortRange</u> (int startPort, int endPort) Add a port range to the list of valid incoming RTP UDP ports
void	<u>addRTPListener</u> (<u>IRTPSessionNotify</u> rtpListener) Add listner to be notified of RTPSession create and destroy.
void	<u>addRTPSession</u> (<u>RTPSession</u> rtpSession) Add an RTP session to this application instance
void	<u>addSharedObjectListener</u> (<u>ISharedObjectNotify</u> sharedObjectListener, boolean isPersistent) Add sharedObject listener.
void	<u>broadcastMsg</u> (java.util.List clientList, String handlerName) Broadcast a message to a specific list of clients connected to this application instance
void	<u>broadcastMsg</u> (java.util.List clientList, String handlerName, Object[] params) Broadcast a message to a specific list of clients connected to this application instance
void	<u>broadcastMsg</u> (String handlerName, Object[] params) Broadcast a message to all clients connected to this applicationInstance
boolean	<u>containsDvrRecorder</u> (String dvrRecorder) Does this application instance contain a references to this DVR recorder.
boolean	<u>containsHTTPStreamer</u> (String httpStreamer) Does this application instance allow streaming of a given HTTPStreamer
boolean	<u>containsLiveStreamPacketizer</u> (String liveStreamPacketizer) Does this application instance contain a references to this live stream packetizer.
boolean	<u>containsLiveStreamTranscoder</u> (String liveStreamTranscoder) Return true if this application instance contains the transcoder name
String	<u>decodeStorageDir</u> (String storageDir) This function will take a storage path that uses variables and expand the variables based on the context.

String[]	<u>getAllowDomains()</u> Get the list of domain names used to control access to this application.
<u>IApplication</u>	<u>getApplication()</u> Get parent application
int	<u>getApplicationInstanceTouchTimeout()</u> Get the application instance touch timeout (milliseconds).
int	<u>getApplicationTimeout()</u> Get application timeout (milliseconds)
<u>IClient</u>	<u>getClient(int index)</u> Deprecated. Get the client connection at index. This method is deprecated. It is best to use <u>getClient()</u> to return a List objects.
<u>IClient</u>	<u>getClientById(int index)</u> Get a client connection by the client Id
int	<u>getClientCount()</u> Get number of client connections currently connected to applicationInstance
int	<u>getClientCountTotal()</u> Get number of client connections in total that have connected to this applicationInstance
int	<u>getClientIdleFrequency()</u> Get default client idle frequency (milliseconds)
java.util.List	<u>getClients()</u> Get the set of clients currently connected to this application instance (replaces getClient(index))
edu.emory.mathcs.backport.java.util.concurrent.locks.WMSReadWriteLock	<u>getClientsLockObj()</u> Get the read/write lock for this application instance
<u>ConnectionCounter</u>	<u>getConnectionCounter()</u> Get the connectionCounter for applicationInstance
ConnectionCounterSimple	<u>getConnectionCounter(int counterIndex)</u> Get the connectionCounter for applicationInstance for a specific technology (see IVHost.COUNTER_*)
String	<u>getContextStr()</u> Returns the application context string in the form [application]/[appInstance].
String	<u>getDateStarted()</u> Get date applicationInstance started
DvrApplicationContext	<u>getDvrApplicationContext()</u> Get live stream dvr application context
<u>WMSProperties</u>	<u>getDvrProperties()</u> Get the property collection of DVR settings that are specific to this application instance.
String	<u>getDvrRecorderList()</u> Get the comma separated list of Dvr Recorder names being used by this application (see conf/Dvr.xml)

<u>IHTTPStreamerApplicationContext</u>	<u>getHTTPStreamerApplicationContext</u> (String httpStreamName, boolean doCreate) Get the HTTPStreamer application context for a given HTTPStreamer adapter
String	<u>getHTTPStreamerList</u> () Get the comma separated list of HTTPStreamers names being used by this application (see conf/HTTPStreamers.xml)
<u>WMSProperties</u>	<u>getHTTPStreamerProperties</u> () Get the property collection of HTTPStreamer settings that are specific to this application instance
int	<u>getHTTPStreamerSessionCount</u> () Get the current number of HTTPStreamerSessions associated with this application instance
int	<u>getHTTPStreamerSessionCount</u> (int protocol) Get the current number of HTTPStreamerSessions associated with this application instance by protocol.
int	<u>getHTTPStreamerSessionCount</u> (int protocol, String streamName) Get the current number of HTTPStreamerSessions associated with this application instance and stream name by protocol .
int	<u>getHTTPStreamerSessionCount</u> (String streamName) Get the current number of HTTPStreamerSessions associated with this application instance and stream name
java.util.Map	<u>getHTTPStreamerSessionCountsByName</u> (int protocol) Get a map of session counts by name for a given protocol
java.util.List	<u>getHTTPStreamerSessions</u> () Get the HTTPStreamerSessions associated with this application instance
java.util.List	<u>getHTTPStreamerSessions</u> (int protocol) Get the HTTPStreamerSessions associated with this application instance by protocol.
java.util.List	<u>getHTTPStreamerSessions</u> (int protocol, String streamName) Get the HTTPStreamerSessions associated with this application instance for a stream name by protocol.
java.util.List	<u>getHTTPStreamerSessions</u> (String streamName) Get the HTTPStreamerSessions associated with this application instance for a stream name
<u>IOPerformanceCounter</u>	<u>getIOPerformanceCounter</u> () Get the performance counter for applicationInstance
<u>IOPerformanceCounter</u>	<u>getIOPerformanceCounter</u> (int counterIndex) Get the performance counter for applicationInstance for a specific technology (see IVHost.COUNTER_*)
long	<u>getLastTouchTime</u> () Get the last time the instance was touched (milliseconds)
<u>ILiveStreamDvrRecorderControl</u>	<u>getLiveStreamDvrRecorderControl</u> () Get the Live Stream DVR Recorder Controller.
<u>ILiveStreamPacketizerControl</u>	<u>getLiveStreamPacketizerControl</u> () Get the Live Stream Packetizer Controller.

String	getLiveStreamPacketizerList() Get the comma separated list of LiveStreamPacketizers names being used by this application (see conf/LiveStreamPacketizers.xml)
WMSProperties	getLiveStreamPacketizerProperties() Get the property collection of LiveStreamPacketizer settings that are specific to this application instance
ILiveStreamTranscoderControl	getLiveStreamTranscoderControl() Get the Live Stream Transcoder Contoller.
String	getLiveStreamTranscoderList() Get comma separated list of transcoders to use for this application instance
WMSProperties	getManagerProperties() Get Application's manager's properties collection
int	getMaximumPendingReadBytes() Set maximum number of bytes a client connection can have waiting to be written before the connection is terminated.
int	getMaximumPendingWriteBytes() Get maximum number a bytes a client connection can have waiting to be sent before the connection is terminated.
int	getMaximumSetBufferTime() Get maximum number of milliseconds allowed for the NetStream.setBufferTime(secs) call.
int	getMaxStorageDirDepth() Maximum folder depth allowed for the StreamStorageDir and SharedObjectStorageDir paths
java.util.List	getMediaCacheFilters() Get the list of allowed MediaCache sources.
WMSProperties	getMediaCasterProperties() Get the property collection of media caster settings that are specific to this application instance
int	getMediacasterRTPRTSPRTPTransportMode() RTP MediaCaster RTSP/RTP transport mode.
MediaCasterStreamMap	getMediaCasterStreams() Get the media caster streams attached to this application instance
IMediaCasterValidateMediaCaster	getMediaCasterValidator() Get the MediaCaster validator interface for this application instance
IMediaListProvider	getMediaListProvider() Get the current media list provider.
int	getMediaReaderContentType(String mediaType) Get the content type of a media stream name prefix (see IMediaReader.CONTENTTYPE_*)
WMSProperties	getMediaReaderProperties() Get the property collection of media reader settings that are specific to this application instance
WMSProperties	getMediaWriterProperties() Get the property collection of media reader settings that are specific to this application instance
ModuleFunctions	getModFunctions() Get list of application modules

Object	<u>getModuleInstance</u> (String name) Get the instance of the module class for this application instance.
ModuleList	<u>getModuleList</u> () Get the list of loaded modules.
String	<u>getName</u> () Get applicationInstance name
int	<u>getPingTimeout</u> () Get ping timeout (milliseconds)
int	<u>getPlayStreamCount</u> (String streamName) Get the number of Flash players playing a given stream name
java.util.Map	<u>getPlayStreamCountsByName</u> () Get a map of stream names to number of Flash players playing the stream name
java.util.List	<u>getPlayStreamsByName</u> (String name) Get a list of media streams that are listening for published stream.
<u>WMSProperties</u>	<u>getProperties</u> () Get applicationInstance properties
boolean[]	<u>getProtocolUsage</u> () Get the protocols in use by this application instance (see IApplicationInstance.PROTOCOLUSAGE_*)
void	<u>getProtocolUsage</u> (boolean[] protocolsInUse) Get the protocols in use by this application instance (see IApplicationInstance.PROTOCOLUSAGE_*)
int	<u>getPublisherCount</u> () Get the current number of server side publishers
java.util.List	<u>getPublishers</u> () Get the set of server side publishers
java.util.List	<u>getPublishStreamNames</u> () Get the list of live stream names currently being published.
int	<u>getPushPublishSessionCount</u> () Get the number of push publishing sessions.
java.util.List	<u>getPushPublishSessions</u> () Get all active push publish sessions
java.util.List	<u>getPushPublishSessions</u> (IMediaStream stream) Get the current push publish sessions for a given IMediaStream
String	<u>getRepeaterOriginUrl</u> () Get the Repeater Origin URL used by the Live Stream Repeater
String	<u>getRepeaterQueryString</u> () Get the Repeater query string that is used to connect to the origin.
String	<u>getRsoStorageDir</u> () Get remote shared object storage path

String	<code>getRsoStoragePath()</code> Get the resolved storage path to the shared objects
int	<code>getRTPAVSyncMethod()</code> Get RTP audio/video sync method (RTPStream.AVSYNCMETHODS_SENDERREPORT, RTPStream.AVSYNCMETHODS_SYSTEMCLOCK, RTPStream.AVSYNCMETHODS_RTPTIMECODE)
int	<code>getRTPIdleFrequency()</code> Set the default RTP idle frequency (milliseconds)
int	<code>getRTPMaxRTCPWaitTime()</code> Get the maximum time to wait for RTCP packets (milliseconds)
String	<code>getRTPPlayAuthenticationMethod()</code> Get the RTP play authentication method (as defined in conf/Authentication.xml)
<code>WMSProperties</code>	<code>getRTPProperties()</code> Get the property collection of RTP settings that are specific to this application instance
String	<code>getRTPPublishAuthenticationMethod()</code> Get the RTP publish authentication method (as defined in conf/Authentication.xml)
int	<code>getRTPSessionCount()</code> Get the number of RTP sessions running under this application instance
int	<code>getRTPSessionCount(String streamName)</code> Get the number of RTP player streams playing a given stream name
java.util.Map	<code>getRTPSessionCountsByName()</code> Get a map of stream names and session counts of RTP sessions
java.util.List	<code>getRTPSessions()</code> Get a list of RTP sessions running under this application instance
java.util.List	<code>getRTPSessions(String streamName)</code> Get a list of RTP sessions running under this application instance playing a given stream name
String	<code>getRTSPBindIpAddress()</code> Set the IP address to which UDP ports will be bound for RTSP/RTP sessions
String	<code>getRTSPConnectionAddressType()</code> Get the connection IP address type (IP4) to used in the Session Description Protocol data exchanged for an RTSP/RTP session
String	<code>getRTSPConnectionIpAddress()</code> Get the connection IP address to used in the Session Description Protocol data exchanged for an RTSP/RTP session
int	<code>getRTSPMaximumPendingWriteBytes()</code> Get the maximum number of pending write bytes for an RTSP session
String	<code>getRTSPOriginAddressType()</code> Get the origin IP address type (IP4) to used in the Session Description Protocol data exchanged for an RTSP/RTP session
String	<code>getRTSPOriginIpAddress()</code> Get the origin IP address to used in the Session Description Protocol data exchanged for an RTSP/RTP session

int	<u>getRTSPSessionTimeout()</u> Get the RTSP session timeout (milliseconds)
String	<u>getSharedObjectReadAccess()</u> Get the default shared object read access
<u>ISharedObjects</u>	<u>getSharedObjects()</u> Get non-persistent shared object collection
<u>ISharedObjects</u>	<u>getSharedObjects(boolean isPersistent)</u> Get either persistent or non-persistent shared object collection
String	<u>getSharedObjectWriteAccess()</u> Get the default shared object write access
String	<u>getStreamAudioSampleAccess()</u> Get the default stream audio sample access
int	<u>getStreamCount()</u> Get the total number of open streams attached to this application instance
<u>IMediaStreamFileMapper</u>	<u>getStreamFileMapper()</u> Get the stream file mapper.
String	<u>getStreamKeyDir()</u> Get the stream key path
String	<u>getStreamKeyPath()</u> Get the resolved key path to the MediaStreams encryption keys
<u>IMediaStreamNameAliasProvider</u>	<u>getStreamNameAliasProvider()</u> Get the stream name alias provider
<u>WMSProperties</u>	<u>getStreamProperties()</u> Get the property collection of stream settings that are specific to this application instance
String	<u>getStreamReadAccess()</u> Get the default stream read access
<u>WMSProperties</u>	<u>getStreamRecorderProperties()</u> Get the property collection of StreamRecorder settings that are specific to this application instance
<u>MediaStreamMap</u>	<u>getStreams()</u> Get all the mediaStream objects attached to this applicationInstance
String	<u>getStreamStorageDir()</u> Get stream storage path
String	<u>getStreamStoragePath()</u> Get the resolved storage path to the MediaStreams
String	<u>getStreamType()</u> Get default streamType for application.
String	<u>getStreamVideoSampleAccess()</u> Get the default stream video sample access
String	<u>getStreamWriteAccess()</u> Get the default stream write access

WMSProperties	getTimedTextProperties() Get the property collection of timed text settings that are specific to this application instance.
String	getTimeRunning() Get time applicationInstance running
double	getTimeRunningSeconds() Get time running in seconds
LiveStreamTranscoderApplicationContext	getTranscoderApplicationContext() Get live stream transcoder application context
WMSProperties	getTranscoderProperties() Get the property collection of Transcoder settings that are specific to this application instance
int	getValidationFrequency() Get time between validation pings (milliseconds)
IVHost	getVHost() Get parent vHost
String	getVODTimedTextProviderList() Get the comma separated list of VODTimedTextProvider names being used by this application (see conf/TimedTextProviders.xml)
java.util.List	getVODTimedTextProviderSet()
void	incClientCountTotal() Increment the total number of connected client counter by one
boolean	isAcceptConnection() Is auto accept connection on/off
boolean	isDebugAppTimeout() If true appTimeout processing will be logged.
boolean	isLive() Returns true if the application stream type of the application instance is live
boolean	isMediaCacheSourceAllowed(String sourceName) Test to see if a MediaCache source name is allowed
boolean	isRTPIncomingDatagramPortValid(int port) Check a port number to be sure it is a valid RTP UDP port for this application instance
boolean	isValidateFMLEConnections() Returns true if validating FMLE connection (default is false)
void	notifyDvrStreamManagerCreate(IDvrStreamManager dvrStoreManager, String streamName) Notify listeners that Dvr Application Store Manager has been created.
void	notifyDvrStreamManagerDestroy(IDvrStreamManager dvrManager) Notify listeners that Dvr Application Store Manager has been destroyed.
void	notifyDvrStreamManagerInit(IDvrStreamManager dvrStoreManager) Notify listeners that Dvr Application Store Manager has been initialized.

void	<u>notifyHTTPSessionCreate</u> (<u>IHTTPStreamerSession</u> httpSession) Notify HTTP session create
void	<u>notifyHTTPSessionDestroy</u> (<u>IHTTPStreamerSession</u> httpSession) Notify HTTP session destroy
void	<u>notifyLiveStreamDvrRecorderCreate</u> (<u>ILiveStreamDvrRecorder</u> dvr, String streamName) Notify Dvr Recorder Create
void	<u>notifyLiveStreamDvrRecorderDestroy</u> (<u>ILiveStreamDvrRecorder</u> dvr) Notify DVR Recorder has been destroyed.
void	<u>notifyLiveStreamDvrRecorderInit</u> (<u>ILiveStreamDvrRecorder</u> dvr, String streamName) Notify DVR Recorder has been initialized.
void	<u>notifyLiveStreamPacketizerCreate</u> (<u>ILiveStreamPacketizer</u> liveStreamPacketizer, String streamName) Notify Live Stream Packetizer Create
void	<u>notifyLiveStreamPacketizerDestroy</u> (<u>ILiveStreamPacketizer</u> liveStreamPacketizer) Notify Live Stream Packetizer Destory
void	<u>notifyLiveStreamPacketizerInit</u> (<u>ILiveStreamPacketizer</u> liveStreamPacketizer, String streamName) Notify Live Stream Packetizer Init
void	<u>notifyLiveStreamTranscoderCreate</u> (<u>ILiveStreamTranscoder</u> liveStreamTranscoder, <u>IMediaStream</u> stream) Notify live stream transcoder create
void	<u>notifyLiveStreamTranscoderDestroy</u> (<u>ILiveStreamTranscoder</u> liveStreamTranscoder, <u>IMediaStream</u> stream) Notify live stream transcoder destroy
void	<u>notifyLiveStreamTranscoderInit</u> (<u>ILiveStreamTranscoder</u> liveStreamTranscoder, <u>IMediaStream</u> stream) Notify live stream transcoder init
void	<u>notifyMediaReaderClose</u> (<u>IMediaReader</u> mediaReader, <u>IMediaStream</u> stream) Notify media reader notifyMediaReaderClose
void	<u>notifyMediaReaderCreate</u> (<u>IMediaReader</u> mediaReader) Notify media reader notifyMediaReaderCreate
void	<u>notifyMediaReaderExtractMetadata</u> (<u>IMediaReader</u> mediaReader, <u>IMediaStream</u> stream) Notify media reader notifyMediaReaderExtractMetadata
void	<u>notifyMediaReaderInit</u> (<u>IMediaReader</u> mediaReader, <u>IMediaStream</u> stream) Notify media reader notifyMediaReaderInit
void	<u>notifyMediaReaderOpen</u> (<u>IMediaReader</u> mediaReader, <u>IMediaStream</u> stream) Notify media reader notifyMediaReaderOpen
void	<u>notifyMediaWriterOnFLVAddMetadata</u> (<u>IMediaStream</u> stream, java.util.Map extraMetadata) Notify all MediaWriter listeners of onFLVAddMetadata

void	<u>notifyMediaWriterOnWriteComplete</u> (<u>IMediaStream</u> stream, java.io.File file) Notify all MediaWriter listeners of onWriteComplete
void	<u>notifyPushPublishSessionCreate</u> (String streamName, <u>IMediaStream</u> stream, <u>IPushPublishSession</u> pushPublishSession) Notify push publish session create
void	<u>notifyPushPublishSessionDestroy</u> (String streamName, <u>IMediaStream</u> stream, <u>IPushPublishSession</u> pushPublishSession) Notify push publish session destroy
void	<u>notifyRTPSessionCreate</u> (<u>RTPSession</u> rtpSession) Notify RTP session create
void	<u>notifyRTPSessionDestroy</u> (<u>RTPSession</u> rtpSession) Notify RTP session destroy
void	<u>parseAllowDomains</u> (String domainFilterStr) Parse a comma delimited list of domain names used to control access to this application.
String	<u>readAppInstConfig</u> (String sName) Method to read xml config file..
void	<u>registerPlayRTPSession</u> (<u>RTPSession</u> rtpSession) Register an RTP session as a play session
void	<u>removeClientListener</u> (<u>IClientNotify</u> clientListener) Remove client listener.
void	<u>removeDvrRecorderListener</u> (<u>ILiveStreamDvrRecorderActionNotify</u> listener) Remove a Dvr Recorder listener (see: <u>ILiveStreamDvrRecorderActionNotify</u>)
void	<u>removeDvrStreamManagerListener</u> (<u>IDvrStreamManagerActionNotify</u> listener) Remove a Dvr Application Store Manager listener (see: <u>IDvrStoreActionNotify</u>)
void	<u>removeHTTPListener</u> (<u>IHTTPSessionNotify</u> httpListener) Remove listner to be notified of HTTPSession create and destroy.
void	<u>removeHTTPStreamerSession</u> (<u>IHTTPStreamerSession</u> httpStreamerSession) Remove a HTTPStreamerSession from this application instance
void	<u>removeLiveStreamPacketizerListener</u> (<u>ILiveStreamPacketizerActionNotify</u> <u>liveStreamPacketizerListener</u>) Remove a Live Stream Packetizer listener (see: <u>ILiveStreamPacketizerActionNotify</u>)
void	<u>removeLiveStreamTranscoderListener</u> (<u>ILiveStreamTranscoderNotify</u> <u>liveStreamTranscoderListener</u>) Remove a live stream transcoder listener
void	<u>removeMediaCasterListener</u> (<u>IMediaCasterNotify</u> mediaCasterListener) Remove mediaCaster listener.
void	<u>removeMediaReaderListener</u> (<u>IMediaReaderActionNotify</u> mediaReaderListener) Remove media reader listener.
void	<u>removeMediaStreamListener</u> (<u>IMediaStreamNotify</u> mediaStreamListener) Remove mediaStream listener.
void	<u>removeMediaWriterListener</u> (<u>IMediaWriterActionNotify</u> listener) remove MediaWriter listener class.

void	<u>removeModuleListener</u> (<u>IModuleNotify</u> moduleListener) Remove module listener
void	<u>removePlayStreamByName</u> (<u>IMediaStream</u> stream) Remove media stream from the list of streams that are listening for a published stream
void	<u>removePublisher</u> (<u>Publisher</u> publisher) Remove a server side publisher from this application instance
void	<u>removePushPublishSession</u> (<u>IMediaStream</u> stream, <u>IPushPublishSession</u> pushPublishSession) Remove a push publish session
void	<u>removePushPublishSessionListener</u> (<u>IPushPublishSessionNotify</u> pushPublishSessionListener) Remove a push publish session listener (see <u>IPushPublishSessionNotify</u>)
void	<u>removePushPublishSessions</u> (<u>IMediaStream</u> stream) Remove all push publish sessions for a give stream
void	<u>removeRTPListener</u> (<u>IRTPSessionNotify</u> rtpListener) Remove listner to be notified of RTPSession create and destroy.
void	<u>removeRTPSession</u> (<u>RTPSession</u> rtpSession) Remove an RTP session from this application instance
void	<u>removeSharedObjectListener</u> (<u>ISharedObjectNotify</u> sharedObjectListener, boolean isPersistent) Remove sharedObject listener.
boolean	<u>resetMediaCasterStream</u> (String streamName) Reset a media caster stream
boolean	<u>resetMediaCasterStream</u> (String streamName, String streamExt) Reset a media caster stream
void	<u>setAcceptConnection</u> (boolean acceptConnection) Set is auto accept connection
void	<u>setAllowDomains</u> (String[] domainFilter) Set the list of domain names used to control access to this application.
void	<u>setApplicationInstanceTouchTimeout</u> (int applicationInstanceTouchTimeout) Set the application instance touch timeout (milliseconds).
void	<u>setApplicationTimeout</u> (int applicationTimeout) Set application timeout (milliseconds)
void	<u>setClientIdleFrequency</u> (int clientIdleFrequency) Set default client idle frequency (milliseconds)
void	<u>setDebugAppTimeout</u> (boolean debugAppTimeout) If true appTimeout processing will be logged.
void	<u>setDvrRecorderList</u> (String recorderList) Set the comma separated list of Dvr Recorder names being used by this application (see conf/Dvr.xml)
void	<u>setHTTPStreamerList</u> (String httpStreamerList) Set the comma separated list of HTTPStreamer names being used by this application (see conf/HTTPStreamers.xml)

void	<u>setLiveStreamDvrRecorderControl</u> (<u>ILiveStreamDvrRecorderControl</u> controller) Set the Live Stream DVR Controller.
void	<u>setLiveStreamPacketizerControl</u> (<u>ILiveStreamPacketizerControl</u> liveStreamPacketizerControl) Set the Live Stream Packetizer Contoller.
void	<u>setLiveStreamPacketizerList</u> (String liveStreamPacketizerList) Set the comma separated list of LiveStreamPacketizers names being used by this application (see conf/LiveStreamPacketizers.xml)
void	<u>setLiveStreamTranscoderControl</u> (<u>ILiveStreamTranscoderControl</u> liveStreamTranscoderControl) Set the Live Stream Transcoder Contoller.
void	<u>setLiveStreamTranscoderList</u> (String liveStreamTranscoderList) Set comma separated list of transcoders to use for this application instance
void	<u>setMaximumPendingReadBytes</u> (int maximumPendingReaderBytes) Get maximum number of bytes a client connection can have waiting to be written before the connection is terminated.
void	<u>setMaximumPendingWriteBytes</u> (int maximumPendingWriteBytes) Set maximum number a bytes a client connection can have waiting to be sent before the connection is terminated.
void	<u>setMaximumSetBufferTime</u> (int maximumSetBufferTime) Set maximum number of milliseconds allowed for the NetStream.setBufferTime(secs) call.
void	<u>setMaxStorageDirDepth</u> (int maxStorageDirDepth) Maximum folder depth allowed for the StreamStorageDir and SharedObjectStorageDir paths
void	<u>setMediacasterRTPRTSPRTPTransportMode</u> (int mediacasterRTPRTSPRTPTransportMode) RTP MediaCaster RTSP/RTP transport mode.
void	<u>setMediaCasterValidator</u> (<u>IMediaCasterValidateMediaCaster</u> mediaCasterValidator) Set the MediaCaster validator interface for this application instance
void	<u>setMediaListProvider</u> (<u>IMediaListProvider</u> mediaListProvider) Set the current media list provider.
void	<u>setName</u> (String name) Set applicationInstance name
void	<u>setPingTimeout</u> (int pingTimeout) Set ping timeout (milliseconds)
void	<u>setRepeaterOriginUrl</u> (String repeaterOriginUrl) Set the Repeater Origin URL used by the Live Stream Repeater
void	<u>setRepeaterQueryString</u> (String repeaterQueryString) Set the Repeater query string that is used to connect to the origin.
void	<u>setRsoStorageDir</u> (String rsoStorageDir) Set remote shared object storage path

void	<u>setRTPAVSyncMethod</u> (int rtpAVSyncMethod) Set RTP audio/video sync method (RTPStream.AVSYNCMETHODS_SENDERREPORT, RTPStream.AVSYNCMETHODS_SYSTEMCLOCK, RTPStream.AVSYNCMETHODS_RTPTIMECODE)
void	<u>setRTPIIdleFrequency</u> (int rtspIdleFrequency) Get the default RTP idle frequency (milliseconds)
void	<u>setRTSPMaxRTCPWaitTime</u> (int rtpMaxRTCPWaitTime) Set the maximum time to wait for RTCP packets (milliseconds)
void	<u>setRTSPPlayAuthenticationMethod</u> (String rtpPlayAuthenticationMethod) Set the RTP play authentication method (as defined in conf/Authentication.xml)
void	<u>setRTSPPublishAuthenticationMethod</u> (String rtpPublishAuthenticationMethod) Set the RTP publish authentication method (as defined in conf/Authentication.xml)
void	<u>setRTSPBindIpAddress</u> (String rtspBindIpAddress) Get the IP address to which UDP ports will be bound for RTSP/RTP sessions
void	<u>setRTSPConnectionAddressType</u> (String rtspConnectionAddressType) Set the connection IP address type (IP4) to used in the Session Description Protocol data exchanged for an RTSP/RTP session
void	<u>setRTSPConnectionIpAddress</u> (String rtspConnectionIpAddress) Set the connection IP address to used in the Session Description Protocol data exchanged for an RTSP/RTP session
void	<u>setRTSPMaximumPendingWriteBytes</u> (int rtspMaximumPendingWriteBytes) Set the maximum number of pending write bytes for an RTSP session
void	<u>setRTSPOriginAddressType</u> (String rtspOriginAddressType) Set the origin IP address type (IP4) to used in the Session Description Protocol data exchanged for an RTSP/RTP session
void	<u>setRTSPOriginIpAddress</u> (String rtspOriginIpAddress) Set the origin IP address to used in the Session Description Protocol data exchanged for an RTSP/RTP session
void	<u>setRTSPSessionTimeout</u> (int rtspSessionTimeout) Set the RTSP session timeout (milliseconds)
void	<u>setSharedObjectReadAccess</u> (String sharedObjectReadAccess) Set the default shared object read access
void	<u>setSharedObjectWriteAccess</u> (String sharedObjectWriteAccess) Set the default shared object write access
void	<u>setStreamAudioSampleAccess</u> (String streamAudioSampleAccess) Set the default stream audio sample access
void	<u>setStreamFileMapper</u> (<u>IMediaStreamFileMapper</u> streamFileMapper) Set the stream file mapper.
void	<u>setStreamKeyDir</u> (String keyStorageDir) Set the stream key path
void	<u>setStreamNameAliasProvider</u> (<u>IMediaStreamNameAliasProvider</u> streamNameAliasProvider) Set the stream name alias provider

void	<u>setStreamReadAccess</u> (String streamReadAccess) Set the default stream read access
void	<u>setStreamStorageDir</u> (String streamStorageDir) Set stream storage path
void	<u>setStreamType</u> (String streamType) Set default stream type for application.
void	<u>setStreamVideoSampleAccess</u> (String streamVideoSampleAccess) Set the default stream video sample access
void	<u>setStreamWriteAccess</u> (String streamWriteAccess) Set the default stream write access
void	<u>setValidateFMLEConnections</u> (boolean validateFMLEConnections) Returns true if validating FMLE connection (default is false)
void	<u>setValidationFrequency</u> (int validationFrequency) Set time between validation pings (milliseconds)
void	<u>setVODTimedTextProviderList</u> (String timedTextProviderList) Set the comma separated list of VODTimedTextProvider names being used by this application (see conf/TimedTextProviders.xml)
void	<u>shutdown</u> (boolean isServerShutdown, boolean isAppShutdown) shutdown applicationInstance
void	<u>shutdownClient</u> (<u>IClient</u> client) shutdown a client connection immediately
boolean	<u>startMediaCasterStream</u> (String streamName, String mediaCasterType) Start a media caster stream
boolean	<u>startMediaCasterStream</u> (String streamName, String streamExt, String mediaCasterType) Start a media caster stream
void	<u>stopMediaCasterStream</u> (String streamName) Stop a media caster stream
void	<u>touch</u> () Touch the application instance so that it stays loaded for at least applicationInstanceTouchTimeout
boolean	<u>writeAppInstConfig</u> (String sName, String data) Method to write xml config file..

Fields

PROTOCOLUSAGE_RTMP

```
public static final int PROTOCOLUSAGE_RTMP
```

Constant value: 0

(continued from last page)

PROTOCOLUSAGE_RTMPT

```
public static final int PROTOCOLUSAGE_RTMPT
```

Constant value: **1**

PROTOCOLUSAGE_RTMP

```
public static final int PROTOCOLUSAGE_RTMP
```

Constant value: **2**

PROTOCOLUSAGE_RTMPTS

```
public static final int PROTOCOLUSAGE_RTMPTS
```

Constant value: **3**

PROTOCOLUSAGE_RTMPE

```
public static final int PROTOCOLUSAGE_RTMPE
```

Constant value: **4**

PROTOCOLUSAGE_RTMPTE

```
public static final int PROTOCOLUSAGE_RTMPTE
```

Constant value: **5**

PROTOCOLUSAGE_RTP

```
public static final int PROTOCOLUSAGE_RTP
```

Constant value: **6**

PROTOCOLUSAGE_CUPERTINO

```
public static final int PROTOCOLUSAGE_CUPERTINO
```

Constant value: **7**

PROTOCOLUSAGE_SMOOTH

```
public static final int PROTOCOLUSAGE_SMOOTH
```

Constant value: **8**

PROTOCOLUSAGE_SANJOSE

```
public static final int PROTOCOLUSAGE_SANJOSE
```

(continued from last page)

Constant value: **9**

PROTOCOLUSAGE_MPEGDASH

```
public static final int PROTOCOLUSAGE_MPEGDASH
```

Constant value: **10**

PROTOCOLUSAGE_WEBM

```
public static final int PROTOCOLUSAGE_WEBM
```

Constant value: **11**

PROTOCOLUSAGE_WOWZ

```
public static final int PROTOCOLUSAGE_WOWZ
```

Constant value: **12**

PROTOCOLUSAGE_WOWZE

```
public static final int PROTOCOLUSAGE_WOWZE
```

Constant value: **13**

PROTOCOLUSAGE_WOWZS

```
public static final int PROTOCOLUSAGE_WOWZS
```

Constant value: **14**

PROTOCOLUSAGE_ORIGINCUPERTINO

```
public static final int PROTOCOLUSAGE_ORIGINCUPERTINO
```

Constant value: **15**

PROTOCOLUSAGE_ORIGINSMOOTH

```
public static final int PROTOCOLUSAGE_ORIGINSMOOTH
```

Constant value: **16**

PROTOCOLUSAGE_ORIGINSANJOSE

```
public static final int PROTOCOLUSAGE_ORIGINSANJOSE
```

Constant value: **17**

PROTOCOLUSAGE_ORIGINMPEGDASH

```
public static final int PROTOCOLUSAGE_ORIGINMPEGDASH
```

Constant value: **18**

PROTOCOLUSAGE_PUSHUBLISHING

```
public static final int PROTOCOLUSAGE_PUSHUBLISHING
```

Constant value: **19**

PROTOCOLUSAGE_MEDIACACHE

```
public static final int PROTOCOLUSAGE_MEDIACACHE
```

Constant value: **20**

PROTOCOLUSAGE_RTMP_IN

```
public static final int PROTOCOLUSAGE_RTMP_IN
```

Constant value: **21**

PROTOCOLUSAGE_RTMP_OUT

```
public static final int PROTOCOLUSAGE_RTMP_OUT
```

Constant value: **22**

PROTOCOLUSAGE_RTMPT_IN

```
public static final int PROTOCOLUSAGE_RTMPT_IN
```

Constant value: **23**

PROTOCOLUSAGE_RTMPT_OUT

```
public static final int PROTOCOLUSAGE_RTMPT_OUT
```

Constant value: **24**

PROTOCOLUSAGE_RTMPS_IN

```
public static final int PROTOCOLUSAGE_RTMPS_IN
```

Constant value: **25**

(continued from last page)

PROTOCOLUSAGE_RTMPS_OUT

```
public static final int PROTOCOLUSAGE_RTMPS_OUT
```

Constant value: **26**

PROTOCOLUSAGE_RTMPTS_IN

```
public static final int PROTOCOLUSAGE_RTMPTS_IN
```

Constant value: **27**

PROTOCOLUSAGE_RTMPTS_OUT

```
public static final int PROTOCOLUSAGE_RTMPTS_OUT
```

Constant value: **28**

PROTOCOLUSAGE_RTMPE_IN

```
public static final int PROTOCOLUSAGE_RTMPE_IN
```

Constant value: **29**

PROTOCOLUSAGE_RTMPE_OUT

```
public static final int PROTOCOLUSAGE_RTMPE_OUT
```

Constant value: **30**

PROTOCOLUSAGE_RTMPTE_IN

```
public static final int PROTOCOLUSAGE_RTMPTE_IN
```

Constant value: **31**

PROTOCOLUSAGE_RTMPTE_OUT

```
public static final int PROTOCOLUSAGE_RTMPTE_OUT
```

Constant value: **32**

PROTOCOLUSAGE_RTP_IN

```
public static final int PROTOCOLUSAGE_RTP_IN
```

Constant value: **33**

PROTOCOLUSAGE_RTP_OUT

```
public static final int PROTOCOLUSAGE_RTP_OUT
```

(continued from last page)

Constant value: **34**

PROTOCOLUSAGE_RTSP_IN

```
public static final int PROTOCOLUSAGE_RTSP_IN
```

Constant value: **35**

PROTOCOLUSAGE_RTSP_OUT

```
public static final int PROTOCOLUSAGE_RTSP_OUT
```

Constant value: **36**

PROTOCOLUSAGE_WOWZ_IN

```
public static final int PROTOCOLUSAGE_WOWZ_IN
```

Constant value: **37**

PROTOCOLUSAGE_WOWZ_OUT

```
public static final int PROTOCOLUSAGE_WOWZ_OUT
```

Constant value: **38**

PROTOCOLUSAGE_WOWZE_IN

```
public static final int PROTOCOLUSAGE_WOWZE_IN
```

Constant value: **39**

PROTOCOLUSAGE_WOWZE_OUT

```
public static final int PROTOCOLUSAGE_WOWZE_OUT
```

Constant value: **40**

PROTOCOLUSAGE_WOWZS_IN

```
public static final int PROTOCOLUSAGE_WOWZS_IN
```

Constant value: **41**

PROTOCOLUSAGE_WOWZS_OUT

```
public static final int PROTOCOLUSAGE_WOWZS_OUT
```

Constant value: **42**

PROTOCOLUSAGE_MC_SHOUTCAST

```
public static final int PROTOCOLUSAGE_MC_SHOUTCAST
```

Constant value: **43**

PROTOCOLUSAGE_MC_LIVEREPEATER

```
public static final int PROTOCOLUSAGE_MC_LIVEREPEATER
```

Constant value: **44**

PROTOCOLUSAGE_MC_RTP

```
public static final int PROTOCOLUSAGE_MC_RTP
```

Constant value: **45**

PROTOCOLUSAGE_RTSP_PULL_IN

```
public static final int PROTOCOLUSAGE_RTSP_PULL_IN
```

Constant value: **46**

PROTOCOLUSAGE_RTSP_PULL_OUT

```
public static final int PROTOCOLUSAGE_RTSP_PULL_OUT
```

Constant value: **47**

PROTOCOLUSAGE_TOTAL

```
public static final int PROTOCOLUSAGE_TOTAL
```

Constant value: **48**

DEFAULT_APPINSTANCE_NAME

```
public static final java.lang.String DEFAULT_APPINSTANCE_NAME
```

Constant value: **_definst_**

Methods

shutdownClient

```
public void shutdownClient(IClient client)
```

shutdown a client connection immediately

Parameters:

(continued from last page)

client - client connection

shutdown

```
public void shutdown(boolean isServerShutdown,  
                     boolean isAppShutdown)
```

shutdown applicationInstance

Parameters:

isServerShutdown - true if due to shutdown of server

isAppShutdown - true if due to shutdown of application

getApplication

```
public IApplication getApplication()
```

Get parent application

Returns:

parent application

getName

```
public String getName()
```

Get applicationInstance name

Returns:

applicationInstance name

setName

```
public void setName(String name)
```

Set applicationInstance name

Parameters:

name - applicationInstance name

getStreams

```
public MediaStreamMap getStreams()
```

(continued from last page)

Get all the mediaStream objects attached to this applicationInstance

Get Stream By StreamId

```
IClient client;  
int streamId;  
  
MediaStreamMap streams = client.getAppInstance().getStreams();  
IMediaStream stream = streams.getStream((IClient)null, streamId);
```

Returns:

collection of mediaStream objects

getVHost

```
public IVHost getVHost()
```

Get parent vHost

Returns:

parent vHost

getProperties

```
public WMSProperties getProperties()
```

Get applicationInstance properties

Returns:

applicationInstance properties

getManagerProperties

```
public WMSProperties getManagerProperties()
```

Get Application's manager's properties collection

Returns:

manager's property collection

getStreamType

```
public String getStreamType()
```

Get default streamType for application.

Returns:

streamType name

setStreamType

```
public void setStreamType(String streamType)
```

Set default stream type for application.

Parameters:

streamType - streamType name

isAcceptConnection

```
public boolean isAcceptConnection()
```

Is auto accept connection on/off

Returns:

auto accept connection

setAcceptConnection

```
public void setAcceptConnection(boolean acceptConnection)
```

Set is auto accept connection

Parameters:

acceptConnection - auto accept connection

getClientCountTotal

```
public int getClientCountTotal()
```

Get number of client connections in total that have connected to this applicationInstance

Returns:

number of client connections

incClientCountTotal

```
public void incClientCountTotal()
```

Increment the total number of connected client counter by one

getClientCount

```
public int getClientCount()
```

Get number of client connections currently connected to applicationInstance

Returns:

number of client connections

getClientById

```
public IClient getClientById(int index)
```

Get a client connection by the client Id

(continued from last page)

Parameters:

index - client Id

Returns:client connection

getClients

```
public java.util.List getClients()
```

Get the set of clients currently connected to this application instance (replaces getClient(index))

Returns:set of clients

getClient

```
public IClient getClient(int index)
```

Deprecated. *Get the client connection at index. This method is deprecated. It is best to use getClient() to return a List objects.*

Parameters:

index - index

Returns:client connection

getSharedObjects

```
public ISharedObjects getSharedObjects()
```

Get non-persistent shared object collection

Returns:collection of non-persistent shared objects

getSharedObjects

```
public ISharedObjects getSharedObjects(boolean isPersistent)
```

Get either persistent or non-persistent shared object collection

Parameters:

isPersistent

Returns:collection of shared objects

addClientListener

```
public void addClientListener(IClientNotify clientListener)
```

(continued from last page)

Add client listener. Listens for connects, disconnect, accepts and reject

Add a Client Listener

```
IApplicationInstance appInstance;

class ClientListener implements IClientNotify
{
    public void onClientConnect(IClient client)
    {
        WMSLoggerFactory.getLogger(null).debug("onClientConnect: "+
            client.getClientId());
    }

    public void onClientDisconnect(IClient client)
    {
        WMSLoggerFactory.getLogger(null).debug("onClientDisconnect: "+
            client.getClientId());
    }

    public void onClientAccept(IClient client)
    {
        WMSLoggerFactory.getLogger(null).debug("onClientAccept: "+
            client.getClientId());
    }

    public void onClientReject(IClient client)
    {
        WMSLoggerFactory.getLogger(null).debug("onClientReject: "+
            client.getClientId());
    }
}

appInstance.addClientListener(new ClientListener());
```

Parameters:

clientListener - client listener

removeClientListener

public void **removeClientListener**([IClientNotify](#) clientListener)

Remove client listener. Listens for connects, disconnect, accepts and reject

Parameters:

clientListener - client listener

addMediaStreamListener

public void **addMediaStreamListener**([IMediaStreamNotify](#) mediaStreamListener)

Add mediaStream listener. Listens for create and destroy

Add a MediaStream Listener

```
IApplicationInstance appInstance;

class MediaStreamListener implements IMediaStreamNotify
{
    public void onMediaStreamCreate(IMediaStream stream)
    {
        WMSLoggerFactory.getLogger(null).debug("onMediaStreamCreate: "+
            stream.getSrc());
    }

    public void onMediaStreamDestroy(IMediaStream stream)
    {
        WMSLoggerFactory.getLogger(null).debug("onMediaStreamDestroy: "+
            stream.getSrc());
    }
}

appInstance.addMediaStreamListener(new MediaStreamListener());
```

Parameters:

mediaStreamListener - mediaStream listener

removeMediaStreamListener

public void **removeMediaStreamListener**([IMediaStreamNotify](#) mediaStreamListener)

Remove mediaStream listener. Listens for create and destroy

Parameters:

mediaStreamListener - mediaStream listener

addSharedObjectListener

public void **addSharedObjectListener**([ISharedObjectNotify](#) sharedObjectListener,
boolean isPersistent)

(continued from last page)

Add sharedObject listener. Listens for create, destroy, clientConnect, clientDisconnect

Add SharedObject Listener

```
IApplicationInstance appInstance;

class SharedObjectListener implements ISharedObjectNotify
{
    public void onSharedObjectCreate(ISharedObject sharedObject)
    {
        WMSLoggerFactory.getLogger(null).debug("onSharedObjectCreate: "+
            sharedObject.getName());
    }

    public void onSharedObjectDestroy(ISharedObject sharedObject)
    {
        WMSLoggerFactory.getLogger(null).debug("onSharedObjectDestroy: "+
            sharedObject.getName());
    }

    public void onSharedObjectConnect(ISharedObject sharedObject, IClient client)
    {
        WMSLoggerFactory.getLogger(null).debug("onSharedObjectConnect: "+
            sharedObject.getName());
    }

    public void onSharedObjectDisconnect(ISharedObject sharedObject, IClient client)
    {
        WMSLoggerFactory.getLogger(null).debug("onSharedObjectDisconnect: "+
            sharedObject.getName());
    }
}

appInstance.addSharedObjectListener(new SharedObjectListener(), false);
```

Parameters:

sharedObjectListener - sharedObject listener
isPersistent

removeSharedObjectListener

```
public void removeSharedObjectListener(ISharedObjectNotify sharedObjectListener,
    boolean isPersistent)
```

Remove sharedObject listener. Listens for create, destroy, clientConnect, clientDisconnect

Parameters:

(continued from last page)

sharedObjectListener - sharedObject listener
isPersistent

addMediaCasterListener

```
public void addMediaCasterListener(IMediaCasterNotify mediaCasterListener)
```

Add mediaCaster listener. Listens for create, destroy, registerPlayer, unregisterPlayer, setSourceStream

Parameters:

mediaCasterListener

addMediaCasterListener

```
public void addMediaCasterListener(IMediaCasterNotify2 mediaCasterListener)
```

Add mediaCaster listener. Listens for create, destroy, registerPlayer, unregisterPlayer, setSourceStream

Parameters:

mediaCasterListener

removeMediaCasterListener

```
public void removeMediaCasterListener(IMediaCasterNotify mediaCasterListener)
```

Remove mediaCaster listener. Listens for create, destroy, registerPlayer, unregisterPlayer, setSourceStream

Parameters:

mediaCasterListener

getConnectionCounter

```
public ConnectionCounter getConnectionCounter( )
```

Get the connectionCounter for applicationInstance

Returns:

connection counter

getConnectionCounter

```
public ConnectionCounterSimple getConnectionCounter(int counterIndex)
```

Get the connectionCounter for applicationInstance for a specific technology (see IVHost.COUNTER_*)

Parameters:

counterIndex - counter index

Returns:

connection counter

getDateStarted

```
public String getDateStarted( )
```

Get date applicationInstance started

Returns:

(continued from last page)

date applicationInstance started

getTimeRunning

```
public String getTimeRunning()
```

Get time applicationInstance running

Returns:

time applicationInstance running

getTimeRunningSeconds

```
public double getTimeRunningSeconds()
```

Get time running in seconds

Returns:

time running in seconds

broadcastMsg

```
public void broadcastMsg(java.util.List clientList,  
                          String handlerName)
```

Broadcast a message to a specific list of clients connected to this application instance

Parameters:

clientList - list of client

handlerName - handler name

broadcastMsg

```
public void broadcastMsg(java.util.List clientList,  
                          String handlerName,  
                          Object[] params)
```

Broadcast a message to a specific list of clients connected to this application instance

Parameters:

clientList - list of client

handlerName - handler name

params - parameters

broadcastMsg

```
public void broadcastMsg(String handlerName,  
                          Object[] params)
```

(continued from last page)

Broadcast a message to all clients connected to this applicationInstance

Broadcast Message to All Clients

```
IApplicationInstance appInstance;
appInstance.broadcastMsg("onNotify", "Hello World", 1.2345, false, new Date());
```

Parameters:

handlerName - handler name

params - variable list of arguments (Java primitive and Strings will be wrapped in AMFData objects)

getIOPerformanceCounter

```
public IOPerformanceCounter getIOPerformanceCounter()
```

Get the performance counter for applicationInstance

Returns:

io performance counter

getIOPerformanceCounter

```
public IOPerformanceCounter getIOPerformanceCounter(int counterIndex)
```

Get the performance counter for applicationInstance for a specific technology (see IVHost.COUNTER_*)

Parameters:

counterIndex - counter index (see IVHost.COUNTER_*)

Returns:

connection counter

addPlayStreamByName

```
public void addPlayStreamByName(IMediaStream stream,
    String name)
```

Add a media stream to the list of streams that are listening for a published stream

Parameters:

stream - media stream

name - stream name

removePlayStreamByName

```
public void removePlayStreamByName(IMediaStream stream)
```

Remove media stream from the list of streams that are listening for a published stream

Parameters:

(continued from last page)

stream - media stream

getPlayStreamCountsByName

```
public java.util.Map getPlayStreamCountsByName( )
```

Get a map of stream names to number of Flash players playing the stream name

Returns:

map of stream names to number of Flash players playing the stream name

getPlayStreamCount

```
public int getPlayStreamCount(String streamName)
```

Get the number of Flash players playing a given stream name

Parameters:

streamName - stream name

Returns:

number of players

getPlayStreamsByName

```
public java.util.List getPlayStreamsByName(String name)
```

Get a list of media streams that are listening for published stream.

Parameters:

name - stream name

Returns:

list of streams or null if no listeners

getMediaCasterStreams

```
public MediaCasterStreamMap getMediaCasterStreams( )
```

Get the media caster streams attached to this application instance

Returns:

media caster streams attached to this application instance

getStreamCount

```
public int getStreamCount( )
```

Get the total number of open streams attached to this application instance

Returns:

the total number of open streams attached to this application instance

getModFunctions

```
public ModuleFunctions getModFunctions( )
```

Get list of application modules

(continued from last page)

Returns:

list of application modules

addModuleListener

```
public void addModuleListener(IModuleNotify moduleListener)
```

Add module listener. Listens for onModuleLoad and onModuleUnload events. See IModuleNotify.

Parameters:

moduleListener - module listener

removeModuleListener

```
public void removeModuleListener(IModuleNotify moduleListener)
```

Remove module listener

Parameters:

moduleListener - module listener

getModuleList

```
public ModuleList getModuleList()
```

Get the list of loaded modules.

Returns:

list of loaded modules

getModuleInstance

```
public Object getModuleInstance(String name)
```

Get the instance of the module class for this application instance.

Parameters:

name - module name as defined in Application.xml

Returns:

instance of class for this application instance

getApplicationTimeout

```
public int getApplicationTimeout()
```

Get application timeout (milliseconds)

Returns:

application timeout (milliseconds)

setApplicationTimeout

```
public void setApplicationTimeout(int applicationTimeout)
```

Set application timeout (milliseconds)

(continued from last page)

Parameters:

applicationTimeout - application timeout (milliseconds)

getPingTimeout

```
public int getPingTimeout()
```

Get ping timeout (milliseconds)

Returns:

ping timeout (milliseconds)

setPingTimeout

```
public void setPingTimeout(int pingTimeout)
```

Set ping timeout (milliseconds)

Parameters:

pingTimeout - ping timeout (milliseconds)

getValidationFrequency

```
public int getValidationFrequency()
```

Get time between validation pings (milliseconds)

Returns:

time between validation pings (milliseconds)

setValidationFrequency

```
public void setValidationFrequency(int validationFrequency)
```

Set time between validation pings (milliseconds)

Parameters:

validationFrequency - time between validation pings (milliseconds)

getMaximumPendingWriteBytes

```
public int getMaximumPendingWriteBytes()
```

Get maximum number a bytes a client connection can have waiting to be sent before the connection is terminated. If set to zero this feature is turned off.

Returns:

maximum number a bytes a client connection can have waiting to be sent before the connection is terminated

setMaximumPendingWriteBytes

```
public void setMaximumPendingWriteBytes(int maximumPendingWriteBytes)
```

Set maximum number a bytes a client connection can have waiting to be sent before the connection is terminated. If set to zero this feature is turned off.

Parameters:

maximumPendingWriteBytes - maximum number a bytes a client connection can have waiting to be sent before the connection is terminated

getMaximumPendingReadBytes

```
public int getMaximumPendingReadBytes()
```

Set maximum number of bytes a client connection can have waiting to be written before the connection is terminated. If set to zero this feature is off.

Returns:

maximum number of bytes a client connection can have waiting to be written before the connection is terminated

setMaximumPendingReadBytes

```
public void setMaximumPendingReadBytes(int maximumPendingReaderBytes)
```

Get maximum number of bytes a client connection can have waiting to be written before the connection is terminated. If set to zero this feature is off.

Parameters:

maximumPendingReaderBytes - maximum number of bytes a client connection can have waiting to be written before the connection is terminated

setMaximumSetBufferTime

```
public void setMaximumSetBufferTime(int maximumSetBufferTime)
```

Set maximum number of milliseconds allowed for the NetStream.setBufferTime(secs) call. If set to zero this feature is turned off.

Parameters:

maximumSetBufferTime - maximum number of milliseconds allowed for the NetStream.setBufferTime(secs) call

getMaximumSetBufferTime

```
public int getMaximumSetBufferTime()
```

Get maximum number of milliseconds allowed for the NetStream.setBufferTime(secs) call. If set to zero this feature is turned off.

Returns:

maximum number of milliseconds allowed for the NetStream.setBufferTime(secs) call

getRepeaterOriginUrl

```
public String getRepeaterOriginUrl()
```

Get the Repeater Origin URL used by the Live Stream Repeater

Returns:

URL used by the Live Stream Repeater

setRepeaterOriginUrl

```
public void setRepeaterOriginUrl(String repeaterOriginUrl)
```

Set the Repeater Origin URL used by the Live Stream Repeater

Parameters:

repeaterOriginUrl - URL used by the Live Stream Repeater

getRepeaterQueryString

```
public String getRepeaterQueryString()
```

Get the Repeater query string that is used to connect to the origin. This value can be used to pass secure URL parameters to the origin for security validation.

Returns:

Repeater query string

setRepeaterQueryString

```
public void setRepeaterQueryString(String repeaterQueryString)
```

Set the Repeater query string that is used to connect to the origin. This value can be used to pass secure URL parameters to the origin for security validation.

Parameters:

repeaterQueryString - Repeater query string

getAllowDomains

```
public String[] getAllowDomains()
```

Get the list of domain names used to control access to this application. Upon connection, if this list is non-null the client.referrer value is checked to make sure the referrer is from a domain in this list.

Returns:

list of domain names used to control access to this application

setAllowDomains

```
public void setAllowDomains(String[] domainFilter)
```

Set the list of domain names used to control access to this application. Upon connection, if this list is non-null the client.referrer value is checked to make sure the referrer is from a domain in this list.

Parameters:

domainFilter - list of domain names used to control access to this application

parseAllowDomains

```
public void parseAllowDomains(String domainFilterStr)
```

Parse a comma delimited list of domain names used to control access to this application. Upon connection, if this list is non-null the client.referrer value is checked to make sure the referrer is from a domain in this list.

Parameters:

domainFilterStr - comma delimited list of domain names

getClientIdleFrequency

```
public int getClientIdleFrequency()
```

Get default client idle frequency (milliseconds)

Returns:

default client idle frequency (milliseconds)

setClientIdleFrequency

```
public void setClientIdleFrequency(int clientIdleFrequency)
```

Set default client idle frequency (milliseconds)

Parameters:

clientIdleFrequency - default client idle frequency (milliseconds)

getRTPIdeFrequency

```
public int getRTPIdeFrequency( )
```

Set the default RTP idle frequency (milliseconds)

Returns:

default RTP idle frequency (milliseconds)

setRTPIdeFrequency

```
public void setRTPIdeFrequency(int rtspIdleFrequency)
```

Get the default RTP idle frequency (milliseconds)

Parameters:

rtspIdleFrequency - default RTP idle frequency (milliseconds)

getStreamStorageDir

```
public String getStreamStorageDir( )
```

Get stream storage path

Returns:

stream storage path

setStreamStorageDir

```
public void setStreamStorageDir(String streamStorageDir)
```

Set stream storage path

Parameters:

streamStorageDir - stream storage path

getStreamKeyDir

```
public String getStreamKeyDir( )
```

Get the stream key path

Returns:

stream key path

setStreamKeyDir

```
public void setStreamKeyDir(String keyStorageDir)
```

(continued from last page)

Set the stream key path

Parameters:

keyStorageDir - stream key path

getRsoStorageDir

```
public String getRsoStorageDir()
```

Get remote shared object storage path

Returns:

remote shared object storage path

setRsoStorageDir

```
public void setRsoStorageDir(String rsoStorageDir)
```

Set remote shared object storage path

Parameters:

rsoStorageDir - remote shared object storage path

getStreamKeyPath

```
public String getStreamKeyPath()
```

Get the resolved key path to the MediaStreams encryption keys

Returns:

resolved key path to the MediaStreams encryption keys

getStreamStoragePath

```
public String getStreamStoragePath()
```

Get the resolved storage path to the MediaStreams

Returns:

resolved storage path to the MediaStreams

getRsoStoragePath

```
public String getRsoStoragePath()
```

Get the resolved storage path to the shared objects

Returns:

resolved storage path to the shared objects

getStreamVideoSampleAccess

```
public String getStreamVideoSampleAccess()
```

Get the default stream video sample access

Returns:

default stream video sample access

(continued from last page)

See Also:

[IClient.getStreamVideoSampleAccess\(\)](#)

setStreamVideoSampleAccess

```
public void setStreamVideoSampleAccess(String streamVideoSampleAccess)
```

Set the default stream video sample access

Parameters:

streamVideoSampleAccess - default stream video sample access

See Also:

[IClient.setStreamVideoSampleAccess\(String\)](#)

getStreamAudioSampleAccess

```
public String getStreamAudioSampleAccess()
```

Get the default stream audio sample access

Returns:

default stream audio sample access

See Also:

[IClient.getStreamAudioSampleAccess\(\)](#)

setStreamAudioSampleAccess

```
public void setStreamAudioSampleAccess(String streamAudioSampleAccess)
```

Set the default stream audio sample access

Parameters:

streamAudioSampleAccess

See Also:

[IClient.setStreamAudioSampleAccess\(String\)](#)

getStreamReadAccess

```
public String getStreamReadAccess()
```

Get the default stream read access

Returns:

default stream read access

See Also:

[IClient.getStreamReadAccess\(\)](#)

setStreamReadAccess

```
public void setStreamReadAccess(String streamReadAccess)
```

Set the default stream read access

(continued from last page)

Parameters:

streamReadAccess - default stream read access

See Also:[IClient.setStreamReadAccess\(String\)](#)

getStreamWriteAccess

```
public String getStreamWriteAccess()
```

Get the default stream write access

Returns:

default stream write access

See Also:[IClient.getStreamWriteAccess\(\)](#)

setStreamWriteAccess

```
public void setStreamWriteAccess(String streamWriteAccess)
```

Set the default stream write access

Parameters:

streamWriteAccess - default stream write access

See Also:[IClient.setStreamWriteAccess\(String\)](#)

getSharedObjectReadAccess

```
public String getSharedObjectReadAccess()
```

Get the default shared object read access

Returns:

default shared object read access

See Also:[IClient.getSharedObjectReadAccess\(\)](#)

setSharedObjectReadAccess

```
public void setSharedObjectReadAccess(String sharedObjectReadAccess)
```

Set the default shared object read access

Parameters:

sharedObjectReadAccess - default shared object read access

See Also:[IClient.setSharedObjectReadAccess\(String\)](#)

getSharedObjectWriteAccess

```
public String getSharedObjectWriteAccess()
```

(continued from last page)

Get the default shared object write access

Returns:

default shared object write access

See Also:

[IClient.getSharedObjectWriteAccess\(\)](#)

setSharedObjectWriteAccess

```
public void setSharedObjectWriteAccess(String sharedObjectWriteAccess)
```

Set the default shared object write access

Parameters:

sharedObjectWriteAccess - default shared object write access

See Also:

[IClient.setSharedObjectWriteAccess\(String\)](#)

getRTPPublishAuthenticationMethod

```
public String getRTPPublishAuthenticationMethod()
```

Get the RTP publish authentication method (as defined in conf/Authentication.xml)

Returns:

RTP publish authentication method

setRTPPublishAuthenticationMethod

```
public void setRTPPublishAuthenticationMethod(String rtpPublishAuthenticationMethod)
```

Set the RTP publish authentication method (as defined in conf/Authentication.xml)

Parameters:

rtpPublishAuthenticationMethod - RTP publish authentication method

getRTPPlayAuthenticationMethod

```
public String getRTPPlayAuthenticationMethod()
```

Get the RTP play authentication method (as defined in conf/Authentication.xml)

Returns:

RTP play authentication method

setRTPPlayAuthenticationMethod

```
public void setRTPPlayAuthenticationMethod(String rtpPlayAuthenticationMethod)
```

Set the RTP play authentication method (as defined in conf/Authentication.xml)

Parameters:

rtpPlayAuthenticationMethod - RTP play authentication method

(continued from last page)

getRTPAVSyncMethod

```
public int getRTPAVSyncMethod( )
```

Get RTP audio/video sync method (RTPStream.AVSYNCMETHODS_SENDERREPORT, RTPStream.AVSYNCMETHODS_SYSTEMCLOCK, RTPStream.AVSYNCMETHODS_RTPTIMECODE)

Returns:

RTP audio/video sync method

setRTPAVSyncMethod

```
public void setRTPAVSyncMethod(int rtpAVSyncMethod)
```

Set RTP audio/video sync method (RTPStream.AVSYNCMETHODS_SENDERREPORT, RTPStream.AVSYNCMETHODS_SYSTEMCLOCK, RTPStream.AVSYNCMETHODS_RTPTIMECODE)

Parameters:

rtpAVSyncMethod - RTP audio/video sync method

getRTPMaxRTCPWaitTime

```
public int getRTPMaxRTCPWaitTime( )
```

Get the maximum time to wait for RTCP packets (milliseconds)

Returns:

maximum time to wait for RTCP packets (milliseconds)

setRTPMaxRTCPWaitTime

```
public void setRTPMaxRTCPWaitTime(int rtpMaxRTCPWaitTime)
```

Set the maximum time to wait for RTCP packets (milliseconds)

Parameters:

rtpMaxRTCPWaitTime - maximum time to wait for RTCP packets (milliseconds)

getRTPSessions

```
public java.util.List getRTPSessions(String streamName)
```

Get a list of RTP sessions running under this application instance playing a given stream name

Parameters:

streamName - stream name

Returns:

list of RTP sessions running under this application instance playing a given stream name

getRTPSessions

```
public java.util.List getRTPSessions( )
```

Get a list of RTP sessions running under this application instance

Returns:

list of RTP sessions running under this application instance

getRTPSessionCountsByName

```
public java.util.Map getRTPSessionCountsByName()
```

Get a map of stream names and session counts of RTP sessions

Returns:

map of stream names and session counts

getRTPSessionCount

```
public int getRTPSessionCount(String streamName)
```

Get the number of RTP player streams playing a given stream name

Parameters:

streamName - stream name

Returns:

the number of RTP sessions

getRTPSessionCount

```
public int getRTPSessionCount()
```

Get the number of RTP sessions running under this application instance

Returns:

the number of RTP sessions running under this application instance

addRTPSession

```
public void addRTPSession(RTPSession rtpSession)
```

Add an RTP session to this application instance

Parameters:

rtpSession - RTP session to add

registerPlayRTPSession

```
public void registerPlayRTPSession(RTPSession rtpSession)
```

Register an RTP session as a play session

Parameters:

rtpSession - RTP session to register

removeRTPSession

```
public void removeRTPSession(RTPSession rtpSession)
```

Remove an RTP session from this application instance

Parameters:

rtpSession - RTP session to remove

getClientsLockObj

```
public edu.emory.mathcs.backport.java.util.concurrent.locks.WMSReadWriteLock  
getClientsLockObj()
```

Get the read/write lock for this application instance

Returns:

read/write lock for this application instance

getStreamProperties

```
public WMSProperties getStreamProperties()
```

Get the property collection of stream settings that are specific to this application instance

Returns:

property collection of stream settings

getMediaCasterProperties

```
public WMSProperties getMediaCasterProperties()
```

Get the property collection of media caster settings that are specific to this application instance

Returns:

property collection of media caster settings

getMediaReaderProperties

```
public WMSProperties getMediaReaderProperties()
```

Get the property collection of media reader settings that are specific to this application instance

Returns:

property collection of media reader settings

getMediaWriterProperties

```
public WMSProperties getMediaWriterProperties()
```

Get the property collection of media reader settings that are specific to this application instance

Returns:

property collection of media reader settings

getRTPProperties

```
public WMSProperties getRTPProperties()
```

Get the property collection of RTP settings that are specific to this application instance

Returns:

property collection of RTP settings

(continued from last page)

getLiveStreamPacketizerProperties

```
public WMSProperties getLiveStreamPacketizerProperties()
```

Get the property collection of LiveStreamPacketizer settings that are specific to this application instance

Returns:

property collection of LiveStreamPacketizer settings

getTranscoderProperties

```
public WMSProperties getTranscoderProperties()
```

Get the property collection of Transcoder settings that are specific to this application instance

Returns:

property collection of Transcoder settings

getHTTPStreamerProperties

```
public WMSProperties getHTTPStreamerProperties()
```

Get the property collection of HTTPStreamer settings that are specific to this application instance

Returns:

property collection of HTTPStreamer settings

getMaxStorageDirDepth

```
public int getMaxStorageDirDepth()
```

Maximum folder depth allowed for the StreamStorageDir and SharedObjectStorageDir paths

Returns:

folder depth

setMaxStorageDirDepth

```
public void setMaxStorageDirDepth(int maxStorageDirDepth)
```

Maximum folder depth allowed for the StreamStorageDir and SharedObjectStorageDir paths

Parameters:

maxStorageDirDepth - folder depth

getStreamFileMapper

```
public IMediaStreamFileMapper getStreamFileMapper()
```

Get the stream file mapper. See IMediaStreamFileMapper

Returns:

streamFileMapper stream file mapper

setStreamFileMapper

```
public void setStreamFileMapper(IMediaStreamFileMapper streamFileMapper)
```

(continued from last page)

Set the stream file mapper. See IMediaStreamFileMapper

Parameters:

streamFileMapper - stream file mapper

decodeStorageDir

```
public String decodeStorageDir(String storageDir)
```

This function will take a storage path that uses variables and expand the variables based on the context. It supports the following variables (as well as any system variables): `${com.wowza.wms.AppHome}`: Application home directory `${com.wowza.wms.ConfigHome}`: Configuration home directory `${com.wowza.wms.context.VHostConfigHome}`: Virtual configuration path `${com.wowza.wms.context.VHost}`: Virtual host name `${com.wowza.wms.context.Application}`: Application name `${com.wowza.wms.context.ApplicationInstance}`: Application instance name

getLiveStreamPacketizerList

```
public String getLiveStreamPacketizerList()
```

Get the comma separated list of LiveStreamPacketizers names being used by this application (see conf/LiveStreamPacketizers.xml)

Returns:

comma separated list of LiveStreamPacketizers names

getHTTPStreamerList

```
public String getHTTPStreamerList()
```

Get the comma separated list of HTTPStreamers names being used by this application (see conf/HTTPStreamers.xml)

Returns:

comma separated list of HTTPStreamers names

setLiveStreamPacketizerList

```
public void setLiveStreamPacketizerList(String liveStreamPacketizerList)
```

Set the comma separated list of LiveStreamPacketizers names being used by this application (see conf/LiveStreamPacketizers.xml)

Parameters:

liveStreamPacketizerList - comma separated list of LiveStreamPacketizers names

setHTTPStreamerList

```
public void setHTTPStreamerList(String httpStreamerList)
```

Set the comma separated list of HTTPStreamer names being used by this application (see conf/HTTPStreamers.xml)

Parameters:

httpStreamerList - comma separated list of HTTPStreamer names

containsHTTPStreamer

```
public boolean containsHTTPStreamer(String httpStreamer)
```

Does this application instance allow streaming of a given HTTPStreamer

Parameters:

(continued from last page)

httpStreamer - HTTP Streamer name

Returns:

true is this type of streaming is allowed

containsLiveStreamPacketizer

```
public boolean containsLiveStreamPacketizer(String liveStreamPacketizer)
```

Does this application instance contain a references to this live stream packetizer. If it is true we consider this a live stream source for the HTTP streamer. If false then we consider this a video on demand source.

Parameters:

liveStreamPacketizer - live stream packetizer name

Returns:

true if contains reference to it

containsDvrRecorder

```
public boolean containsDvrRecorder(String dvrRecorder)
```

Does this application instance contain a references to this DVR recorder. If it is true we consider this a DVR source for the HTTP streamer.

Parameters:

dvrRecorder - DVR recorder name

Returns:

true if contains reference to it

getVODTimedTextProviderList

```
public String getVODTimedTextProviderList()
```

Get the comma separated list of VODTimedTextProvider names being used by this application (see conf/TimedTextProviders.xml)

Returns:

comma separated list of VODTimedTextProvider names

setVODTimedTextProviderList

```
public void setVODTimedTextProviderList(String timedTextProviderList)
```

Set the comma separated list of VODTimedTextProvider names being used by this application (see conf/TimedTextProviders.xml)

Parameters:

timedTextProviderList - comma separated list of VODTimedTextProvider names

getVODTimedTextProviderSet

```
public java.util.List getVODTimedTextProviderSet()
```

(continued from last page)

getTimedTextProperties

```
public WMSProperties getTimedTextProperties()
```

Get the property collection of timed text settings that are specific to this application instance. These are defined in Application/TimedText/Properties tag in Application.xml

Returns:

property collection of Timed Text settings

getStreamNameAliasProvider

```
public IMediaStreamNameAliasProvider getStreamNameAliasProvider()
```

Get the stream name alias provider

Returns:

stream name alias provider

setStreamNameAliasProvider

```
public void setStreamNameAliasProvider(IMediaStreamNameAliasProvider  
streamNameAliasProvider)
```

Set the stream name alias provider

Parameters:

streamNameAliasProvider - stream name alias provider

getPublishers

```
public java.util.List getPublishers()
```

Get the set of server side publishers

Returns:

set of server side publishers

getPublisherCount

```
public int getPublisherCount()
```

Get the current number of server side publishers

Returns:

number of server side publishers

addPublisher

```
public void addPublisher(Publisher publisher)
```

Add a server side publisher to this application instance

Parameters:

publisher - server side publisher

(continued from last page)

removePublisher

```
public void removePublisher(Publisher publisher)
```

Remove a server side publisher from this application instance

Parameters:

publisher - server side publisher

getHTTPStreamerSessions

```
public java.util.List getHTTPStreamerSessions(int protocol,  
String streamName)
```

Get the HTTPStreamerSessions associated with this application instance for a stream name by protocol. See (IHTTPStreamerSession.SESSIONPROTOCOL_*) for protocols

Parameters:

protocol - streaming protocol (IHTTPStreamerSession.SESSIONPROTOCOL_*)
streamName - stream name

Returns:

HTTPStreamerSessions associated with this application instance

getHTTPStreamerSessions

```
public java.util.List getHTTPStreamerSessions(String streamName)
```

Get the HTTPStreamerSessions associated with this application instance for a stream name

Parameters:

streamName - stream name

Returns:

HTTPStreamerSessions associated with this application instance

getHTTPStreamerSessionCountsByName

```
public java.util.Map getHTTPStreamerSessionCountsByName(int protocol)
```

Get a map of session counts by name for a given protocol

Parameters:

protocol - streaming protocol (IHTTPStreamerSession.SESSIONPROTOCOL_*)

Returns:

map of session counts by name

getHTTPStreamerSessionCount

```
public int getHTTPStreamerSessionCount(String streamName)
```

Get the current number of HTTPStreamerSessions associated with this application instance and stream name

Parameters:

streamName - stream name

Returns:

number of HTTPStreamerSessions associated with this application instance

getHTTPStreamerSessionCount

```
public int getHTTPStreamerSessionCount(int protocol,  
    String streamName)
```

Get the current number of HTTPStreamerSessions associated with this application instance and stream name by protocol . See (IHTTPStreamerSession.SESSIONPROTOCOL_*) for protocols

Parameters:

protocol - streaming protocol (IHTTPStreamerSession.SESSIONPROTOCOL_*)
streamName - stream name

Returns:

HTTPStreamerSessions associated with this application instance

getHTTPStreamerSessions

```
public java.util.List getHTTPStreamerSessions()
```

Get the HTTPStreamerSessions associated with this application instance

Returns:

HTTPStreamerSessions associated with this application instance

getHTTPStreamerSessions

```
public java.util.List getHTTPStreamerSessions(int protocol)
```

Get the HTTPStreamerSessions associated with this application instance by protocol. See (IHTTPStreamerSession.SESSIONPROTOCOL_*) for protocols

Parameters:

protocol - streaming protocol (IHTTPStreamerSession.SESSIONPROTOCOL_*)

Returns:

HTTPStreamerSessions associated with this application instance

getHTTPStreamerSessionCount

```
public int getHTTPStreamerSessionCount()
```

Get the current number of HTTPStreamerSessions associated with this application instance

Returns:

current number of HTTPStreamerSessions associated with this application instance

getHTTPStreamerSessionCount

```
public int getHTTPStreamerSessionCount(int protocol)
```

Get the current number of HTTPStreamerSessions associated with this application instance by protocol. See (IHTTPStreamerSession.SESSIONPROTOCOL_*) for protocols

Parameters:

protocol - streaming protocol (IHTTPStreamerSession.SESSIONPROTOCOL_*)

Returns:

current number of HTTPStreamerSessions associated with this application instance

addHTTPStreamerSession

```
public void addHTTPStreamerSession(IHTTPStreamerSession httpStreamerSession)
```

Add a HTTPStreamerSession to this application instance

Parameters:

httpStreamerSession - HTTPStreamerSession

removeHTTPStreamerSession

```
public void removeHTTPStreamerSession(IHTTPStreamerSession httpStreamerSession)
```

Remove a HTTPStreamerSession from this application instance

Parameters:

httpStreamerSession - HTTPStreamerSession

getHTTPStreamerApplicationContext

```
public IHTTPStreamerApplicationContext getHTTPStreamerApplicationContext(String  
httpStreamName,  
boolean doCreate)
```

Get the HTTPStreamer application context for a given HTTPStreamer adapter

Parameters:

httpStreamName - HTTPStreamer adapter name
doCreate - create if it does not exist

Returns:

HTTPStreamer application context

addRTPIncomingDatagramPortRange

```
public void addRTPIncomingDatagramPortRange(int startPort,  
int endPort)
```

Add a port range to the list of valid incoming RTP UDP ports

Parameters:

startPort - starting port number
endPort - end port number

addRTPIncomingDatagramPortAll

```
public void addRTPIncomingDatagramPortAll()
```

Allow all incoming RTP UDP ports for this application instance

isRTPIncomingDatagramPortValid

```
public boolean isRTPIncomingDatagramPortValid(int port)
```

Check a port number to be sure it is a valid RTP UDP port for this application instance

Parameters:

port - port number

(continued from last page)

Returns:

true if the port is valid

readAppInstConfig

```
public String readAppInstConfig(String sName)
```

Method to read xml config file..

writeAppInstConfig

```
public boolean writeAppInstConfig(String sName,  
    String data)
```

Method to write xml config file..

getLiveStreamPacketizerControl

```
public ILiveStreamPacketizerControl getLiveStreamPacketizerControl()
```

Get the Live Stream Packetizer Controller. This class will get called each time a stream is to be packetized using the LiveStreamPacketizer mechanism.

Returns:

Live Stream Packetizer Controller

setLiveStreamPacketizerControl

```
public void setLiveStreamPacketizerControl(ILiveStreamPacketizerControl  
    liveStreamPacketizerControl)
```

Set the Live Stream Packetizer Controller. This class will get called each time a stream is to be packetized using the LiveStreamPacketizer mechanism.

Parameters:

liveStreamPacketizerControl - Live Stream Packetizer Controller

resetMediaCasterStream

```
public boolean resetMediaCasterStream(String streamName)
```

Reset a media caster stream

Parameters:

streamName - stream name

Returns:

true if successful

resetMediaCasterStream

```
public boolean resetMediaCasterStream(String streamName,  
    String streamExt)
```

Reset a media caster stream

Parameters:

streamName - stream name

(continued from last page)

streamExt - stream extension

Returns:

true if successful

startMediaCasterStream

```
public boolean startMediaCasterStream(String streamName,  
    String streamExt,  
    String mediaCasterType)
```

Start a media caster stream

Parameters:

streamName - stream name

streamExt - stream extension

mediaCasterType - media caster stream type

Returns:

true if successful

startMediaCasterStream

```
public boolean startMediaCasterStream(String streamName,  
    String mediaCasterType)
```

Start a media caster stream

Parameters:

streamName - stream name

mediaCasterType - media caster stream type

Returns:

true if successful

stopMediaCasterStream

```
public void stopMediaCasterStream(String streamName)
```

Stop a media caster stream

Parameters:

streamName - stream name

getContextStr

```
public String getContextStr()
```

Returns the application context string in the form [application]/[appInstance].

Returns:

application context string

getPublishStreamNames

```
public java.util.List getPublishStreamNames()
```

Get the list of live stream names currently being published.

(continued from last page)

Returns:

list of live stream names currently being published

addMediaWriterListener

```
public void addMediaWriterListener(IMediaWriterActionNotify listener)
```

Add a MediaWriter listener class. See IMediaWriterActionNotify

Parameters:

listener - MediaWriter listener class

removeMediaWriterListener

```
public void removeMediaWriterListener(IMediaWriterActionNotify listener)
```

remove MediaWriter listener class. See IMediaWriterActionNotify

Parameters:

listener - MediaWriter listener class

notifyMediaWriterOnWriteComplete

```
public void notifyMediaWriterOnWriteComplete(IMediaStream stream,  
java.io.File file)
```

Notify all MediaWriter listeners of onWriteComplete

Parameters:

stream - media stream

file - file that was written

notifyMediaWriterOnFLVAddMetadata

```
public void notifyMediaWriterOnFLVAddMetadata(IMediaStream stream,  
java.util.Map extraMetadata)
```

Notify all MediaWriter listeners of onFLVAddMetadata

Parameters:

stream - media stream

extraMetadata - meta to add to the file

getMediaCasterValidator

```
public IMediaCasterValidateMediaCaster getMediaCasterValidator()
```

Get the MediaCaster validator interface for this application instance

Returns:

MediaCaster validator interface

setMediaCasterValidator

```
public void setMediaCasterValidator(IMediaCasterValidateMediaCaster  
mediaCasterValidator)
```

Set the MediaCaster validator interface for this application instance

(continued from last page)

Parameters:

mediaCasterValidator - MediaCaster validator interface

touch

```
public void touch()
```

Touch the application instance so that it stays loaded for at least applicationInstanceTouchTimeout

getLastTouchTime

```
public long getLastTouchTime()
```

Get the last time the instance was touched (milliseconds)

Returns:

last time the instance was touched (milliseconds)

getApplicationInstanceTouchTimeout

```
public int getApplicationInstanceTouchTimeout()
```

Get the application instance touch timeout (milliseconds). Default is 5000.

Returns:

application instance touch timeout (milliseconds)

setApplicationInstanceTouchTimeout

```
public void setApplicationInstanceTouchTimeout(int applicationInstanceTouchTimeout)
```

Set the application instance touch timeout (milliseconds). Default is 5000.

Parameters:

applicationInstanceTouchTimeout - application instance touch timeout (milliseconds)

getRTSPSessionTimeout

```
public int getRTSPSessionTimeout()
```

Get the RTSP session timeout (milliseconds)

Returns:

RTSP session timeout (milliseconds)

setRTSPSessionTimeout

```
public void setRTSPSessionTimeout(int rtspSessionTimeout)
```

Set the RTSP session timeout (milliseconds)

Parameters:

rtspSessionTimeout - RTSP session timeout (milliseconds)

getRTSPMaximumPendingWriteBytes

```
public int getRTSPMaximumPendingWriteBytes()
```

(continued from last page)

Get the maximum number of pending write bytes for an RTSP session

Returns:

maximum number of pending write bytes for an RTSP session

setRTSPMaximumPendingWriteBytes

```
public void setRTSPMaximumPendingWriteBytes(int rtspMaximumPendingWriteBytes)
```

Set the maximum number of pending write bytes for an RTSP session

Parameters:

rtspMaximumPendingWriteBytes - maximum number of pending write bytes for an RTSP session

addMediaReaderListener

```
public void addMediaReaderListener(IMediaReaderActionNotify mediaReaderListener)
```

Add media reader listener. see IMediaReaderActionNotify

Parameters:

mediaReaderListener - media reader listener

removeMediaReaderListener

```
public void removeMediaReaderListener(IMediaReaderActionNotify mediaReaderListener)
```

Remove media reader listener. see IMediaReaderActionNotify

Parameters:

mediaReaderListener - media reader listener

notifyMediaReaderCreate

```
public void notifyMediaReaderCreate(IMediaReader mediaReader)
```

Notify media reader notifyMediaReaderCreate

Parameters:

mediaReader - media reader

notifyMediaReaderInit

```
public void notifyMediaReaderInit(IMediaReader mediaReader,  
    IMediaStream stream)
```

Notify media reader notifyMediaReaderInit

Parameters:

mediaReader - media reader

stream - media stream

notifyMediaReaderOpen

```
public void notifyMediaReaderOpen(IMediaReader mediaReader,  
    IMediaStream stream)
```

Notify media reader notifyMediaReaderOpen

(continued from last page)

Parameters:

mediaReader - media reader
stream - media stream

notifyMediaReaderExtractMetaData

```
public void notifyMediaReaderExtractMetaData(IMediaReader mediaReader,  
      IMediaStream stream)
```

Notify media reader notifyMediaReaderExtractMetaData

Parameters:

mediaReader - media reader
stream - media stream

notifyMediaReaderClose

```
public void notifyMediaReaderClose(IMediaReader mediaReader,  
      IMediaStream stream)
```

Notify media reader notifyMediaReaderClose

Parameters:

mediaReader - media reader
stream - media stream

getRTSPBindIpAddress

```
public String getRTSPBindIpAddress()
```

Set the IP address to which UDP ports will be bound for RTSP/RTP sessions

Returns:

IP address to which UDP ports will be bound for RTSP/RTP sessions

setRTSPBindIpAddress

```
public void setRTSPBindIpAddress(String rtspBindIpAddress)
```

Get the IP address to which UDP ports will be bound for RTSP/RTP sessions

Parameters:

rtspBindIpAddress - IP address to which UDP ports will be bound for RTSP/RTP sessions

getRTSPConnectionIpAddress

```
public String getRTSPConnectionIpAddress()
```

Get the connection IP address to used in the Session Description Protocol data exchanged for an RTSP/RTP session

Returns:

connection IP address to used in the Session Description Protocol data exchanged for an RTSP/RTP session

setRTSPConnectionIpAddress

```
public void setRTSPConnectionIpAddress(String rtspConnectionIpAddress)
```

Set the connection IP address to used in the Session Description Protocol data exchanged for an RTSP/RTP session

(continued from last page)

Parameters:

rtspConnectionIpAddress - connection IP address to used in the Session Description Protocol data exchanged for an RTSP/RTP session

getRTSPConnectionAddressType

```
public String getRTSPConnectionAddressType()
```

Get the connection IP address type (IP4) to used in the Session Description Protocol data exchanged for an RTSP/RTP session

Returns:

the connection IP address type (IP4) to used in the Session Description Protocol data exchanged for an RTSP/RTP session

setRTSPConnectionAddressType

```
public void setRTSPConnectionAddressType(String rtspConnectionAddressType)
```

Set the connection IP address type (IP4) to used in the Session Description Protocol data exchanged for an RTSP/RTP session

Parameters:

rtspConnectionAddressType

getRTSPOriginIpAddress

```
public String getRTSPOriginIpAddress()
```

Get the origin IP address to used in the Session Description Protocol data exchanged for an RTSP/RTP session

Returns:

origin IP address to used in the Session Description Protocol data exchanged for an RTSP/RTP session

setRTSPOriginIpAddress

```
public void setRTSPOriginIpAddress(String rtspOriginIpAddress)
```

Set the origin IP address to used in the Session Description Protocol data exchanged for an RTSP/RTP session

Parameters:

rtspOriginIpAddress - origin IP address to used in the Session Description Protocol data exchanged for an RTSP/RTP session

getRTSPOriginAddressType

```
public String getRTSPOriginAddressType()
```

Get the origin IP address type (IP4) to used in the Session Description Protocol data exchanged for an RTSP/RTP session

Returns:

origin IP address type (IP4) to used in the Session Description Protocol data exchanged for an RTSP/RTP session

setRTSPOriginAddressType

```
public void setRTSPOriginAddressType(String rtspOriginAddressType)
```

Set the origin IP address type (IP4) to used in the Session Description Protocol data exchanged for an RTSP/RTP session

Parameters:

(continued from last page)

rtspOriginAddressType - origin IP address type (IP4) to used in the Session Description Protocol data exchanged for an RTSP/RTP session

addLiveStreamPacketizerListener

```
public void addLiveStreamPacketizerListener(ILiveStreamPacketizerActionNotify liveStreamPacketizerListener)
```

Add a Live Stream Packetizer listener (see: [ILiveStreamPacketizerActionNotify](#))

Parameters:

liveStreamPacketizerListener - Live Stream Packetizer listener

removeLiveStreamPacketizerListener

```
public void removeLiveStreamPacketizerListener(ILiveStreamPacketizerActionNotify liveStreamPacketizerListener)
```

Remove a Live Stream Packetizer listener (see: [ILiveStreamPacketizerActionNotify](#))

Parameters:

liveStreamPacketizerListener - Live Stream Packetizer listener

notifyLiveStreamPacketizerCreate

```
public void notifyLiveStreamPacketizerCreate(ILiveStreamPacketizer liveStreamPacketizer,  
String streamName)
```

Notify Live Stream Packetizer Create

Parameters:

liveStreamPacketizer - Live Stream Packetizer listener

notifyLiveStreamPacketizerDestroy

```
public void notifyLiveStreamPacketizerDestroy(ILiveStreamPacketizer liveStreamPacketizer)
```

Notify Live Stream Packetizer Destory

Parameters:

liveStreamPacketizer - Live Stream Packetizer listener

notifyLiveStreamPacketizerInit

```
public void notifyLiveStreamPacketizerInit(ILiveStreamPacketizer liveStreamPacketizer,  
String streamName)
```

Notify Live Stream Packetizer Init

Parameters:

liveStreamPacketizer - Live Stream Packetizer listener

isValidateFMLEConnections

```
public boolean isValidateFMLEConnections()
```

Returns true if validating FMLE connection (default is false)

(continued from last page)

Returns:

true if validating FMLE connection

setValidateFMLEConnections

```
public void setValidateFMLEConnections(boolean validateFMLEConnections)
```

Returns true if validating FMLE connection (default is false)

Parameters:

validateFMLEConnections - true if validating FMLE connection

addLiveStreamTranscoderListener

```
public void addLiveStreamTranscoderListener(ILiveStreamTranscoderNotify  
liveStreamTranscoderListener)
```

Add a live stream transcoder listener

Parameters:

liveStreamTranscoderListener - live stream transcoder listener

removeLiveStreamTranscoderListener

```
public void removeLiveStreamTranscoderListener(ILiveStreamTranscoderNotify  
liveStreamTranscoderListener)
```

Remove a live stream transcoder listener

Parameters:

liveStreamTranscoderListener - live stream transcoder listener

notifyLiveStreamTranscoderCreate

```
public void notifyLiveStreamTranscoderCreate(ILiveStreamTranscoder  
liveStreamTranscoder,  
    IMediaStream stream)
```

Notify live stream transcoder create

Parameters:liveStreamTranscoder - live stream transcoder
stream - stream

notifyLiveStreamTranscoderDestroy

```
public void notifyLiveStreamTranscoderDestroy(ILiveStreamTranscoder  
liveStreamTranscoder,  
    IMediaStream stream)
```

Notify live stream transcoder destroy

Parameters:liveStreamTranscoder - live stream transcoder
stream - stream

(continued from last page)

notifyLiveStreamTranscoderInit

```
public void notifyLiveStreamTranscoderInit(ILiveStreamTranscoder liveStreamTranscoder,  
      IMediaStream stream)
```

Notify live stream transcoder init

Parameters:

liveStreamTranscoder - live stream transcoder

stream - stream

containsLiveStreamTranscoder

```
public boolean containsLiveStreamTranscoder(String liveStreamTranscoder)
```

Return true if this application instance contains the transcoder name

Parameters:

liveStreamTranscoder - transcoder name

Returns:

true if this application instance contains the transcoder name

getLiveStreamTranscoderList

```
public String getLiveStreamTranscoderList()
```

Get comma separated list of transcoders to use for this application instance

Returns:

comma separated list of transcoders

setLiveStreamTranscoderList

```
public void setLiveStreamTranscoderList(String liveStreamTranscoderList)
```

Set comma separated list of transcoders to use for this application instance

Parameters:

liveStreamTranscoderList - comma separated list of transcoders

getLiveStreamTranscoderControl

```
public ILiveStreamTranscoderControl getLiveStreamTranscoderControl()
```

Get the Live Stream Transcoder Contoller. This class will get called each time a stream is to be transcoded using the LiveStreamTranscoder mechanism.

Returns:

Live Stream Transcoder Contoller

setLiveStreamTranscoderControl

```
public void setLiveStreamTranscoderControl(ILiveStreamTranscoderControl  
liveStreamTranscoderControl)
```

Set the Live Stream Transcoder Contoller. This class will get called each time a stream is to be transcoded using the LiveStreamTranscoder mechanism.

(continued from last page)

Parameters:

liveStreamTranscoderControl - Live Stream Transcoder Contoller

getTranscoderApplicationContext

```
public LiveStreamTranscoderApplicationContext getTranscoderApplicationContext()
```

Get live stream transcoder application context

Returns:

live stream transcoder application context

getDvrProperties

```
public WMSProperties getDvrProperties()
```

Get the property collection of DVR settings that are specific to this application instance. These are defined in Application/DVR/Properties tag in Application.xml

Returns:

property collection of DVR settings

getDvrApplicationContext

```
public DvrApplicationContext getDvrApplicationContext()
```

Get live stream dvr application context

Returns:

live stream dvr application context

getLiveStreamDvrRecorderControl

```
public ILiveStreamDvrRecorderControl getLiveStreamDvrRecorderControl()
```

Get the Live Stream DVR Recorder Controller. This class will get called each time a stream is to be DVR-ed.

Returns:

Live Stream DVR Controller

setLiveStreamDvrRecorderControl

```
public void setLiveStreamDvrRecorderControl(ILiveStreamDvrRecorderControl controller)
```

Set the Live Stream DVR Controller.

Parameters:

controller - Live Stream DVR Controller

getDvrRecorderList

```
public String getDvrRecorderList()
```

Get the comma separated list of Dvr Recorder names being used by this application (see conf/Dvr.xml)

Returns:

comma separated list of Dvr Recorder names

setDvrRecorderList

```
public void setDvrRecorderList(String recorderList)
```

Set the comma separated list of Dvr Recorder names being used by this application (see conf/Dvr.xml)

Parameters:

recorderList - comma separated list of Dvr Recorder names

addDvrRecorderListener

```
public void addDvrRecorderListener(ILiveStreamDvrRecorderActionNotify listener)
```

Add a Dvr Recorder listener (see: [ILiveStreamDvrRecorderActionNotify](#))

Parameters:

listener - Dvr Recorder listener

removeDvrRecorderListener

```
public void removeDvrRecorderListener(ILiveStreamDvrRecorderActionNotify listener)
```

Remove a Dvr Recorder listener (see: [ILiveStreamDvrRecorderActionNotify](#))

Parameters:

listener - Dvr Recorder listener

notifyLiveStreamDvrRecorderCreate

```
public void notifyLiveStreamDvrRecorderCreate(ILiveStreamDvrRecorder dvr,  
String streamName)
```

Notify Dvr Recorder Create

Parameters:

dvr - DVR Recorder listener
streamName - stream Name

notifyLiveStreamDvrRecorderInit

```
public void notifyLiveStreamDvrRecorderInit(ILiveStreamDvrRecorder dvr,  
String streamName)
```

Notify DVR Recorder has been initialized.

Parameters:

dvr - DVR Recorder listener * @param streamName stream Name

notifyLiveStreamDvrRecorderDestroy

```
public void notifyLiveStreamDvrRecorderDestroy(ILiveStreamDvrRecorder dvr)
```

Notify DVR Recorder has been destroyed.

Parameters:

dvr - DVR Recorder listener

(continued from last page)

addDvrStreamManagerListener

```
public void addDvrStreamManagerListener(IDvrStreamManagerActionNotify listener)
```

Add a Dvr Application Store Manager listener (see: IDvrStoreActionNotify)

Parameters:

listener - Dvr Application Store Manager listener

removeDvrStreamManagerListener

```
public void removeDvrStreamManagerListener(IDvrStreamManagerActionNotify listener)
```

Remove a Dvr Application Store Manager listener (see: IDvrStoreActionNotify)

Parameters:

listener - Dvr Application Store Manager listener

notifyDvrStreamManagerCreate

```
public void notifyDvrStreamManagerCreate(IDvrStreamManager dvrStoreManager,  
String streamName)
```

Notify listeners that Dvr Application Store Manager has been created.

Parameters:

dvrStoreManager - Dvr Application Store Manager
streamName

notifyDvrStreamManagerInit

```
public void notifyDvrStreamManagerInit(IDvrStreamManager dvrStoreManager)
```

Notify listeners that Dvr Application Store Manager has been initialized.

Parameters:

dvrStoreManager - Dvr Application Store Manager

notifyDvrStreamManagerDestroy

```
public void notifyDvrStreamManagerDestroy(IDvrStreamManager dvrManager)
```

Notify listeners that Dvr Application Store Manager has been destroyed.

Parameters:

dvrManager - Dvr Application Store Manager

getMediaReaderContentType

```
public int getMediaReaderContentType(String mediaType)
```

Get the content type of a media stream name prefix (see IMediaReader.CONTENTTYPE_*)

Parameters:

mediaType - mediaType (such as flv or smil)

Returns:

content type (see IMediaReader.CONTENTTYPE_*)

getMediaListProvider

```
public IMediaListProvider getMediaListProvider()
```

Get the current media list provider. The media list provider is used to resolve amlst:streamname requests to a media list (equivalent to a SMIL file).

Returns:

media list provider

setMediaListProvider

```
public void setMediaListProvider(IMediaListProvider mediaListProvider)
```

Set the current media list provider. The media list provider is used to resolve amlst:streamname requests to a media list (equivalent to a SMIL file).

Parameters:

mediaListProvider - media list provider

getMediacasterRTPRTSPRTPTransportMode

```
public int getMediacasterRTPRTSPRTPTransportMode()
```

RTP MediaCaster RTSP/RTP transport mode. See RTPMediaCaster.RTSPTRANSPORTMODE_*

Returns:

RTP MediaCaster RTSP/RTP transport mode

setMediacasterRTPRTSPRTPTransportMode

```
public void setMediacasterRTPRTSPRTPTransportMode(int  
mediacasterRTPRTSPRTPTransportMode)
```

RTP MediaCaster RTSP/RTP transport mode. See RTPMediaCaster.RTSPTRANSPORTMODE_*

Parameters:

mediacasterRTPRTSPRTPTransportMode - RTP MediaCaster RTSP/RTP transport mode

getProtocolUsage

```
public boolean[] getProtocolUsage()
```

Get the protocols in use by this application instance (see IApplicationInstance.PROTOCOLUSAGE_*)

Returns:

protocols in use by this application instance (see IApplicationInstance.PROTOCOLUSAGE_*)

getProtocolUsage

```
public void getProtocolUsage(boolean[] protocolsInUse)
```

Get the protocols in use by this application instance (see IApplicationInstance.PROTOCOLUSAGE_*)

isDebugAppTimeout

```
public boolean isDebugAppTimeout()
```

(continued from last page)

If true appTimeout processing will be logged.

Returns:

true appTimeout processing will be logged

setDebugAppTimeout

```
public void setDebugAppTimeout(boolean debugAppTimeout)
```

If true appTimeout processing will be logged.

Parameters:

debugAppTimeout - true appTimeout processing will be logged

addRTPListener

```
public void addRTPListener(IRTPSessionNotify rtpListener)
```

Add listner to be notified of RTPSession create and destroy.

Parameters:

rtpListener - RTP listener

removeRTPListener

```
public void removeRTPListener(IRTPSessionNotify rtpListener)
```

Remove listner to be notified of RTPSession create and destroy.

Parameters:

rtpListener - RTP listener

notifyRTPSessionCreate

```
public void notifyRTPSessionCreate(RTPSession rtpSession)
```

Notify RTP session create

Parameters:

rtpSession - rtp session

notifyRTPSessionDestroy

```
public void notifyRTPSessionDestroy(RTPSession rtpSession)
```

Notify RTP session destory

Parameters:

rtpSession - rtp session

addHTTPListener

```
public void addHTTPListener(IHTTPSessionNotify httpListener)
```

Add listner to be notified of HTTPSession create and destroy.

Parameters:

httpListener - http listener

removeHTTPListener

```
public void removeHTTPListener(IHTTPSessionNotify httpListener)
```

Remove listner to be notified of HTTPSession create and destroy.

Parameters:

httpListener - http listener

notifyHTTPSessionCreate

```
public void notifyHTTPSessionCreate(IHTTPStreamerSession httpSession)
```

Notify HTTP session create

Parameters:

httpSession - HTTP session

notifyHTTPSessionDestroy

```
public void notifyHTTPSessionDestroy(IHTTPStreamerSession httpSession)
```

Notify HTTP session destroy

Parameters:

httpSession - HTTP session

getMediaCacheFilters

```
public java.util.List getMediaCacheFilters()
```

Get the list of allowed MediaCache sources.

Returns:

list of allowed MediaCache sources

isMediaCacheSourceAllowed

```
public boolean isMediaCacheSourceAllowed(String sourceName)
```

Test to see if a MediaCache source name is allowed

Parameters:

sourceName - MediaCache source name

Returns:

true if allowed

getPushPublishSessions

```
public java.util.List getPushPublishSessions(IMediaStream stream)
```

Get the current push publish sessions for a given IMediaStream

Parameters:

stream - stream

(continued from last page)

Returns:

push publish sessions

getPushPublishSessions

```
public java.util.List getPushPublishSessions()
```

Get all active push publish sessions

Returns:

all active push publish sessions

addPushPublishSession

```
public void addPushPublishSession(IMediaStream stream,  
    IPushPublishSession pushPublishSession)
```

Add a push publish session to the list of active push publish sessions

Parameters:

stream - stream

pushPublishSession - push publish session

removePushPublishSession

```
public void removePushPublishSession(IMediaStream stream,  
    IPushPublishSession pushPublishSession)
```

Remove a push publish session

Parameters:

stream - stream

pushPublishSession - push publish session

removePushPublishSessions

```
public void removePushPublishSessions(IMediaStream stream)
```

Remove all push publish sessions for a give stream

Parameters:

stream - stream

addPushPublishSessionListener

```
public void addPushPublishSessionListener(IPushPublishSessionNotify  
    pushPublishSessionListener)
```

Add a push publish session listener (see [IPushPublishSessionNotify](#))**Parameters:**

pushPublishSessionListener - push publish session

removePushPublishSessionListener

```
public void removePushPublishSessionListener(IPushPublishSessionNotify  
    pushPublishSessionListener)
```

Remove a push publish session listener (see [IPushPublishSessionNotify](#))

(continued from last page)

Parameters:

pushPublishSessionListener - push publish session

notifyPushPublishSessionCreate

```
public void notifyPushPublishSessionCreate(String streamName,  
    IMediaStream stream,  
    IPushPublishSession pushPublishSession)
```

Notify push publish session create

Parameters:

streamName - stream name

stream - stream

pushPublishSession - push publish session

notifyPushPublishSessionDestroy

```
public void notifyPushPublishSessionDestroy(String streamName,  
    IMediaStream stream,  
    IPushPublishSession pushPublishSession)
```

Notify push publish session destroy

Parameters:

streamName - stream name

stream - stream

pushPublishSession - push publish session

getPushPublishSessionCount

```
public int getPushPublishSessionCount()
```

Get the number of push publishing sessions.

Returns:

number of push publishing sessions

getStreamRecorderProperties

```
public WMSPProperties getStreamRecorderProperties()
```

Get the property collection of StreamRecorder settings that are specific to this application instance

Returns:

property collection of StreamRecorder settings

isLive

```
public boolean isLive()
```

Returns true if the application stream type of the application instance is live

Returns:

com.wowza.wms.application Interface IApplicationInstanceNotify

public interface **IApplicationInstanceNotify**
extends

IApplicationInstanceNotify: listener interface used by IApplication addApplicationInstanceListener

Method Summary

void	onApplicationInstanceCreate (IApplicationInstance applicationInstance) Triggered when applicationInstance created
void	onApplicationInstanceDestroy (IApplicationInstance applicationInstance) Triggered when applicationInstance destroyed

Methods

onApplicationInstanceCreate

public void **onApplicationInstanceCreate**([IApplicationInstance](#) applicationInstance)

Triggered when applicationInstance created

Parameters:

applicationInstance - applicationInstance

onApplicationInstanceDestroy

public void **onApplicationInstanceDestroy**([IApplicationInstance](#) applicationInstance)

Triggered when applicationInstance destroyed

Parameters:

applicationInstance - applicationInstance

com.wowza.wms.application Interface IApplicationNotify

public interface **IApplicationNotify**
extends

IApplicationNotify: listener interface used by IVHost addApplicationListener

Method Summary

void	onApplicationCreate (IApplication application) Triggered when application created
void	onApplicationDestroy (IApplication application) Triggered when application destroyed

Methods

onApplicationCreate

public void **onApplicationCreate**([IApplication](#) application)

Triggered when application created

Parameters:

application - application

onApplicationDestroy

public void **onApplicationDestroy**([IApplication](#) application)

Triggered when application destroyed

Parameters:

application - application

com.wowza.wms.application Class WMSProperties

```

java.lang.Object
  |-- java.util.AbstractMap
        |-- java.util.HashMap
              |-- com.wowza.wms.application.WMSProperties
  
```

All Implemented Interfaces:

java.util.Map, java.io.Serializable, Cloneable, java.util.Map

```

public class WMSProperties
extends java.util.HashMap
  
```

WMSProperties: generic properties container used by many class to store extended property information. Acts like a simple Map with some simple utilities for performing type conversion.

Constructor Summary

public	WMSProperties()
--------	---------------------------------

Method Summary

static void	cloneProperties (WMSProperties from, WMSProperties to) Copy all properties from "from" properties object to "to" properties object.
String[]	getAllAsStrings () Return all properties as String[].
Object	getProperty (String name) Get property value as generic object.
boolean	getPropertyBoolean (String name, boolean defaultVal) Get property as boolean, return default value if does not exist.
double	getPropertyDouble (String name, double defaultVal) Get property as double, return default value if does not exist.
int	getPropertyInt (String name, int defaultVal) Get property as int, return default value if does not exist.
long	getPropertyLong (String name, long defaultVal) Get property as long, return default value if does not exist.
String	getPropertyStr (String name) Get property as String
String	getPropertyStr (String name, String defaultVal) Get property as String, return default value if does not exist.
void	putAll (java.util.Map m)

void	setProperty (String name, Object value) Set property to generic object.
String	toString ()

Methods inherited from class java.util.HashMap

clear, clone, containsKey, containsValue, entrySet, get, isEmpty, keySet, put, putAll, remove, size, values

Methods inherited from class java.util.AbstractMap

clear, clone, containsKey, containsValue, entrySet, equals, get, hashCode, isEmpty, keySet, put, putAll, remove, size, toString, values

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface java.util.Map

clear, containsKey, containsValue, entrySet, equals, get, hashCode, isEmpty, keySet, put, putAll, remove, size, values

Methods inherited from interface java.util.Map

clear, containsKey, containsValue, entrySet, equals, get, hashCode, isEmpty, keySet, put, putAll, remove, size, values

Constructors

WMSProperties

```
public WMSProperties()
```

Methods

cloneProperties

```
public static void cloneProperties(WMSProperties from,
    WMSProperties to)
```

Copy all properties from "from" properties object to "to" properties object.

Parameters:

from - source properties
to - destination properties

setProperty

```
public void setProperty(String name,
    Object value)
```

(continued from last page)

Set property to generic object.

Parameters:

name - property name
value - value

getProperty

```
public Object getProperty(String name)
```

Get property value as generic object.

Parameters:

name - property name

Returns:

value, null if does not exist

getPropertyStr

```
public String getPropertyStr(String name)
```

Get property as String

Parameters:

name - property name

Returns:

value as String, null if does not exist

getPropertyStr

```
public String getPropertyStr(String name,  
    String defaultVal)
```

Get property as String, return default value if does not exist.

Parameters:

name - property name
defaultVal - default value

Returns:

value as String, defaultVal if does not exist

getPropertyBoolean

```
public boolean getPropertyBoolean(String name,  
    boolean defaultVal)
```

Get property as boolean, return default value if does not exist.

Parameters:

name - property name
defaultVal - default value

Returns:

value as boolean, defaultVal if does not exist

getPropertyInt

```
public int getPropertyInt(String name,  
    int defaultVal)
```

Get property as int, return default value if does not exist.

Parameters:

name - property name
defaultVal - default value

Returns:

value as boolean, defaultVal if does not exist

getPropertyLong

```
public long getPropertyLong(String name,  
    long defaultVal)
```

Get property as long, return default value if does not exist.

Parameters:

name - property name
defaultVal - default value

Returns:

value as long, defaultVal if does not exist

getPropertyDouble

```
public double getPropertyDouble(String name,  
    double defaultVal)
```

Get property as double, return default value if does not exist.

Parameters:

name - property name
defaultVal - default value

Returns:

value as double, defaultVal if does not exist

getAllAsStrings

```
public String[] getAllAsStrings()
```

Return all properties as String[]. Format is "key=value".

Returns:

all properties as String[]

toString

```
public String toString()
```

(continued from last page)

putAll

```
public void putAll(java.util.Map m)
```

Package

com.wowza.wms.authentication

com.wowza.wms.authentication

Class AuthenticateUsernamePasswordProviderBase

java.lang.Object

└-com.wowza.wms.authentication.AuthenticateUsernamePasswordProviderBase

All Implemented Interfaces:

[IAuthenticateUsernamePasswordProvider](#)

public abstract class **AuthenticateUsernamePasswordProviderBase**

extends Object

implements [IAuthenticateUsernamePasswordProvider](#)

Base class for implementing HTTP and RTSP based custom authentication class.

Field Summary

protected	client
protected	rtpSession
protected	vhost

Constructor Summary

public	AuthenticateUsernamePasswordProviderBase()
--------	--

Method Summary

IClient	getClient() Get client
RTPSession	getRTPSession() Get RTP Session
IVHost	getVHost() Get vhost
void	setClient(IClient client) Set client
void	setRTPSession(RTPSession rtpSession) Set RTP Session
void	setVHost(IVHost vhost) Set vhost

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface[com.wowza.wms.authentication.IAuthenticateUsernamePasswordProvider](#)[getClient](#), [getPassword](#), [getRTPSession](#), [getVHost](#), [setClient](#), [setRTPSession](#), [setVHost](#), [userExists](#)

Fields

vhostprotected com.wowza.wms.vhost.IVHost **vhost****client**protected com.wowza.wms.client.IClient **client****rtpSession**protected com.wowza.wms.rtp.model.RTPSession **rtpSession**

Constructors

AuthenticateUsernamePasswordProviderBasepublic **AuthenticateUsernamePasswordProviderBase**()

Methods

getVHostpublic [IVHost](#) **getVHost**()

Get vhost

Returns:

vhost

setVHostpublic void **setVHost**([IVHost](#) vhost)

Set vhost

Parameters:

vhost - vhost

(continued from last page)

getRTPSession

```
public RTPSession getRTPSession()
```

Get RTP Session

Returns:

RTP Session

setRTPSession

```
public void setRTPSession(RTPSession rtpSession)
```

Set RTP Session

Parameters:

rtpSession - RTP Session

getClient

```
public IClient getClient()
```

Get client

Returns:

client

setClient

```
public void setClient(IClient client)
```

Set client

Parameters:

client - client

com.wowza.wms.authentication Interface IAuthenticate

public interface **IAuthenticate**
extends

IAuthenticate: HTTP and RTSP authentication interface

Field Summary

public static final	PASSWORDFILEFORMAT_CLEAR Value: 1
public static final	PASSWORDFILEFORMAT_UNKNOWN Value: 0

Method Summary

void	init (IApplicationInstance appInstance, AuthenticationItem authenticationItem) Initialize authentication class when instantiated as part of an application instance
void	init (IVHost vhost, AuthenticationItem authenticationItem) Initialize authentication class when instantiated as part of a vhost

Fields

PASSWORDFILEFORMAT_UNKNOWN

public static final int **PASSWORDFILEFORMAT_UNKNOWN**

Constant value: **0**

PASSWORDFILEFORMAT_CLEAR

public static final int **PASSWORDFILEFORMAT_CLEAR**

Constant value: **1**

Methods

init

public void **init**([IApplicationInstance](#) appInstance,
AuthenticationItem authenticationItem)

Initialize authentication class when instantiated as part of an application instance

(continued from last page)

Parameters:

appInstance - application instance
authenticationItem - authentication item

init

```
public void init(IVHost vhost,  
                AuthenticationItem authenticationItem)
```

Initialize authentication class when instantiated as part of a vhost

Parameters:

vhost - vhost
authenticationItem - authentication item

com.wowza.wms.authentication
Interface IAuthenticateHTTPProvider

public interface IAuthenticateHTTPProvider
extends

IAuthenticateHTTPProvider: HTTP authentication provider

Method Summary	
boolean	authenticateHTTPProvider (IVHost vhost, IHTTPRequest req, IHTTPResponse resp) Called for each HTTP authentication

Methods

authenticateHTTPProvider

public boolean **authenticateHTTPProvider**([IVHost](#) vhost, [IHTTPRequest](#) req, [IHTTPResponse](#) resp)

Called for each HTTP authentication

Parameters:

- vhost - virtual host
- req - HTTP request
- resp - HTTP response

Returns:

true if authentication was successful

com.wowza.wms.authentication Interface IAuthenticateRTSP

public interface **IAuthenticateRTSP**
extends

IAuthenticateRTSP: RTSP authentication provider

Method Summary

boolean	<code>authenticateRTSP(RTPSession rtspSession, com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp)</code> Called for each RTSP authentication
---------	---

Methods

authenticateRTSP

```
public boolean authenticateRTSP(RTPSession rtspSession,  
    com.wowza.wms.rtsp.RTSPRequestMessage req,  
    com.wowza.wms.rtsp.RTSPResponseMessages resp)
```

Called for each RTSP authentication

Parameters:

`rtspSession` - RTP session
`req` - RTSP request
`resp` - RTSP response

Returns:

true if authentication was successful

com.wowza.wms.authentication Interface IAuthenticateSIP

public interface **IAuthenticateSIP**
extends

IAuthenticateRTSP: RTSP authentication provider

Method Summary

boolean	<code>authenticateSIP(RTPSession rtspSession, com.wowza.wms.sip.SIPRequestMessage req, com.wowza.wms.sip.SIPResponseMessages resp)</code> Called for each RTSP authentication
---------	--

Methods

authenticateSIP

```
public boolean authenticateSIP(RTPSession rtspSession,  
    com.wowza.wms.sip.SIPRequestMessage req,  
    com.wowza.wms.sip.SIPResponseMessages resp)
```

Called for each RTSP authentication

Parameters:

`rtspSession` - RTP session
`req` - RTSP request
`resp` - RTSP response

Returns:

true if authentication was successful

com.wowza.wms.authentication Interface IAuthenticateUsernamePasswordProvider

All Known Implementing Classes:

[AuthenticateUsernamePasswordProviderBase](#)

public interface **IAuthenticateUsernamePasswordProvider**
extends

IAuthenticateUsernamePasswordProvider: HTTP and RTSP username/password validator interface

Method Summary

IClient	getClient() Get client
String	getPassword(String username) Get password for a given user
RTPSession	getRTPSession() Get RTP session
IVHost	getVHost() Get vhost
void	setClient(IClient client) Set client
void	setRTPSession(RTPSession rtpSession) Set RTP session
void	setVHost(IVHost vhost) Set vhost
boolean	userExists(String username) Return true if user exists

Methods

userExists

public boolean **userExists**(String username)

Return true if user exists

Parameters:

username - username

Returns:

true if user exists

(continued from last page)

getPassword

```
public String getPassword(String username)
```

Get password for a given user

Parameters:

username - username

Returns:

password

getVHost

```
public IVHost getVHost()
```

Get vhost

Returns:

vhost

setVHost

```
public void setVHost(IVHost vhost)
```

Set vhost

Parameters:

vhost - vhost

getClient

```
public IClient getClient()
```

Get client

Returns:

client

setClient

```
public void setClient(IClient client)
```

Set client

Parameters:

client - client

getRTPSession

```
public RTPSession getRTPSession()
```

Get RTP session

Returns:

RTP session

(continued from last page)

setRTPSession

```
public void setRTPSession(RTPSession rtpSession)
```

Set RTP session

Parameters:

`rtpSession` - RTP session

Package

com.wowza.wms.client

com.wowza.wms.client Class ConnectionCounter

```
java.lang.Object
|
+-com.wowza.wms.client.ConnectionCounterBase
|
+-com.wowza.wms.client.ConnectionCounter
```

```
public class ConnectionCounter
extends ConnectionCounterBase
```

Fields inherited from class com.wowza.wms.client.ConnectionCounterBase

DATEFORMATSTR, NEVERSTR, REJECTREASON_APPLICATION, REJECTREASON_COUNT, REJECTREASON_LICENSE

Constructor Summary

public	ConnectionCounter() Create empty ConnectionCounter
--------	---

Method Summary

void	acceptConnection (ConnectionHolder connectionHolder, byte[] license) Accept a new connection.
void	addConnectionListener (IConnectionNotify connectionNotify) Add a connection listener.
void	decrement (ConnectionHolder connectionHolder, boolean isValid, java.util.Date date, long stamp, byte[] license) Decrement connection counters.
void	disconnect (ConnectionHolder connectionHolder, byte[] license) Disconnect connection.
long	getCurrent () Get total number of client currently conencted to this object.
long	getLastConnectAcceptedStamp () Get time (milliseconds) of the last conenction to this object.
String	getLastConnectAcceptedStampString () Get time (milliseconds) of the last conenction to this object as formatted string.
java.util.Date	getLastConnectAcceptedTime () Get time (milliseconds) of the last accepted conenction to this object.
String	getLastConnectAcceptedTimeString () Get time (milliseconds) of the last accepted conenction to this object as formatted string.
long	getLastConnectRejectedByReasonStamp (int reason) Get time (milliseconds) of the last rejected conenction by reason to this object.

String	<u>getLastConnectRejectedByReasonStampString</u> (int reason) Get time (milliseconds) of the last rejected connection by reason to this object as formatted string.
java.util.Date	<u>getLastConnectRejectedByReasonTime</u> (int reason) Get date and time of last rejected connection by reason to this object as Date object.
String	<u>getLastConnectRejectedByReasonTimeString</u> (int reason) Get date and time of last rejected connection by reason to this object as formatted string.
long	<u>getLastConnectRejectedStamp</u> () Get time (milliseconds) of the last rejected connection to this object.
String	<u>getLastConnectRejectedStampString</u> () Get time (milliseconds) of the last rejected connection to this object as formatted string.
java.util.Date	<u>getLastConnectRejectedTime</u> () Get date and time of last rejected connection to this object as Date object.
String	<u>getLastConnectRejectedTimeString</u> () Get date and time of last rejected connection to this object as formatted string.
long	<u>getLastDisconnectStamp</u> () Get time (milliseconds) of the last disconnected connection to this object.
String	<u>getLastDisconnectStampString</u> () Get time (milliseconds) of the last disconnected connection to this object as formatted string.
java.util.Date	<u>getLastDisconnectTime</u> () Get date and time of last disconnected connection to this object as Date object.
String	<u>getLastDisconnectTimeString</u> () Get date and time of last disconnected connection to this object as Date object as formatted string.
long	<u>getTotal</u> () Get total number of connection attempts to this object.
long	<u>getTotalAccepted</u> () Get total number of accepted connections to this object.
long	<u>getTotalRejected</u> () Get total number of rejected connections to this object.
void	<u>incrementAccept</u> (ConnectionHolder connectionHolder, java.util.Date date, long stamp, byte[] license) Increment accepted connections.
void	<u>incrementReject</u> (ConnectionHolder connectionHolder, int reason, java.util.Date date, long stamp, byte[] license) Increment reject connection.
void	<u>rejectConnection</u> (ConnectionHolder connectionHolder, int reason, byte[] license) Reject connection.
void	<u>removeConnectionListener</u> (IConnectionNotify connectionNotify) Remove connection listener
void	<u>setConnectionValidator</u> (IConnectionValidator connectionValidator) Set the connection validator.

boolean	validateNewConnection (ConnectionHolder connectionHolder, byte[] license) Validate a new connection.
---------	---

Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Constructors

ConnectionCounter

```
public ConnectionCounter()
```

Create empty ConnectionCounter

Methods

addConnectionListener

```
public void addConnectionListener(IConnectionNotify connectionNotify)
```

Add a connection listener. Receives following events: `onClientConnect`, `onClientDisconnect`, `onClientAccept` and `onClientReject`.

Parameters:

`connectionNotify` - connection listener

removeConnectionListener

```
public void removeConnectionListener(IConnectionNotify connectionNotify)
```

Remove connection listener

Parameters:

`connectionNotify` - connection listener

setConnectionValidator

```
public void setConnectionValidator(IConnectionValidator connectionValidator)
```

Set the connection validator.

Parameters:

`connectionValidator` - connection validator

validateNewConnection

```
public boolean validateNewConnection(ConnectionHolder connectionHolder,  
byte[] license)
```

Validate a new connection.

NOTE: This is a private internal call. Server will not work properly if this method is not called properly using internal security mechanism.

(continued from last page)

Parameters:

connectionHolder - connection holder
license - security data

Returns:

true if connection is accepted

acceptConnection

```
public void acceptConnection(ConnectionHolder connectionHolder,  
    byte[] license)
```

Accept a new connection.

NOTE: This is a private internal call. Server will not work properly if this method is not called properly using internal security mechanism.

Parameters:

connectionHolder - connection holder
license - security data

incrementAccept

```
public void incrementAccept(ConnectionHolder connectionHolder,  
    java.util.Date date,  
    long stamp,  
    byte[] license)
```

Increment accepted connections.

NOTE: This is a private internal call. Server will not work properly if this method is not called properly using internal security mechanism.

Parameters:

connectionHolder - connection holder
date - date the connection occurred
stamp - time stamp connection occurred (milliseconds)
license - security data

rejectConnection

```
public void rejectConnection(ConnectionHolder connectionHolder,  
    int reason,  
    byte[] license)
```

Reject connection.

NOTE: This is a private internal call. Server will not work properly if this method is not called properly using internal security mechanism.

Parameters:

connectionHolder - connection holder
reason - reason the connection was refused. See REJECTREASON_*
license - security data

(continued from last page)

incrementReject

```
public void incrementReject(ConnectionHolder connectionHolder,  
    int reason,  
    java.util.Date date,  
    long stamp,  
    byte[] license)
```

Increment reject connection.

NOTE: This is a private internal call. Server will not work properly if this method is not called properly using internal security mechanism.

Parameters:

connectionHolder - connection holder
reason - reason the connection was refused. See REJECTREASON_*
date - date the connection occurred
stamp - time stamp connection occurred (milliseconds)
license - security data

disconnect

```
public void disconnect(ConnectionHolder connectionHolder,  
    byte[] license)
```

Disconnect connection.

NOTE: This is a private internal call. Server will not work properly if this method is not called properly using internal security mechanism.

Parameters:

connectionHolder - connection holder
license - security data

decrement

```
public void decrement(ConnectionHolder connectionHolder,  
    boolean isValid,  
    java.util.Date date,  
    long stamp,  
    byte[] license)
```

Decrement connection counters.

NOTE: This is a private internal call. Server will not work properly if this method is not called properly using internal security mechanism.

Parameters:

connectionHolder - connection holder
isValid - is a valid connection
date - date the connection occurred
stamp - time stamp connection occurred (milliseconds)
license - security data

getCurrent

```
public long getCurrent()
```

Get total number of client currently connected to this object.

Returns:

total number of client currently connected to this object

getLastConnectAcceptedStamp

```
public long getLastConnectAcceptedStamp()
```

Get time (milliseconds) of the last conenction to this object.

Returns:

time (milliseconds) of the last conenction to this object

getLastConnectAcceptedStampString

```
public String getLastConnectAcceptedStampString()
```

Get time (milliseconds) of the last conenction to this object as formatted string.

Returns:

time (milliseconds) of the last conenction to this object as formatted string

getLastConnectAcceptedTime

```
public java.util.Date getLastConnectAcceptedTime()
```

Get time (milliseconds) of the last accepeted conenction to this object.

Returns:

time (milliseconds) of the last accepeted conenction to this object

getLastConnectAcceptedTimeString

```
public String getLastConnectAcceptedTimeString()
```

Get time (milliseconds) of the last accepeted conenction to this object as formatted string.

Returns:

time (milliseconds) of the last accepeted conenction to this object as formatted string

getLastConnectRejectedStamp

```
public long getLastConnectRejectedStamp()
```

Get time (milliseconds) of the last rejected conenction to this object.

Returns:

time (milliseconds) of the last rejected conenction to this object

getLastConnectRejectedStampString

```
public String getLastConnectRejectedStampString()
```

Get time (milliseconds) of the last rejected conenction to this object as formatted string.

Returns:

time (milliseconds) of the last rejected conenction to this object as formatted string

getLastConnectRejectedTime

```
public java.util.Date getLastConnectRejectedTime()
```

(continued from last page)

Get date and time of last rejected connection to this object as Date object.

Returns:

date and time of last reject connection to this object as Date object

getLastConnectRejectedTimeString

```
public String getLastConnectRejectedTimeString()
```

Get date and time of last rejected connection to this object as formatted string.

Returns:

date and time of last reject connection to this object as formatted string

getLastConnectRejectedByReasonStampString

```
public String getLastConnectRejectedByReasonStampString(int reason)
```

Get time (milliseconds) of the last rejected connection by reason to this object as formatted string.

Parameters:

reason - reason, see REJECTREASON_*

Returns:

time (milliseconds) of the last rejected connection by reason to this object as formatted string

getLastConnectRejectedByReasonStamp

```
public long getLastConnectRejectedByReasonStamp(int reason)
```

Get time (milliseconds) of the last rejected connection by reason to this object.

Parameters:

reason - reason, see REJECTREASON_*

Returns:

time (milliseconds) of the last rejected connection by reason to this object

getLastConnectRejectedByReasonTime

```
public java.util.Date getLastConnectRejectedByReasonTime(int reason)
```

Get date and time of last rejected connection by reason to this object as Date object.

Parameters:

reason - reason, see REJECTREASON_*

Returns:

date and time of last reject connection by reason to this object as Date object

getLastConnectRejectedByReasonTimeString

```
public String getLastConnectRejectedByReasonTimeString(int reason)
```

Get date and time of last rejected connection by reason to this object as formatted string.

Parameters:

reason - reason, see REJECTREASON_*

(continued from last page)

Returns:

date and time of last rejected connection by reason to this object as formatted string

getLastDisconnectStampString

```
public String getLastDisconnectStampString()
```

Get time (milliseconds) of the last disconnected connection to this object as formatted string.

Returns:

time (milliseconds) of the last disconnected connection to this object as formatted string

getLastDisconnectStamp

```
public long getLastDisconnectStamp()
```

Get time (milliseconds) of the last disconnected connection to this object.

Returns:

time (milliseconds) of the last disconnected connection to this object

getLastDisconnectTime

```
public java.util.Date getLastDisconnectTime()
```

Get date and time of last disconnected connection to this object as Date object.

Returns:

date and time of last disconnected connection to this object as Date object

getLastDisconnectTimeString

```
public String getLastDisconnectTimeString()
```

Get date and time of last disconnected connection to this object as Date object as formatted string.

Returns:

date and time of last disconnected connection to this object as Date object as formatted string

getTotal

```
public long getTotal()
```

Get total number of connection attempts to this object.

Returns:

total number of connection attempts to this object

getTotalAccepted

```
public long getTotalAccepted()
```

Get total number of accepted connections to this object.

Returns:

total number of accepted connections to this object

getTotalRejected

```
public long getTotalRejected()
```

Get total number of rejected connections to this object.

Returns:

total number of rejected connections to this object

com.wowza.wms.client Interface IClient

public interface **IClient**
extends

IClient: public interface to Client object.

Field Summary

public static final	AUDIOSAMPLE_ACCESS_ALL Value: *
public static final	AUDIOSAMPLE_ACCESS_NONE Value:
public static final	READ_ACCESS_ALL Value: *
public static final	READ_ACCESS_NONE Value:
public static final	VIDEOSAMPLE_ACCESS_ALL Value: *
public static final	VIDEOSAMPLE_ACCESS_NONE Value:
public static final	WRITE_ACCESS_ALL Value: *
public static final	WRITE_ACCESS_NONE Value:

Method Summary

void	acceptConnection () Accept connection
void	acceptConnection (AMFData successObj) Accept connection
void	acceptConnection (String successStr) Accept connection
void	addAcceptConnectionAttribute (String key, AMFDataObj item) Add and attribute to the resultObj that gets passed back to the client on successful connection

void	<u>addAcceptConnectionAttribute</u> (String key, String item) Add and attribute to the resultObj that gets passed back to the client on successful connection
void	<u>call</u> (String handlerName) Simplified call client method/handler call.
void	<u>call</u> (String handlerName, <u>IModuleCallResult</u> resultObj, Object[] params) Call client method/handler.
void	<u>clearFastPlaySettings</u> () Force clear the fastPlay settings
void	<u>fcSubscribe</u> (String streamName) Subscribe to a live stream (for live stream repeater to start start from edge to origin)
void	<u>fcSubscribe</u> (String streamName, String mediaCasterType) Subscribe to a live stream (use a particular mediaCasterType)
void	<u>fcUnSubscribe</u> (String streamName) UnSubscribe from a stream
void	<u>fcUnSubscribeAll</u> () UnSubscribe to all streams that this client is current subscribed to
<u>IApplicationInstance</u>	<u>getAppInstance</u> () Get parent applicationInstance.
<u>IApplication</u>	<u>getApplication</u> () Get parent application.
int	<u>getBufferTime</u> () Get default buffer time for newly created mediaStream objects
int	<u>getClientId</u> () Get client id.
long	<u>getConnectTime</u> () Get time in milliseconds the client connected to the server.
String	<u>getDateStarted</u> () Get date and time of client connection
<u>ElapsedTimer</u>	<u>getElapsedTime</u> () Get elapsed time client has been connected.
<u>FastPlaySettings</u>	<u>getFastPlaySettings</u> () Get the current fastPlay settings.
String	<u>getFlashVer</u> () Get client flash version
int	<u>getIdleFrequency</u> () Get client idle frequency (milliseconds)
String	<u>getIp</u> () Client ip address
long	<u>getLastValidateTime</u> () Get last time (millisecond) the connection was validated with a ping

int	<u>getLiveRepeaterCapabilities()</u> Get the live repeater capabilities of this connection
String	<u>getLiveStreamPacketizerList()</u> Get the comma separated list of LiveStreamPacketizers names being used by this client (see conf/LiveStreamPacketizers.xml)
String	<u>getLiveStreamTranscoderList()</u> Get the comma separated list of LiveStreamTranscoders names being used by this client (see conf/LiveStreamTranscoders.xml)
int	<u>getMaximumPendingWriteBytes()</u> Get maximum number a bytes a client connection can have waiting to be sent before the connection is terminated.
int	<u>getMaximumSetBufferTime()</u> Get maximum number of milliseconds allowed for the NetStream.setBufferTime(secs) call.
<u>IOPerformanceCounter</u>	<u>getMediaIOPerformanceCounter()</u> Get performance counter for media bytes transferred to this client.
int	<u>getObjectEncoding()</u> Get the object encoding level
String	<u>getPageUrl()</u> Get the pageUrl for this connection.
long	<u>getPingRoundTripTime()</u> Get turn around time (milliseconds) of last ping request
int	<u>getPingTimeout()</u> Get the ping timeout (milliseconds)
int	<u>getPlayStreamCount()</u> Get the number of play streams
java.util.List	<u>getPlayStreams()</u> Get a collection of all play streams.
<u>WMSProperties</u>	<u>getProperties()</u> Get client properties
int	<u>getProtocol()</u> Get connection protocol (1 = RTMP, 3 = RTMPT)
int	<u>getPublishStreamCount()</u> Get the number of publish streams
java.util.List	<u>getPublishStreams()</u> Get a collection of publish streams.
String	<u>getQueryStr()</u> Get the query string part of the connection string.
String	<u>getReferrer()</u> Get the referrer data for this connection.
String	<u>getRepeaterOriginUrl()</u> Get the origin URL used by the Live Stream Repeater

AMFObj	getRespAMFAudioObj (IMediaStream stream) Get the audio response AMFObj for a given mediaStream
AMFObj	getRespAMFDataObj (IMediaStream stream) Get the data response AMFObj for a given mediaStream
AMFObj	getRespAMFVideoObj (IMediaStream stream) Get the video response AMFObj for a given mediaStream
ResponseFunctions	getRespFunctions () Get client responseFunctions object.
AMFObj	getResponseAMFObj (int index) Get the response channel AMFObj for channel index.
RTPStream	getRTPStream () If this client was created due to an RTP connection to the server return the underlying RTPStream object
HostPort	getServerHostPort () Get the hostPort object for the connection that is servicing this client
String	getSharedObjectReadAccess () Get the shared object read access value.
String	getSharedObjectWriteAccess () Get the shared object write access value.
String	getStreamAudioSampleAccess () Get the audio sample access value.
java.io.File	getStreamFile (String streamName) Get File object for stream with given name.
java.io.File	getStreamFile (String streamName, String streamExt) Get File object for stream with given name and extension.
java.io.File	getStreamFile (String streamName, String streamExt, boolean doCreateFolder) Get File object for stream with given name and extension.
String	getStreamReadAccess () Get the stream read access value.
String	getStreamType () Get default streamType
String	getStreamVideoSampleAccess () Get the video sample access value.
String	getStreamWriteAccess () Get the stream write access value.
String	getTimeRunning () Get elapsed time of connection
double	getTimeRunningSeconds () Get time running in seconds

<u>IOPerformanceCounter</u>	<u>getTotalIOPerformanceCounter()</u> Get performance counter for all bytes transferred to this client.
String	<u>getUri()</u> Get the full URI of the connection string
<u>IVHost</u>	<u>getVHost()</u> Get parent vHost
com.wowza.wms.protocol.wowz.WOWZSession	<u>getWowzSession()</u>
ClientWriteListener	<u>getWriteListener()</u> Object that tracks write operations
boolean	<u>isAcceptConnection()</u> Is auto accept connection
boolean	<u>isAcceptConnectionSendClientId()</u> Include the client id in the accept connection response
boolean	<u>isConnected()</u> Is this client connected
boolean	<u>isEncrypted()</u> Is this connection encrypted (RTMPE or RTMPTE)
boolean	<u>isFlashMediaLiveEncoder()</u> Returns true if this connection is the Flash Media Live Encoder
boolean	<u>isFlashVersion10()</u> Returns true if the Flash version is equal or greater than 10.x.x.x
boolean	<u>isFlashVersion90115()</u> Returns true if the Flash version is equal or greater than 9.0.115.x
boolean	<u>isFlashVersionH264Capable()</u> Returns true if the connected client is capable of playing H.264 video (Flash player 9.0.45.x or greater)
boolean	<u>isLiveRepeater()</u> Returns true if this connection is from the live stream repeater
boolean	<u>isObjectEncodingAMF0()</u> Is the object encoding for this client AMF0
boolean	<u>isObjectEncodingAMF3()</u> Is the object encoding for this client AMF3
boolean	<u>isSecure()</u> Is this connection protected by either SSL or encryption (RTMPE, RTMPTE, RTMPS)
boolean	<u>isSendCloseOnRejectConnection()</u>
boolean	<u>isSSL()</u> Is this connection SSL (RTMPS)
boolean	<u>isValidateFMLEConnections()</u> Returns true if validating FMLE connection (default is false)

int	ping (IModulePingResult pingResult) Ping client.
void	redirectConnection (String url) Redirection connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected and info.ex.code of 302)
void	redirectConnection (String url, String description) Redirection connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected and info.ex.code of 302)
void	redirectConnection (String url, String description, AMFData errorObj) Redirection connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected and info.ex.code of 302).
void	redirectConnection (String url, String description, String errorStr) Redirection connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected and info.ex.code of 302)
void	rejectConnection () Reject connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected)
void	rejectConnection (AMFData errorObj) Reject connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected)
void	rejectConnection (String errorStr) Reject connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected)
void	rejectConnection (String description, AMFData errorObj) Reject connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected)
void	rejectConnection (String description, String errorStr) Reject connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected)
void	reparentClient (IVHost vhost) Move a client object to a new vhost.
void	setAcceptConnection (boolean acceptConnection) Set the default for accept connection
void	setAcceptConnectionDescription (String description) Call this method from onConnect to set the info.description property returned in NetConnection onStatus handler
void	setAcceptConnectionExObj (AMFDataObj acceptConnectionExObj) Call this method from onConnect to set the info.ex property returned in NetConnection onStatus handler
void	setAcceptConnectionObj (AMFData acceptConnectionObj) Call this method from onConnect to add an "application" property to the info object that is passed back to the Flash client in the onStatus handler.
void	setAcceptConnectionSendClientId (boolean acceptConnectionSendClientId) Include the client id in the connection response

void	<u>setBufferTime</u> (int bufferTime) Set default buffer time for newly created mediaStream objects
void	<u>setFastPlaySettings</u> (<u>FastPlaySettings</u> fastPlaySettings) Set fastPlay settings
void	<u>setFlashVer</u> (String flashVer) Set client flash version
void	<u>setIdleFrequency</u> (int idleFrequency) Set client idle frequency (milliseconds)
void	<u>setLastValidateTime</u> (long lastValidateTime) Set the last time (milliseconds) the connection was validated with ping
void	<u>setLiveRepeaterCapabilities</u> (int liveRepeaterCapabilities) Set the live repeater capabilities of this connection
void	<u>setLiveStreamPacketizerList</u> (String liveStreamPacketizerList) Set the comma separated list of LiveStreamPacketizers names being used by this client (see conf/LiveStreamPacketizers.xml)
void	<u>setLiveStreamTranscoderList</u> (String liveStreamTranscoderList) Set the comma separated list of LiveStreamTranscoders names being used by this client (see conf/LiveStreamTranscoders.xml)
void	<u>setObjectEncoding</u> (int objectEncoding) Set the object encoding level
void	<u>setRepeaterOriginUrl</u> (String repeaterOriginUrl) Set the origin URL used by the Live Stream Repeater
void	<u>setSendCloseOnRejectConnection</u> (boolean sendCloseOnRejectConnection)
void	<u>setSharedObjectReadAccess</u> (String sharedObjectReadAccess) Set the shared object read access value.
void	<u>setSharedObjectWriteAccess</u> (String sharedObjectWriteAccess) Set the shared object write access value.
void	<u>setShutdownClient</u> (boolean shutdownClient) Gracefully and forcefully shutdown a client.
void	<u>setStreamAudioSampleAccess</u> (String audioSampleAccess) Set the stream audio sample access value.
void	<u>setStreamReadAccess</u> (String streamReadAccess) Set the stream object read access value.
void	<u>setStreamType</u> (String streamType) Set default streamType
void	<u>setStreamVideoSampleAccess</u> (String videoSampleAccess) Set the stream video sample access value.
void	<u>setStreamWriteAccess</u> (String streamWriteAccess) Set the stream object write access value.

void	<code>setThreadContext()</code> Set the thread logging context to this client
void	<code>setValidateFMLEConnections()</code> (boolean validateFMLEConnections) Returns true if validating FMLE connection (default is false)
void	<code>setWowzSession()</code> (com.wowza.wms.protocol.wowz.WOWZSession wowzSession)
void	<code>shutdownClient()</code> Gracefully shutdown a client.
int	<code>testFlashVersion()</code> (int[] version) Test to see if the connected client flash version is equal to or greater than a given value.
void	<code>touch()</code> Update the last touch time for client

Fields

VIDEOSAMPLE_ACCESS_ALL

```
public static final java.lang.String VIDEOSAMPLE_ACCESS_ALL
```

Constant value: *

VIDEOSAMPLE_ACCESS_NONE

```
public static final java.lang.String VIDEOSAMPLE_ACCESS_NONE
```

Constant value:

AUDIOSAMPLE_ACCESS_ALL

```
public static final java.lang.String AUDIOSAMPLE_ACCESS_ALL
```

Constant value: *

AUDIOSAMPLE_ACCESS_NONE

```
public static final java.lang.String AUDIOSAMPLE_ACCESS_NONE
```

Constant value:

READ_ACCESS_ALL

```
public static final java.lang.String READ_ACCESS_ALL
```

Constant value: *

(continued from last page)

READ_ACCESS_NONE

```
public static final java.lang.String READ_ACCESS_NONE
```

Constant value:

WRITE_ACCESS_ALL

```
public static final java.lang.String WRITE_ACCESS_ALL
```

Constant value: *

WRITE_ACCESS_NONE

```
public static final java.lang.String WRITE_ACCESS_NONE
```

Constant value:

Methods

getClientId

```
public int getClientId()
```

Get client id. Assigned by server and connection time.

Returns:

client id

getFlashVer

```
public String getFlashVer()
```

Get client flash version

Returns:

client flash version string

setFlashVer

```
public void setFlashVer(String flashVer)
```

Set client flash version

Parameters:

flashVer - client flash version string

getTotalIOPerformanceCounter

```
public IOPerformanceCounter getTotalIOPerformanceCounter()
```

Get performance counter for all bytes transferred to this client. This includes function calls.

Returns:

performance counter

getMediaIOPerformanceCounter

public [IOPerformanceCounter](#) **getMediaIOPerformanceCounter**()

Get performance counter for media bytes transferred to this client. Only includes bytes to mediaStream objects

Returns:

performance counter

getAppInstance

public [IApplicationInstance](#) **getAppInstance**()

Get parent applicationInstance. Is null if connection reject or before accepted.

Returns:

parent applicationInstance

getApplication

public [IApplication](#) **getApplication**()

Get parent application. Is null if connection reject or before accepted.

Returns:

application

getPlayStreams

public java.util.List **getPlayStreams**()

Get a collection of all play streams. Play streams are streams that are created due to a call to play.

Iterate Play Streams

```
IClient client;

List playStreams = client.getPlayStreams();
Iterator iter = playStreams.iterator();
while(iter.hasNext())
{
    IMediaStream stream = (IMediaStream)iter.next();
    WMSLoggerFactory.getLogger(null).debug("stream: "+stream.getName());
}
```

Returns:

collection of play streams

getPublishStreams

```
public java.util.List getPublishStreams()
```

Get a collection of publish streams. Publish streams are streams that are created due to a call to publish.

Iterate Public Streams

```
IClient client;

List publishStreams = client.getPublishStreams();
Iterator iter = publishStreams.iterator();
while(iter.hasNext())
{
    IMediaStream stream = (IMediaStream)iter.next();
    WMSLoggerFactory.getLogger(null).debug("stream: "+stream.getName());
}
```

Returns:

collection of publish stream

getPlayStreamCount

```
public int getPlayStreamCount()
```

Get the number of play streams

Returns:

number of play streams

getPublishStreamCount

```
public int getPublishStreamCount()
```

Get the number of publish streams

Returns:

number of publish streams

shutdownClient

```
public void shutdownClient()
```

Gracefully shutdown a client. Only use this method to shutdown a client if you know the client is connected to Wowza Pro. If the client may be disconnected from Wowza Pro, use `IClient.setShutdownClient(true)`

(continued from last page)

touch

```
public void touch()
```

Update the last touch time for client

isConnected

```
public boolean isConnected()
```

Is this client connected

Returns:

is connected

getProperties

```
public WMSProperties getProperties()
```

Get client properties

Returns:

collection of client properties

getStreamType

```
public String getStreamType()
```

Get default streamType

Returns:

streamType

setStreamType

```
public void setStreamType(String streamType)
```

Set default streamType

Parameters:

streamType - streamType

isAcceptConnection

```
public boolean isAcceptConnection()
```

Is auto accept connection

Returns:

auto accept connection

acceptConnection

```
public void acceptConnection()
```

Accept connection

(continued from last page)

acceptConnection

```
public void acceptConnection(String successStr)
```

Accept connection

Parameters:

successStr - application property value added to NetConnection.Connect.Success event object

acceptConnection

```
public void acceptConnection(AMFData successObj)
```

Accept connection

Parameters:

successObj - application property value added to NetConnection.Connect.Success event object

rejectConnection

```
public void rejectConnection()
```

Reject connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected)

rejectConnection

```
public void rejectConnection(String errorStr)
```

Reject connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected)

Parameters:

errorStr - error (returned in onStatus handler in info.application parameter)

rejectConnection

```
public void rejectConnection(AMFData errorObj)
```

Reject connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected)

Parameters:

errorObj - error AMFData (returned in onStatus handler in info.application parameter)

rejectConnection

```
public void rejectConnection(String description,  
String errorStr)
```

Reject connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected)

Parameters:

description - description (returned in onStatus handler in info.description parameter)

errorStr - error (returned in onStatus handler in info.application parameter)

rejectConnection

```
public void rejectConnection(String description,  
AMFData errorObj)
```

Reject connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected)

(continued from last page)

Parameters:

description - description (returned in onStatus handler in info.description parameter)
errorObj - error AMFData (returned in onStatus handler in info.application parameter)

redirectConnection

```
public void redirectConnection(String url)
```

Redirection connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected and info.ex.code of 302)

Parameters:

url - url (returned in onStatus handler in info.ex.redirect parameter)

redirectConnection

```
public void redirectConnection(String url,  
    String description)
```

Redirection connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected and info.ex.code of 302)

Parameters:

url - url (returned in onStatus handler in info.ex.redirect parameter)
description - description (returned in onStatus handler in info.description parameter)

redirectConnection

```
public void redirectConnection(String url,  
    String description,  
    String errorStr)
```

Redirection connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected and info.ex.code of 302)

Parameters:

url - url (returned in onStatus handler in info.ex.redirect parameter)
description - description (returned in onStatus handler in info.description parameter)
errorStr - error (returned in onStatus handler in info.application parameter)

redirectConnection

```
public void redirectConnection(String url,  
    String description,  
    AMFData errorObj)
```

(continued from last page)

Redirection connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected and info.ex.code of 302). Code must be added client side to capture the NetConnection.Connect.Rejected message, look for the info.ex.code value of 302 and attempt a reconnect to the info.ex.redirect url. Client side example:

```
public function ncOnStatus(infoObject:NetStatusEvent)
{
    if (infoObject.info.code == "NetConnection.Connect.Rejected")
    {
        if (infoObject.info.ex.code == 302)
        {
            trace("Connection redirected: "+infoObject.info.ex.redirect);
            nc.connect(infoObject.info.ex.redirect);
        }
        else
            trace("Connection rejected");
    }
}
nc.addEventListener(NetStatusEvent.NET_STATUS, ncOnStatus);
```

Parameters:

url - url (returned in onStatus handler in info.ex.redirect parameter)
description - description (returned in onStatus handler in info.description parameter)
errorObj - error AMFData (returned in onStatus handler in info.application parameter)

isAcceptConnectionSendClientId

```
public boolean isAcceptConnectionSendClientId()
```

Include the client id in the accept connection response

Returns:

true if client id is included

setAcceptConnectionSendClientId

```
public void setAcceptConnectionSendClientId(boolean acceptConnectionSendClientId)
```

Include the client id in the connection response

Parameters:

acceptConnectionSendClientId - true if client id is included

getVHost

```
public IVHost getVHost()
```

Get parent vHost

Returns:

vHost object

call

```
public void call(String handlerName,  
    IModuleCallResult resultObj,  
    Object[] params)
```

Call client method/handler. You can call any method/handler attached to NetConnection object. Such as netConnection.myFunc

Call Client Handler Method With Result Object

```
IClient client;  
  
class ReturnObj implements IModuleCallResult  
{  
    public void onResult(IClient client, RequestFunction function,  
        AMFDataList params)  
    {  
        WMSLoggerFactory.getLogger(null).debug("onResult");  
    }  
}  
  
client.call("onCustomMethod", new ReturnObj(), "param1", 1.2345, false, new Date());
```

Parameters:

handlerName - handler name

resultObj - if client method returns data this object will receive callback, null if don't care or no return on client method/handler.

params - variable list of argument. Will be automatically wrapped in AMFData objects

call

```
public void call(String handlerName)
```

Simplified call client method/handler call. No resultObj. No parameters

Parameters:

handlerName - handler name

ping

```
public int ping(IModulePingResult pingResult)
```


(continued from last page)

Ping client. Unlike FMS this ping is asynchronous. pingResult object will be notified of successful response by client or timeout.

Ping Client and Kill Connection If Client Has Died

```
class PingResult implements IModulePingResult
{
    public void onResult(IClient client, long pingTime, int pingId, boolean result)
    {
        WMSLogger log = WMSLoggerFactory.getLogger(null);
        log.debug("onResult: result:"+result);
        if (!result)
        {
            // client has died lets kill it
            client.getAppInstance().shutdownClient(client);
        }
        else
            log.debug("lastPingTime: "+client.getPingRoundTripTime());
    }
}
client.ping(new PingResult());
```

Parameters:

pingResult - ping result object. Will receive callback on result of ping or timeout. Can be null.

Returns:

ping id

getIp

```
public String getIp()
```

Client ip address

Returns:

client ip address

getFastPlaySettings

```
public FastPlaySettings getFastPlaySettings()
```

Get the current fastPlay settings. FastPlay settings when set from the client are immediately cleared on the next call to seek/unpause/play.

Returns:

fastPlay settings

(continued from last page)

setFastPlaySettings

```
public void setFastPlaySettings(FastPlaySettings fastPlaySettings)
```

Set fastPlay settings

Parameters:

fastPlaySettings - fastPlay settings

clearFastPlaySettings

```
public void clearFastPlaySettings()
```

Force clear the fastPlay settings

getRespFunctions

```
public ResponseFunctions getRespFunctions()
```

Get client responseFunctions object. The responseFunctions object is the method by which events are sent to the client. This code snippet shows how to send an onStatus event for a mediaStream (stream) to the client.

Send Status Message to Client

```
ResponseFunctions respFunctions = client.getRespFunctions();

ResponseFunction resp = new ResponseFunction();
resp.createDefaultMessage("onStatus", 0.0);
resp.setRetAMFObj(stream.getRespAMFDataObj());
resp.setSrc(stream.getSrc());

AMFDataObj data = new AMFDataObj();
data.put("level", new AMFDataItem("status"));
data.put("code", new AMFDataItem("NetStream.Publish.Success"));
data.put("description", new AMFDataItem(stream.getName()+" is now unpublished.));
data.put("clientId", new AMFDataItem(clientID));
resp.addBody(data);

respFunctions.add(resp);
```

Returns:

responseFunctions object

getConnectTime

```
public long getConnectTime()
```

(continued from last page)

Get time in milliseconds the client connected to the server. To get the duration of time in milliseconds that the client has been connected to the server us the following formula: `(System.currentTimeMillis() - client.getConnectTime())`

Returns:

duration connected (milliseconds)

getPingRoundTripTime

```
public long getPingRoundTripTime()
```

Get turn around time (milliseconds) of last ping request

Returns:

time (milliseconds)

getDateStarted

```
public String getDateStarted()
```

Get date and time of client connection

Returns:

date and time of connection

getTimeRunning

```
public String getTimeRunning()
```

Get elapsed time of connection

Returns:

elapsed time of connection

getTimeRunningSeconds

```
public double getTimeRunningSeconds()
```

Get time running in seconds

Returns:

time running in seconds

getStreamFile

```
public java.io.File getStreamFile(String streamName)
```

Get File object for stream with given name. Assumes file extension is .flv.

Parameters:

streamName - stream name

Returns:

file

getStreamFile

```
public java.io.File getStreamFile(String streamName,  
    String streamExt)
```

(continued from last page)

Get File object for stream with given name and extension. For example test.flv would be `getStreamFile("test", "flv");`

Get File Descriptor

```
IClient client;
File file = client.getStreamFile("test", "flv");
WMSLogger log = WMSLoggerFactory.getLogger(null);
if (file != null)
{
    if (file.exists())
        log.debug("getStreamFile: " + file.getPath());
}
```

Parameters:

streamName - stream name
streamExt - stream extension

Returns:

file

getStreamFile

```
public java.io.File getStreamFile(String streamName,
    String streamExt,
    boolean doCreateFolder)
```

Get File object for stream with given name and extension. If doCreateFolder is true, create folder necessary to make path exist. For example test.flv would be `getStreamFile("test", "flv");`

Parameters:

streamName - stream name
streamExt - stream extension
doCreateFolder - create folders if needed

Returns:

file

getBufferTime

```
public int getBufferTime()
```

Get default buffer time for newly created mediaStream objects

Returns:

default buffer time (milliseconds)

setBufferTime

```
public void setBufferTime(int bufferTime)
```

(continued from last page)

Set default buffer time for newly created mediaStream objects

Parameters:

bufferTime - bufer time (milliseconds)

getResponseAMFObj

```
public AMFObj getResponseAMFObj(int index)
```

Get the response channel AMFObj for channel index.

Parameters:

index - channel index

Returns:

AMFObj

getRespAMFAudioObj

```
public AMFObj getRespAMFAudioObj(IMediaStream stream)
```

Get the audio response AMFObj for a given mediaStream

Parameters:

stream - mediaStream

Returns:

response channel AMFObj

getRespAMFVideoObj

```
public AMFObj getRespAMFVideoObj(IMediaStream stream)
```

Get the video response AMFObj for a given mediaStream

Parameters:

stream - mediaStream

Returns:

response channel AMFObj

getRespAMFDataObj

```
public AMFObj getRespAMFDataObj(IMediaStream stream)
```

Get the data response AMFObj for a given mediaStream

Parameters:

stream - mediaStream

Returns:

response channel AMFObj

getQueryStr

```
public String getQueryStr()
```

(continued from last page)

Get the query string part of the connection string. If the connection string where:
rtmp://localhost/app/appInst?data1=myData1&data2=myData2, then query string would be
data1=myData1&data2=myData2

Returns:query string

getReferrer

```
public String getReferrer()
```

Get the referrer data for this connection. The referrer in Wowza Streaming Engine terms is the full url to the Flash movie that connected to the server. Checking referre upon connection is a great way to secure server connections and only allow your Flash movies to connect to the server.

Only Allow Your .swf Files to Connect to Wowza Pro server

```
public class MyModule extends ModuleBase
{
    static public void onConnect(IClient client, RequestFunction function,
        AMFDataList params)
    {
        String referrer = client.getReferrer();

        if (referrer.indexOf("www.mycompany.com") >= 0 &&
            referrer.indexOf("mycoolflash.swf") >= 0)
            client.acceptConnection("valid referrer");
        else
            client.rejectConnection("invalid referrer: "+referrer);
    }
}
```

Returns:referrer

getPageUrl

```
public String getPageUrl()
```

Get the pageUrl for this connection.

Returns:pageUrl for this connection

getUri

```
public String getUri()
```

Get the full URI of the connection string

(continued from last page)

Returns:

URI of connection string

getProtocol

```
public int getProtocol()
```

Get connection protocol (1 = RTMP, 3 = RTMPT)

Returns:

connection protocol (1 = RTMP, 3 = RTMPT)

getServerHostPort

```
public HostPort getServerHostPort()
```

Get the hostPort object for the connection that is servicing this client

Returns:

vHost hostPort servicing request

isSecure

```
public boolean isSecure()
```

Is this connection protected by either SSL or encryption (RTMPE, RTMPTE, RTMPS)

Returns:

is connection protected by either SSL or encryption (RTMPE, RTMPTE, RTMPS)

isSSL

```
public boolean isSSL()
```

Is this connection SSL (RTMPS)

Returns:

is connection (RTMPS)

isEncrypted

```
public boolean isEncrypted()
```

Is this connection encrypted (RTMPE or RTMPTE)

Returns:

is connection encrypted (RTMPE or RTMPTE)

getIdleFrequency

```
public int getIdleFrequency()
```

Get client idle frequency (milliseconds)

Returns:

client idle frequency (milliseconds)

setIdleFrequency

```
public void setIdleFrequency(int idleFrequency)
```

Set client idle frequency (milliseconds)

Parameters:

idleFrequency - client idle frequency (milliseconds)

getSharedObjectReadAccess

```
public String getSharedObjectReadAccess()
```

Get the shared object read access value. see setSharedObjectReadAccess for more information.

Returns:

shared object read access value

setSharedObjectReadAccess

```
public void setSharedObjectReadAccess(String sharedObjectReadAccess)
```

Set the shared object read access value. This list is a semi-colon delimited list of shared objects that this client has access to. By default the value is set to IClient.READ_ACCESS_ALL which indicates this client can read all shared objects. If you want to disable read access to all shared objects set this value to IClient.READ_ACCESS_NONE. If this value is set to anything other than IClient.READ_ACCESS_ALL or IClient.READ_ACCESS_NONE then each item in the semi-color delimited list is compared against incoming shared object read requests. If any item in the list completely matches the shared object name or the start of the shared object name, then access is allowed. For example, if the sharedObjectReadAccess list is set to "testa/testb;testc" then the following requests would be granted the following access:

- testc: Granted Access
- testc/test: Granted Access
- testC/test: Denied Access (incorrect case)
- testa/testb: Granted Access
- testa/testb123: Granted Access
- testa/testb/file123: Granted Access
- testa/test: Denied Access (incomplete match)

Values in this list are always case sensitive.

Parameters:

sharedObjectReadAccess - shared object read access value

getSharedObjectWriteAccess

```
public String getSharedObjectWriteAccess()
```

Get the shared object write access value. see setSharedObjectWriteAccess for more information.

Returns:

shared object write access value

setSharedObjectWriteAccess

```
public void setSharedObjectWriteAccess(String sharedObjectWriteAccess)
```


(continued from last page)

Set the shared object write access value. This list is a semi-colon delimited list of shared objects that this client has access to. By default the value is set to IClient.WRITE_ACCESS_ALL which indicates this client can write all shared objects. If you want to disable write access to all shared objects set this value to IClient.WRITE_ACCESS_NONE. If this value is set to anything other than IClient.WRITE_ACCESS_ALL or IClient.WRITE_ACCESS_NONE then each item in the semi-color delimited list is compared against incoming shared object write requests. If any item in the list completely matches the shared object name or the start of the shared object name, then access is allowed. For example, if the sharedObjectWriteAccess list is set to "testa/testb;testc" then the following requests would be granted the following access:

- testc: Granted Access
- testc/test: Granted Access
- testC/test: Denied Access (incorrect case)
- testa/testb: Granted Access
- testa/testb123: Granted Access
- testa/testb/file123: Granted Access
- testa/test: Denied Access (incomplete match)

Values in this list are always case sensitive.

Parameters:

sharedObjectWriteAccess

getStreamVideoSampleAccess

```
public String getStreamVideoSampleAccess()
```

Get the video sample access value. see setVideoStreamAccess for more information.

Returns:

video sample access

setStreamVideoSampleAccess

```
public void setStreamVideoSampleAccess(String videoSampleAccess)
```

Set the stream video sample access value. This list is a semi-colon delimited list of stream names that this client has access to. By default the value is set to IClient.VIDEOSTREAM_ACCESS_NONE which indicates this client can access no stream data client side. If you want to enable access to all stream names set this value to IClient.VIDEOSTREAM_ACCESS_ALL. If this value is set to anything other than IClient.VIDEOSTREAM_ACCESS_ALL or IClient.VIDEOSTREAM_ACCESS_NONE then each item in the semi-color delimited list is compared against incoming stream play requests. If any item in the list completely matches the stream name or the start of the stream name, then access is allowed. For example, if the streamVideoSampleAccess list is set to "testa/testb;testc" then the following play requests would be granted the following access:

- testc: Granted Access
- testc/test: Granted Access
- testC/test: Denied Access (incorrect case)
- testa/testb: Granted Access
- testa/testb123: Granted Access
- testa/testb/file123: Granted Access
- testa/test: Denied Access (incomplete match)

Values in this list are always case sensitive.

Parameters:

videoSampleAccess

getStreamAudioSampleAccess

```
public String getStreamAudioSampleAccess()
```

Get the audio sample access value. see setVideoStreamAccess for more information.

Returns:

(continued from last page)

audio sample access

setStreamAudioSampleAccess

```
public void setStreamAudioSampleAccess(String audioSampleAccess)
```

Set the stream audio sample access value. This list is a semi-colon delimited list of stream names that this client has access to. By default the value is set to IClient.AUDIOSTREAM_ACCESS_NONE which indicates this client can access no stream data client side. If you want to enable access to all stream names set this value to IClient.AUDIOSTREAM_ACCESS_ALL. If this value is set to anything other than IClient.AUDIOSTREAM_ACCESS_ALL or IClient.AUDIOSTREAM_ACCESS_NONE then each item in the semi-color delimited list is compared against incoming stream play requests. If any item in the list completely matches the stream name or the start of the stream name, then access is allowed. For example, if the streamAudioSampleAccess list is set to "testa/testb;testc" then the following play requests would be granted the following access:

- testc: Granted Access
- testc/test: Granted Access
- testC/test: Denied Access (incorrect case)
- testa/testb: Granted Access
- testa/testb123: Granted Access
- testa/testb/file123: Granted Access
- testa/test: Denied Access (incomplete match)

Values in this list are always case sensitive.

Parameters:

audioSampleAccess

getStreamReadAccess

```
public String getStreamReadAccess()
```

Get the stream read access value. see setStreamReadAccess for more information.

Returns:

stream read access value

setStreamReadAccess

```
public void setStreamReadAccess(String streamReadAccess)
```

Set the stream object read access value. This list is a semi-colon delimited list of stream names that this client has access to. By default the value is set to IClient.READ_ACCESS_ALL which indicates this client can play all streams. If you want to disable read access to all stream names set this value to IClient.READ_ACCESS_NONE. If this value is set to anything other than IClient.READ_ACCESS_ALL or IClient.READ_ACCESS_NONE then each item in the semi-color delimited list is compared against incoming stream play requests. If any item in the list completely matches the stream name or the start of the stream name, then access is allowed. For example, if the streamReadAccess list is set to "testa/testb;testc" then the following play requests would be granted the following access:

- testc: Granted Access
- testc/test: Granted Access
- testC/test: Denied Access (incorrect case)
- testa/testb: Granted Access
- testa/testb123: Granted Access
- testa/testb/file123: Granted Access
- testa/test: Denied Access (incomplete match)

Values in this list are always case sensitive.

Parameters:

streamReadAccess - shared object read access value

getStreamWriteAccess

```
public String getStreamWriteAccess()
```

Get the stream write access value. see setStreamWriteAccess for more information.

Returns:

stream write access value

setStreamWriteAccess

```
public void setStreamWriteAccess(String streamWriteAccess)
```

Set the stream object write access value. This list is a semi-colon delimited list of stream names that this client has access to. By default the value is set to IClient.WRITE_ACCESS_ALL which indicates this client can play all streams. If you want to disable write access to all stream names set this value to IClient.WRITE_ACCESS_NONE. If this value is set to anything other than IClient.WRITE_ACCESS_ALL or IClient.WRITE_ACCESS_NONE then each item in the semi-color delimited list is compared against incoming stream play requests. If any item in the list completely matches the stream name or the start of the stream name, then access is allowed. For example, if the streamWriteAccess list is set to "testa/testb;testc" then the following play requests would be granted the following access:

- testc: Granted Access
- testc/test: Granted Access
- testC/test: Denied Access (incorrect case)
- testa/testb: Granted Access
- testa/testb123: Granted Access
- testa/testb/file123: Granted Access
- testa/test: Denied Access (incomplete match)

Values in this list are always case sensitive.

Parameters:

streamWriteAccess - shared object read access value

getWriteListener

```
public ClientWriteListener getWriteListener()
```

Object that tracks write operations

Returns:

Object that tracks write operations

addAcceptConnectionAttribute

```
public void addAcceptConnectionAttribute(String key,  
    AMFDataObj item)
```

Add and attribute to the resultObj that gets passed back to the client on successful connection

Parameters:

key - key
item - item to add

addAcceptConnectionAttribute

```
public void addAcceptConnectionAttribute(String key,  
    String item)
```

Add and attribute to the resultObj that gets passed back to the client on successful connection

(continued from last page)

Parameters:

key - key
item - item to add

getRepeaterOriginUrl

```
public String getRepeaterOriginUrl()
```

Get the origin URL used by the Live Stream Repeater

Returns:

URL used by the Live Stream Repeater

setRepeaterOriginUrl

```
public void setRepeaterOriginUrl(String repeaterOriginUrl)
```

Set the origin URL used by the Live Stream Repeater

Parameters:

repeaterOriginUrl - URL used by the Live Stream Repeater

getLastValidateTime

```
public long getLastValidateTime()
```

Get last time (millisecond) the connection was validated with a ping

Returns:

last time (millisecond) the connection was validated with a ping

setLastValidateTime

```
public void setLastValidateTime(long lastValidateTime)
```

Set the last time (milliseconds) the connection was validated with ping

Parameters:

lastValidateTime - last time (milliseconds) the connection was validated with ping

getPingTimeout

```
public int getPingTimeout()
```

Get the ping timeout (milliseconds)

Returns:

ping timeout (milliseconds)

isLiveRepeater

```
public boolean isLiveRepeater()
```

Returns true if this connection is from the live stream repeater

Returns:

true if this connection is from the live stream repeater

isFlashVersionH264Capable

```
public boolean isFlashVersionH264Capable()
```

Returns true if the connected client is capable of playing H.264 video (Flash player 9.0.45.x or greater)

Returns:

returns true if the connected client is capable of playing H.264 video (Flash player 9.0.45.x or greater)

isFlashVersion90115

```
public boolean isFlashVersion90115()
```

Returns true if the Flash version is equal or greater than 9.0.115.x

Returns:

returns true if the Flash version is equal or greater than 9.0.115.x

isFlashVersion10

```
public boolean isFlashVersion10()
```

Returns true if the Flash version is equal or greater than 10.x.x.x

Returns:

returns true if the Flash version is equal or greater than 10.x.x.x

isFlashMediaLiveEncoder

```
public boolean isFlashMediaLiveEncoder()
```

Returns true if this connection is the Flash Media Live Encoder

Returns:

true if this connection is the Flash Media Live Encoder

testFlashVersion

```
public int testFlashVersion(int[] version)
```

Test to see if the connected client flash version is equal to or greater than a given value. Example: int isGood = testFlashVersion({ 9, 0, 5, 12});

Parameters:

version - array of version values

Returns:

1 if greater, 0 if equal, -1 if less than

isObjectEncodingAMF3

```
public boolean isObjectEncodingAMF3()
```

Is the object encoding for this client AMF3

Returns:

true of the object encoding for this client is AMF3

isObjectEncodingAMF0

```
public boolean isObjectEncodingAMF0()
```

Is the object encoding for this client AMF0

Returns:

true of the object encoding for this client is AMF0

setObjectEncoding

```
public void setObjectEncoding(int objectEncoding)
```

Set the object encoding level

Parameters:

objectEncoding - object encoding level

getObjectEncoding

```
public int getObjectEncoding()
```

Get the object encoding level

Returns:

object encoding level

setAcceptConnectionObj

```
public void setAcceptConnectionObj(AMFData acceptConnectionObj)
```

Call this method from onConnect to add an "application" property to the info object that is passed back to the Flash client in the onStatus handler.

Parameters:

acceptConnectionObj - connection object

setAcceptConnectionDescription

```
public void setAcceptConnectionDescription(String description)
```

Call this method from onConnect to set the info.description property returned in NetConnection onStatus handler

Parameters:

description - description

setAcceptConnectionExObj

```
public void setAcceptConnectionExObj(AMFDataObj acceptConnectionExObj)
```

Call this method from onConnect to set the info.ex property returned in NetConnection onStatus handler

Parameters:

acceptConnectionExObj - exObj AMFData item return in info.ex property of NetConnection onStatus handler

(continued from last page)

getRTPStream

```
public RTPStream getRTPStream( )
```

If this client was created due to an RTP connection to the server return the underlying RTPStream object

Returns:

underlying RTPStream object

setAcceptConnection

```
public void setAcceptConnection(boolean acceptConnection)
```

Set the default for accept connection

Parameters:

acceptConnection - default for accept connection

setShutdownClient

```
public void setShutdownClient(boolean shutdownClient)
```

Gracefully and forcefully shutdown a client.

Parameters:

shutdownClient - set to true to gracefully and forcefully shutdown a client

reparentClient

```
public void reparentClient(IVHost vhost)
```

Move a client object to a new vhost. This can only be done right after the handshake process has completed. See IVHostNotify.onVHostClientConnect.

Parameters:

vhost - new vhost

getMaximumSetBufferTime

```
public int getMaximumSetBufferTime( )
```

Get maximum number of milliseconds allowed for the NetStream.setBufferTime(secs) call. If set to zero this feature is turned off.

Returns:

maximum number of milliseconds allowed for the NetStream.setBufferTime(secs) call

getMaximumPendingWriteBytes

```
public int getMaximumPendingWriteBytes( )
```

Get maximum number a bytes a client connection can have waiting to be sent before the connection is terminated. If set to zero this feature is turned off.

Returns:

maximum number a bytes a client connection can have waiting to be sent before the connection is terminated

(continued from last page)

getLiveStreamPacketizerList

```
public String getLiveStreamPacketizerList()
```

Get the comma separated list of LiveStreamPacketizers names being used by this client (see conf/LiveStreamPacketizers.xml)

Returns:

comma separated list of LiveStreamPacketizers names

setLiveStreamPacketizerList

```
public void setLiveStreamPacketizerList(String liveStreamPacketizerList)
```

Set the comma separated list of LiveStreamPacketizers names being used by this client (see conf/LiveStreamPacketizers.xml)

Parameters:

liveStreamPacketizerList - comma separated list of LiveStreamPacketizers names

getLiveStreamTranscoderList

```
public String getLiveStreamTranscoderList()
```

Get the comma separated list of LiveStreamTranscoders names being used by this client (see conf/LiveStreamTranscoders.xml)

Returns:

comma separated list of LiveStreamTranscoders names

setLiveStreamTranscoderList

```
public void setLiveStreamTranscoderList(String liveStreamTranscoderList)
```

Set the comma separated list of LiveStreamTranscoders names being used by this client (see conf/LiveStreamTranscoders.xml)

Parameters:

liveStreamTranscoderList - comma separated list of LiveStreamTranscoders names

getElapsedTime

```
public ElapsedTimer getElapsedTime()
```

Get elapsed time client has been connected.

Returns:

elapsed time client has been connected.

getLiveRepeaterCapabilities

```
public int getLiveRepeaterCapabilities()
```

Get the live repeater capabilities of this connection

Returns:

live repeater capabilities of this connection

setLiveRepeaterCapabilities

```
public void setLiveRepeaterCapabilities(int liveRepeaterCapabilities)
```


(continued from last page)

Set the live repeater capabilities of this connection

Parameters:

liveRepeaterCapabilities - live repeater capabilities of this connection

fcSubscribe

```
public void fcSubscribe(String streamName)
```

Subscribe to a live stream (for live stream repeater to start start from edge to origin)

Parameters:

streamName - stream name

fcSubscribe

```
public void fcSubscribe(String streamName,  
    String mediaCasterType)
```

Subscribe to a live stream (use a particular mediaCasterType)

Parameters:

streamName - stream name

mediaCasterType - media caster type name

fcUnSubscribeAll

```
public void fcUnSubscribeAll()
```

UnSubscribe to all streams that this client is current subscribed to

fcUnSubscribe

```
public void fcUnSubscribe(String streamName)
```

UnSubscribe from a stream

Parameters:

streamName - stream name

isValidFMLEConnections

```
public boolean isValidFMLEConnections()
```

Returns true if validating FMLE connection (default is false)

Returns:

true if validating FMLE connection

setValidateFMLEConnections

```
public void setValidateFMLEConnections(boolean validateFMLEConnections)
```

Returns true if validating FMLE connection (default is false)

Parameters:

validateFMLEConnections - true if validating FMLE connection

(continued from last page)

setThreadContext

```
public void setThreadContext()
```

Set the thread logging context to this client

getWowzSession

```
public com.wowza.wms.protocol.wowz.WOWZSession getWowzSession()
```

setWowzSession

```
public void setWowzSession(com.wowza.wms.protocol.wowz.WOWZSession wowzSession)
```

isSendCloseOnRejectConnection

```
public boolean isSendCloseOnRejectConnection()
```

setSendCloseOnRejectConnection

```
public void setSendCloseOnRejectConnection(boolean sendCloseOnRejectConnection)
```

com.wowza.wms.client Interface IClientNotify

public interface **IClientNotify**
extends

IClientNotify: listener interface used by [IApplicationInstance.addClientListener\(IClientNotify\)](#)

See Also:

[IApplicationInstance.addClientListener\(IClientNotify\)](#)

Method Summary

void	onClientAccept (IClient client) Triggered when client connection accepted
void	onClientConnect (IClient client) Triggered when client attempt connection
void	onClientDisconnect (IClient client) Triggered when client disconnected
void	onClientReject (IClient client) Triggered when client connection rejected

Methods

onClientConnect

public void **onClientConnect**([IClient](#) client)

Triggered when client attempt connection

Parameters:

client - client

onClientDisconnect

public void **onClientDisconnect**([IClient](#) client)

Triggered when client disconnected

Parameters:

client - client

onClientAccept

public void **onClientAccept**([IClient](#) client)

Triggered when client connection accepted

Parameters:

client - client

onClientReject

```
public void onClientReject(IClient client)
```

Triggered when client connection rejected

Parameters:

client - client

com.wowza.wms.client Interface IConnectionNotify

public interface **IConnectionNotify**
extends

IConnectionNotify: internal class to manage connection counting

Method Summary

void	onAcceptConnection (ConnectionCounter connectionCounter, ConnectionHolder connectionHolder, java.util.Date date, long stamp) Triggered when client connection accepted
void	onDisconnect (ConnectionCounter connectionCounter, ConnectionHolder connectionHolder, boolean isValid, java.util.Date date, long stamp) Triggered when client disconnected
void	onRejectConnection (ConnectionCounter connectionCounter, ConnectionHolder connectionHolder, int reason, java.util.Date date, long stamp) Triggered when client connection rejected

Methods

onAcceptConnection

```
public void onAcceptConnection(ConnectionCounter connectionCounter,  
    ConnectionHolder connectionHolder,  
    java.util.Date date,  
    long stamp)
```

Triggered when client connection accepted

Parameters:

connectionCounter
connectionHolder
date
stamp

onRejectConnection

```
public void onRejectConnection(ConnectionCounter connectionCounter,  
    ConnectionHolder connectionHolder,  
    int reason,  
    java.util.Date date,  
    long stamp)
```

Triggered when client connection rejected

Parameters:

connectionCounter
connectionHolder
reason

(continued from last page)

date
stamp

onDisconnect

```
public void onDisconnect(ConnectionCounter connectionCounter,  
    ConnectionHolder connectionHolder,  
    boolean isValid,  
    java.util.Date date,  
    long stamp)
```

Triggered when client disconnected

Parameters:

connectionCounter
connectionHolder
isValid
date
stamp

Package

com.wowza.wms.closedcaption.cea608

com.wowza.wms.closedcaption.cea608

Class ClosedCaptionCEA608Utils

java.lang.Object

└-com.wowza.wms.timedtext.cea608.CEA608Constants

└-[com.wowza.wms.timedtext.cea608.ClosedCaptionCEA608Utils](#)

└-com.wowza.wms.closedcaption.cea608.ClosedCaptionCEA608Utils

public class **ClosedCaptionCEA608Utils**

extends [ClosedCaptionCEA608Utils](#)

Fields inherited from class com.wowza.wms.timedtext.cea608.ClosedCaptionCEA608Utils

[CEA608_EXTENDED_FR_SP_MAP](#), [CEA608_EXTENDED_NA_MAP](#),
[CEA608_EXTENDED_PORT_GERMAN_DANISH_MAP](#), [CEA608_MUSICAL_NOTE](#), [CEA608_SOLID_BLOCK](#), [CHANNEL1](#),
[CHANNEL2](#), [COL_0](#), [COL_12](#), [COL_16](#), [COL_20](#), [COL_24](#), [COL_28](#), [COL_4](#), [COL_5](#), [COL0_BLUE](#), [COL0_CYAN](#),
[COL0_GREEN](#), [COL0_HIBYTE](#), [COL0_LOWBYTE](#), [COL0_MAGENTA](#), [COL0_RED](#), [COL0_WHITE](#), [COL0_YELLOW](#),
[COLN_LOWBYTE](#), [CONTROLCODES_AOF](#), [CONTROLCODES_AON](#), [CONTROLCODES_BAO](#), [CONTROLCODES_BAS](#),
[CONTROLCODES_BBO](#), [CONTROLCODES_BBS](#), [CONTROLCODES_BCO](#), [CONTROLCODES_BCS](#), [CONTROLCODES_BGO](#),
[CONTROLCODES_BGS](#), [CONTROLCODES_BMO](#), [CONTROLCODES_BMS](#), [CONTROLCODES BRO](#), [CONTROLCODES_BRS](#),
[CONTROLCODES_BS](#), [CONTROLCODES_BT](#), [CONTROLCODES_BWO](#), [CONTROLCODES_BWS](#), [CONTROLCODES_BYO](#),
[CONTROLCODES_BYS](#), [CONTROLCODES_CR](#), [CONTROLCODES_DER](#), [CONTROLCODES_EDM](#), [CONTROLCODES_ENM](#),
[CONTROLCODES_EOC](#), [CONTROLCODES_FA](#), [CONTROLCODES_FAU](#), [CONTROLCODES_FON](#), [CONTROLCODES_RCL](#),
[CONTROLCODES_RDC](#), [CONTROLCODES_RTD](#), [CONTROLCODES_RU2](#), [CONTROLCODES_RU3](#), [CONTROLCODES_RU4](#),
[CONTROLCODES_TO1](#), [CONTROLCODES_TO2](#), [CONTROLCODES_TO3](#), [CONTROLCODES_TR](#), [COUNT_COLS](#),
[COUNT_ROWS](#), [DONOTHING_BYTE](#), [DONOTHING_SHORT](#), [FORMAT_COLOR](#), [FORMAT_FLASH](#), [FORMAT_ITALICS](#),
[FORMAT_ITALICS_UNDERLINE](#), [LANG_ASCII](#), [LANG_ASCII_DBL](#), [LANG_GB231280](#), [LANG_KSC56011987](#),
[LANG_PRIV1](#), [LANG_PRIV2](#), [LANG_REG1](#), [MAX_CC_COUNT](#), [ODDPARITY](#), [SEI_PAYLOADTYPE](#), [SEI_STARTCODE](#),
[SEI_USERDATATYPECODE](#), [UTF8_BRITISH_POUND](#), [UTF8_BULLET](#), [UTF8_CENTS](#),
[UTF8_CLOSING_DOUBLE_QUOTE](#), [UTF8_CLOSING_GUILLEMETS](#), [UTF8_CONNECTING_VERTICAL_BAR](#),
[UTF8_COPYRIGHT](#), [UTF8_CURRENCY](#), [UTF8_DASH](#), [UTF8_DEGREE](#), [UTF8_DOUBLEQUOTE](#), [UTF8_EM_DASH](#),
[UTF8_EXCLAMATION](#), [UTF8_GREATERTHAN](#), [UTF8_INVERTED_EXCLAMATION](#), [UTF8_LEFT_SINGLE_QUOTATION](#),
[UTF8_LEFTPAREN](#), [UTF8_LESSTHAN](#), [UTF8_LOWER_A](#), [UTF8_LOWER_A_WITH_CIRCUMFLEX](#),
[UTF8_LOWER_A_WITH_DIAERESIS](#), [UTF8_LOWER_A_WITH_GRAVE](#), [UTF8_LOWER_A_WITH_RING](#),
[UTF8_LOWER_A_WITH_TILDE](#), [UTF8_LOWER_B](#), [UTF8_LOWER_C](#), [UTF8_LOWER_D](#), [UTF8_LOWER_E](#),
[UTF8_LOWER_E_WITH_CIRCUMFLEX](#), [UTF8_LOWER_E_WITH_DIAERESIS](#), [UTF8_LOWER_E_WITH_GRAVE](#),
[UTF8_LOWER_F](#), [UTF8_LOWER_G](#), [UTF8_LOWER_H](#), [UTF8_LOWER_I](#), [UTF8_LOWER_I_WITH_CIRCUMFLEX](#),
[UTF8_LOWER_I_WITH_DIAERESIS](#), [UTF8_LOWER_I_WITH_GRAVE](#), [UTF8_LOWER_J](#), [UTF8_LOWER_K](#),
[UTF8_LOWER_L](#), [UTF8_LOWER_M](#), [UTF8_LOWER_N](#), [UTF8_LOWER_O](#), [UTF8_LOWER_O_WITH_CIRCUMFLEX](#),
[UTF8_LOWER_O_WITH_DIAERESIS](#), [UTF8_LOWER_O_WITH_GRAVE](#), [UTF8_LOWER_O_WITH_STROKE](#),
[UTF8_LOWER_O_WITH_TILDE](#), [UTF8_LOWER_P](#), [UTF8_LOWER_Q](#), [UTF8_LOWER_R](#), [UTF8_LOWER_S](#),
[UTF8_LOWER_T](#), [UTF8_LOWER_U](#), [UTF8_LOWER_U_WITH_CIRCUMFLEX](#), [UTF8_LOWER_U_WITH_DIAERESIS](#),
[UTF8_LOWER_U_WITH_GRAVE](#), [UTF8_LOWER_V](#), [UTF8_LOWER_W](#), [UTF8_LOWER_X](#), [UTF8_LOWER_Y](#),
[UTF8_LOWER_Z](#), [UTF8_MINUSHYPHEN](#), [UTF8_MUSICAL_NOTE](#), [UTF8_NBSP](#), [UTF8_NOTMATCH](#), [UTF8_ONE_HALF](#),
[UTF8_OPENING_DOUBLE_QUOTE](#), [UTF8_OPENING_GUILLEMETS](#), [UTF8_PERIOD](#), [UTF8_PIPE](#), [UTF8_QUESTION](#),
[UTF8_REGISTERED_MARK](#), [UTF8_REVERSE_QUERY](#), [UTF8_RIGHTPAREN](#), [UTF8_SERVICE_MARK](#),
[UTF8_SINGLEQUOTE](#), [UTF8_SPACE](#), [UTF8_TRADEMARK](#), [UTF8_UNDERSCORE](#), [UTF8_UPPER_A](#),
[UTF8_UPPER_A_WITH_ACUTE](#), [UTF8_UPPER_A_WITH_CURCUMFLEX](#), [UTF8_UPPER_A_WITH_DIAERESIS](#),
[UTF8_UPPER_A_WITH_GRAVE](#), [UTF8_UPPER_A_WITH_RING](#), [UTF8_UPPER_A_WITH_TILDE](#), [UTF8_UPPER_B](#),
[UTF8_UPPER_C](#), [UTF8_UPPER_C_WITH_CEDILLA](#), [UTF8_UPPER_D](#), [UTF8_UPPER_E](#),
[UTF8_UPPER_E_WITH_ACUTE](#), [UTF8_UPPER_E_WITH_CIRCUMFLEX](#), [UTF8_UPPER_E_WITH_DIAERESIS](#),
[UTF8_UPPER_E_WITH_GRAVE](#), [UTF8_UPPER_F](#), [UTF8_UPPER_G](#), [UTF8_UPPER_H](#), [UTF8_UPPER_I](#),
[UTF8_UPPER_I_WITH_CIRCUMFLEX](#), [UTF8_UPPER_I_WITH_DIAERESIS](#), [UTF8_UPPER_I_WITH_GRAVE](#),
[UTF8_UPPER_I_WITH_TILDE](#), [UTF8_UPPER_J](#), [UTF8_UPPER_K](#), [UTF8_UPPER_L](#), [UTF8_UPPER_M](#),
[UTF8_UPPER_N](#), [UTF8_UPPER_O](#), [UTF8_UPPER_O_WITH_ACUTE](#), [UTF8_UPPER_O_WITH_CIRCUMFLEX](#),
[UTF8_UPPER_O_WITH_DIAERESIS](#), [UTF8_UPPER_O_WITH_GRAVE](#), [UTF8_UPPER_O_WITH_STROKE](#),
[UTF8_UPPER_O_WITH_TILDE](#), [UTF8_UPPER_P](#), [UTF8_UPPER_Q](#), [UTF8_UPPER_R](#), [UTF8_UPPER_S](#),
[UTF8_UPPER_T](#), [UTF8_UPPER_U](#), [UTF8_UPPER_U_WITH_ACUTE](#), [UTF8_UPPER_U_WITH_CIRCUMFLEX](#),
[UTF8_UPPER_U_WITH_DIAERESIS](#), [UTF8_UPPER_U_WITH_GRAVE](#), [UTF8_UPPER_V](#), [UTF8_UPPER_W](#),
[UTF8_UPPER_X](#), [UTF8_UPPER_Y](#), [UTF8_UPPER_Z](#), [UTF8_YEN](#), [UTF8MAP](#), [UTF8MAP_INDEX_OFFSET](#)

Fields inherited from class `com.wowza.wms.timedtext.cea608.CEA608Constants`

[CC_CHANNEL_CC1](#), [CC_CHANNEL_CC2](#), [CC_CHANNEL_CC3](#), [CC_CHANNEL_CC4](#), [CC_TYPE_DTVCC_PACKET_DATA](#),
[CC_TYPE_DTVCC_PACKET_START](#), [CC_TYPE_NTSC_FIELD_1](#), [CC_TYPE_NTSC_FIELD_2](#)

Constructor Summary

public [ClosedCaptionCEA608Utils\(\)](#)

Methods inherited from class [com.wowza.wms.timedtext.cea608.ClosedCaptionCEA608Utils](#)

[calcParity](#), [calcParity](#), [cea608CharToCharUTF8](#), [cea608ExtendedCharacterToUTF8](#),
[charUTF8ToCAE608](#), [charUTF8ToCAE608](#), [checkParity](#), [checkParity](#)

Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`,
`wait`

Constructors

ClosedCaptionCEA608Utils

```
public ClosedCaptionCEA608Utils()
```

Package

com.wowza.wms.drm.cenc

com.wowza.wms.drm.cenc Class CencInfo

java.lang.Object

└─com.wowza.wms.drm.cenc.CencInfo

All Implemented Interfaces:

IEncryptionInfo

```
public class CencInfo
  extends Object
  implements IEncryptionInfo
```

Field Summary

public static final	ALGORITHMID_CBC Value: 2
public static final	ALGORITHMID_CTR Value: 1
public static final	ALGORITHMID_NONE Value: 0
public static final	KEY_ROTATION_TYPE_EXTERNAL Value: 3
public static final	KEY_ROTATION_TYPE_INTERNAL_RANDOM Value: 2
public static final	KEY_ROTATION_TYPE_INTERNAL_STATIC Value: 1
public static final	KEY_ROTATION_TYPE_NONE Value: 0

Constructor Summary

public	CencInfo()
--------	----------------------------

Method Summary

void	addDRM (String hashKey, ICencDRMInfo drm) Adds a DRM system info instance to the list of DRM systems associated with the stream encryption.
static CencInfo	deserialize (byte[] data)

static String	formatKID(String KID)
String	generateRandomKID()
int	getAlgorithm()
String	getAlgorithmString()
String	getCencMp4ProtSchemeString()
String	getCencMpegTsProtSchemeString()
java.util.Map	getDRMs()
byte[]	getEncIVBytes()
byte[]	getEncKeyBytes()
java.util.List	getEncKeyBytesList()
String	getEncKeyString()
byte[]	getEncRepeaterBytes()
boolean	getIsRotatingKeys()
int	getKeyRotationType()
String	getKID()
byte[]	getKIDBytes()
byte[]	getKIDBytesSwapped()
java.util.List	getKIDsBytesList()
java.util.List	getKIDsBytesListSwapped()
java.util.List	getKIDsList()
String	getNameSpaceString()
int	getSerializeSize()
boolean	isValid()

static boolean	loadFromKeyFile (IApplicationInstance appInstance, String streamName, CencInfo cencInfo, long chunkId, int chunkDurationTarget, String keyFileParamPrefix)
byte[]	serialize ()
void	setAlgorithm (int algorithm) Sets the encryption algorithm to be used.
void	setAppInstance (IApplicationInstance appInstance)
void	setEncKeyBytes (byte[] keyBytes) Sets the encryption content key as a byte buffer.
void	setEncKeyBytesList (java.util.List keyBytesList)
void	setEncRepeaterBytes (byte[] encRepeaterBytes)
void	setIVBytes (byte[] encIVBytes)
void	setKeyRotationInterval (int interval)
void	setKeyRotationType (int keyRotationType)
void	setKID (byte[] KIDbytes) Sets the encryption key id.
void	setKID (String KID) Sets the encryption key id.
void	setKIDsList (java.util.List KIDList) Sets the list of encryption key id strings.
static byte[]	swapKIDBytes (byte[] KID)
String	toString ()
void	updateKey (long chunkId, int chunkDurationTarget)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface com.wowza.wms.httpstreamer.util.IEncryptionInfo

getEncIVBytes, getEncKeyBytes

Fields

(continued from last page)

ALGORITHMID_NONE

```
public static final int ALGORITHMID_NONE
```

Constant value: **0**

ALGORITHMID_CTR

```
public static final int ALGORITHMID_CTR
```

Constant value: **1**

ALGORITHMID_CBC

```
public static final int ALGORITHMID_CBC
```

Constant value: **2**

KEY_ROTATION_TYPE_NONE

```
public static final int KEY_ROTATION_TYPE_NONE
```

Constant value: **0**

KEY_ROTATION_TYPE_INTERNAL_STATIC

```
public static final int KEY_ROTATION_TYPE_INTERNAL_STATIC
```

Constant value: **1**

KEY_ROTATION_TYPE_INTERNAL_RANDOM

```
public static final int KEY_ROTATION_TYPE_INTERNAL_RANDOM
```

Constant value: **2**

KEY_ROTATION_TYPE_EXTERNAL

```
public static final int KEY_ROTATION_TYPE_EXTERNAL
```

Constant value: **3**

Constructors

CencInfo

```
public CencInfo()
```

Methods

(continued from last page)

setEncKeyBytes

```
public void setEncKeyBytes(byte[] keyBytes)
```

Sets the encryption content key as a byte buffer.

Parameters:

keyBytes - the encryption content key as a byte buffer

setEncKeyBytesList

```
public void setEncKeyBytesList(java.util.List keyBytesList)
```

setKeyRotationType

```
public void setKeyRotationType(int keyRotationType)
```

getKeyRotationType

```
public int getKeyRotationType()
```

getIsRotatingKeys

```
public boolean getIsRotatingKeys()
```

generateRandomKID

```
public String generateRandomKID()
```

updateKey

```
public void updateKey(long chunkId,  
    int chunkDurationTarget)
```

setKeyRotationInterval

```
public void setKeyRotationInterval(int interval)
```

getEncKeyBytes

```
public byte[] getEncKeyBytes()
```

getEncKeyBytesList

```
public java.util.List getEncKeyBytesList()
```

getEncKeyString

```
public String getEncKeyString()
```

setIVBytes

```
public void setIVBytes(byte[] encIVBytes)
```

getEncIVBytes

```
public byte[] getEncIVBytes()
```

swapKIDBytes

```
public static byte[] swapKIDBytes(byte[] KID)
```

setKID

```
public void setKID(String KID)
```

Sets the encryption key id. String must be in uuid form "F6005DCF-7F93-4B8E-85C7-F977740DA059" (big endian)

Parameters:

KID - key id

setKID

```
public void setKID(byte[] KIDbytes)
```

Sets the encryption key id.

Parameters:

KID - key id

setKIDsList

```
public void setKIDsList(java.util.List KIDList)
```

Sets the list of encryption key id strings. Ids must be in uuid form "F6005DCF-7F93-4B8E-85C7-F977740DA059" (big endian)

Parameters:

KID - key id

getKID

```
public String getKID()
```

getKIDBytes

```
public byte[] getKIDBytes()
```

getKIDBytesSwapped

```
public byte[] getKIDBytesSwapped()
```

getKIDsList

```
public java.util.List getKIDsList()
```

getKIDsBytesList

```
public java.util.List getKIDsBytesList()
```

getKIDsBytesListSwapped

```
public java.util.List getKIDsBytesListSwapped()
```

getAlgorithm

```
public int getAlgorithm()
```

getAlgorithmString

```
public String getAlgorithmString()
```

setAlgorithm

```
public void setAlgorithm(int algorithm)
```

Sets the encryption algorithm to be used. Currently only CencInfo.ALGORITHMID_CTR is supported.

Parameters:

algorithm - encryption algorithm

isValid

```
public boolean isValid()
```

getNameSpaceString

```
public String getNameSpaceString()
```

getCencMp4ProtSchemeString

```
public String getCencMp4ProtSchemeString()
```

getCencMpegTsProtSchemeString

```
public String getCencMpegTsProtSchemeString()
```

addDRM

```
public void addDRM(String hashKey,  
    ICencDRMInfo drm)
```

Adds a DRM system info instance to the list of DRM systems associated with the stream encryption. The hashKey should be some unique string to identify the DRM System, such as a concatenation of the DRM system's system name and systemId

Example hashKey: "myDRM:12345678-1234-1234-1234-123456789123"

Parameters:

hashKey - hash key

drm - the drm info instance

getDRMs

```
public java.util.Map getDRMs()
```

getEncRepeaterBytes

```
public byte[] getEncRepeaterBytes()
```

setEncRepeaterBytes

```
public void setEncRepeaterBytes(byte[] encRepeaterBytes)
```

setAppInstance

```
public void setAppInstance(IApplicationInstance appInstance)
```

formatKID

```
public static String formatKID(String KID)
```

loadFromKeyFile

```
public static boolean loadFromKeyFile(IApplicationInstance appInstance,  
    String streamName,  
    CencInfo cencInfo,  
    long chunkId,  
    int chunkDurationTarget,  
    String keyFileParamPrefix)
```

getSerializeSize

```
public int getSerializeSize()
```

serialize

```
public byte[] serialize()
```

deserialize

```
public static CencInfo deserialize(byte[] data)
```

toString

```
public String toString()
```

com.wowza.wms.drm.cenc Interface ICencDRMInfo

public interface **ICencDRMInfo**
extends

Interface for Common Encryption (CENC) DRM information classes. DRM system-specific metadata can be attached to a stream's CENC data to indicate that a stream or a media segment within that stream is protected by the particular DRM system, and may be authorized and decrypted by a player client using the provided DRM system metadata. The metadata provided by classes implementing ICencDRMInfo is inserted into a stream's manifest and/or fragment files. CENC DRM support is in accordance with the Common Encryption standard (ISO/IEC 23001-7).

Method Summary

void	<u>deserialize</u> (byte[] data) Sets all internal class data by deserializing the data from the provided byte buffer representation of the class.
byte[]	<u>generateKey</u> (byte[] KID) Generates a 128-bit content key for use in AES encryption.
String	<u>getMPEGDashCPSubElements</u> (boolean keyRotation) Returns a string defining zero or more custom xml elements to be inserted as sub-elements to the ContentProtection element that defines the DRM system in an MPEG-DASH MPD.
String	<u>getNameSpaces</u> () Returns a string defining zero or more custom xml name space strings to be inserted into an xml-formatted manifest, such as an MPEG-DASH MPD.
byte[]	<u>getPsshData</u> (boolean keyRotation) Returns the DRM system-specific binary data associated with the encryption of the media.
String	<u>getSystemId</u> () Returns the DRM system id.
String	<u>getSystemName</u> () Returns a string defining the DRM system name.
String	<u>getURL</u> () Returns the license acquisition URL.
boolean	<u>isComplete</u> () Verifies whether all required class parameters have been set, such as systemId.
byte[]	<u>serialize</u> () Returns a byte buffer with a serialized representation of the class, which could later be reconstituted back into the class via the deserialize method.
void	<u>setPsshData</u> (byte[] psshData) Sets the DRM system-specific binary data associated with the encryption of the media.
void	<u>setSystemId</u> (String systemId) Sets the DRM system id.
void	<u>setURL</u> (String url) Sets the license acquisition URL.

boolean	supportsKeyGeneration() Verifies whether dynamic key generation is supported.
void	updateKeyInfo() (byte[] KID, byte[] contentKey) Sets the new key id and a 128-bit content key values for use in AES encryption.

Methods

setSystemId

```
public void setSystemId(String systemId)
```

Sets the DRM system id.

Example: "12345678-1234-1234-1234-123456789012"

Parameters:

systemId - DRM system id

getSystemId

```
public String getSystemId()
```

Returns the DRM system id.

Returns:

DRM system id string

getURL

```
public String getURL()
```

Returns the license acquisition URL.

Returns:

license acquisition URL string

setURL

```
public void setURL(String url)
```

Sets the license acquisition URL.

Example: "http://www.mycompany.com/sdrm/rightsmanager.asmx"

Parameters:

url - license acquisition URL string

getNameSpaces

```
public String getNameSpaces()
```

(continued from last page)

Returns a string defining zero or more custom xml name space strings to be inserted into an xml-formatted manifest, such as an MPEG-DASH MPD.

Example: `xmlns:sdrma="urn:mycompany:superdrm:a" xmlns:sdrmb="urn:mycompany:superdrm:b"`

Returns:

zero or more xml name space strings separate by spaces, with a "" or 'null' return indicating no custom namespaces

getMPEGDashCPSubElements

```
public String getMPEGDashCPSubElements(boolean keyRotation)
```

Returns a string defining zero or more custom xml elements to be inserted as sub-elements to the ContentProtection element that defines the DRM system in an MPEG-DASH MPD. The element would typically contain Base64-encoded system-specific metadata.

Example: `<sdrma:hdr><!-- base64-encoded data --></sdrma:hdr>`

The resulting ContentProtection element would look something like:

```
<ContentProtection schemeIdUri="urn:uuid:12345678-1234-1234-1234-123456789012"
value="SuperDRM">
  <sdrma:hdr><!-- base64-encoded data --></sdrma:hdr>
</ContentProtection>
```

Parameters:

`keyRotation` - indicates whether key rotation is enabled

Returns:

string defining the MPEG-DASH ContentProtection sub-elements

getSystemName

```
public String getSystemName()
```

Returns a string defining the DRM system name. This string would be used as the value of the **value** attribute in the ContentProtection element that defines the DRM system in an MPEG-DASH MPD.

Example: `"SuperDRM"`

The resulting ContentProtection element in an MPEG-DASH MPD would look something like:

```
<ContentProtection schemeIdUri="urn:uuid:12345678-1234-1234-1234-123456789012"
value="SuperDRM">
  <sdrma:hdr><!-- base64-encoded data --></sdrma:hdr>
</ContentProtection>
```

Returns:

string defining the DRM system name

supportsKeyGeneration

```
public boolean supportsKeyGeneration()
```

Verifies whether dynamic key generation is supported. This capability is only required for dynamic random key rotation support.

Returns:

true if dynamic key generation is supported, else false

See Also:

[generateKey\(byte\[\]\)](#)

generateKey

```
public byte[] generateKey(byte[] KID)
```

Generates a 128-bit content key for use in AES encryption. Support for this method is only required when dynamic random key generation & rotation is supported.

Parameters:

KID - key id for which a content key should be generated

Returns:

byte buffer containing the generated content key, or null if not supported

See Also:

[supportsKeyGeneration\(\)](#)

updateKeyInfo

```
public void updateKeyInfo(byte[] KID,  
    byte[] contentKey)
```

Sets the new key id and a 128-bit content key values for use in AES encryption. Support for this method is only required in a Common Encryption scenario where dynamic random key generation & rotation is supported.

Parameters:

KID - new key id in effect

contentKey - new content key in effect

See Also:

[supportsKeyGeneration\(\)](#)

serialize

```
public byte[] serialize()
```

Returns a byte buffer with a serialized representation of the class, which could later be reconstituted back into the class via the deserialize method.

Returns:

byte buffer containing a serialized representation of the class

See Also:

[deserialize\(byte\[\]\)](#)

deserialize

```
public void deserialize(byte[] data)
```

Sets all internal class data by deserializing the data from the provided byte buffer representation of the class.

Parameters:

data - byte buffer containing the serialized representation of the class

See Also:

[serialize\(\)](#)

(continued from last page)

getPsshData

```
public byte[] getPsshData(boolean keyRotation)
```

Returns the DRM system-specific binary data associated with the encryption of the media. This data will be inserted into an ISOBMFF pssh box and identified by the associated systemId.

Parameters:

keyRotation - indicates whether key rotation is enabled

Returns:

byte buffer containing the DRM system-specific binary data for the pssh box

setPsshData

```
public void setPsshData(byte[] psshData)
```

Sets the DRM system-specific binary data associated with the encryption of the media. This data will be inserted into an ISOBMFF pssh box and identified by the associated systemId.

Parameters:

psshData - DRM system-specific binary data for the pssh box

isComplete

```
public boolean isComplete()
```

Verifies whether all required class parameters have been set, such as systemId.

Returns:

true if all required class parameters have been set, else false

Package

com.wowza.wms.dvr

com.wowza.wms.dvr Class DefaultDvrStreamVersionHandler

java.lang.Object

└─com.wowza.wms.dvr.DefaultDvrStreamVersionHandler

All Implemented Interfaces:

[IDvrStreamVersionHandler](#)

public class **DefaultDvrStreamVersionHandler**
extends Object
implements [IDvrStreamVersionHandler](#)

Default stream Version Handler. May be sub-classed

Constructor Summary

public	DefaultDvrStreamVersionHandler()
--------	--

Method Summary

IDvrStreamStore	determineExistingStoreForPlaying (IDvrStreamManager dvrMgr, String baseStreamName)
IDvrStreamStore	determineExistingStoreForRecording (IDvrStreamManager dvrMgr, String baseStreamName)
String	getArchiveStrategy (IDvrStreamManager dvrMgr, String baseStreamName)
boolean	handleArchivedStream (IDvrStreamManager dvrMgr, String baseStreamName, String vStreamName, java.util.SortedSet versions, DvrManifestHolder manifestHolder)
boolean	shouldDeleteArchivedStream (IDvrStreamManager dvrMgr, IDvrStreamStore store)
boolean	shouldLoadArchivedStream (IDvrStreamManager dvrMgr, String baseStreamName, String vStreamName, java.util.SortedSet versions, DvrManifestHolder manifest)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface [com.wowza.wms.dvr.IDvrStreamVersionHandler](#)

[determineExistingStoreForPlaying](#), [determineExistingStoreForRecording](#), [getArchiveStrategy](#), [handleArchivedStream](#), [shouldDeleteArchivedStream](#), [shouldLoadArchivedStream](#)

Constructors

DefaultDvrStreamVersionHandler

```
public DefaultDvrStreamVersionHandler()
```

Methods

getArchiveStrategy

```
public String getArchiveStrategy(IDvrStreamManager dvrMgr,  
    String baseStreamName)
```

determineExistingStoreForRecording

```
public IDvrStreamStore determineExistingStoreForRecording(IDvrStreamManager dvrMgr,  
    String baseStreamName)
```

determineExistingStoreForPlaying

```
public IDvrStreamStore determineExistingStoreForPlaying(IDvrStreamManager dvrMgr,  
    String baseStreamName)
```

handleArchivedStream

```
public boolean handleArchivedStream(IDvrStreamManager dvrMgr,  
    String baseStreamName,  
    String vStreamName,  
    java.util.SortedSet versions,  
    DvrManifestHolder manifestHolder)
```

shouldLoadArchivedStream

```
public boolean shouldLoadArchivedStream(IDvrStreamManager dvrMgr,  
    String baseStreamName,  
    String vStreamName,  
    java.util.SortedSet versions,  
    DvrManifestHolder manifest)
```

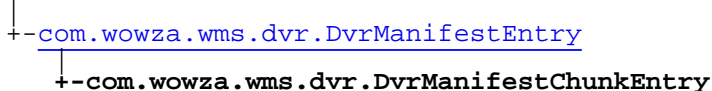
shouldDeleteArchivedStream

```
public boolean shouldDeleteArchivedStream(IDvrStreamManager dvrMgr,  
    IDvrStreamStore store)
```

com.wowza.wms.dvr

Class DvrManifestChunkEntry

java.lang.Object



public class **DvrManifestChunkEntry**
 extends [DvrManifestEntry](#)

Entry in DVR manifest that contains an associated chunk.

Field Summary

protected	artifact
protected	encryptions

Fields inherited from class [com.wowza.wms.dvr.DvrManifestEntry](#)

[dvrStart](#), [dvrStop](#), [index](#), [MANIFESTFILE_KEY_ARTIFACT](#), [MANIFESTFILE_KEY_AUDIO_CODEC](#), [MANIFESTFILE_KEY_CHUNKINDEX](#), [MANIFESTFILE_KEY_DVRTIME](#), [MANIFESTFILE_KEY_ENCRYPTIONS](#), [MANIFESTFILE_KEY_INDEX](#), [MANIFESTFILE_KEY_METADATA](#), [MANIFESTFILE_KEY_NAME](#), [MANIFESTFILE_KEY_PACKETTIME](#), [MANIFESTFILE_KEY_SIZE](#), [MANIFESTFILE_KEY_START](#), [MANIFESTFILE_KEY_STOP](#), [MANIFESTFILE_KEY_TYPE](#), [MANIFESTFILE_KEY_UTCTIME](#), [MANIFESTFILE_KEY_VIDEO_CODEC](#), [packetTime](#), [SERIALIZE_CURRENT_VERSION](#), [type](#), [utcTime](#)

Constructor Summary

public	DvrManifestChunkEntry (int type, int index, long dvrStart, long dvrStop, long packetTime, long utcTime, DvrChunkArtifact artifact) Constructor
public	DvrManifestChunkEntry (int type, int index, long dvrStart, long dvrStop, long packetTime, long utcTime, DvrChunkArtifact artifact, DvrEncryptionInfoHolder encryptions) Constructor

Method Summary

String	getArtifactsTextRepresentation ()
DvrChunkArtifact	getDvrArtifact () Get DVR artifact reference.
DvrEncryptionInfoHolder	getEncryptions () Get associated encryptions.
String	getEncryptionsTextRepresentation ()
String	getManifestRepresentation ()

void	<u>serialize</u> (java.io.DataOutputStream out)
void	<u>setEncryptions</u> (DvrEncryptionInfoHolder encryptions) Set associated encryptions.
String	<u>toString</u> ()

Methods inherited from class [com.wowza.wms.dvr.DvrManifestEntry](#)

[encodeBytes](#), [getCommonInitialTextRepString](#), [getDuration](#), [getIndex](#),
[getManifestRepresentation](#), [getPacketStartTime](#), [getStartTimecode](#), [getStopTimecode](#),
[getType](#), [getUtcStartTime](#), [serialize](#), [serialize](#)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait,
wait

Fields

artifact

protected com.wowza.wms.dvr.DvrChunkArtifact **artifact**

encryptions

protected com.wowza.wms.dvr.DvrEncryptionInfoHolder **encryptions**

Constructors

DvrManifestChunkEntry

```
public DvrManifestChunkEntry(int type,
                             int index,
                             long dvrStart,
                             long dvrStop,
                             long packetTime,
                             long utcTime,
                             DvrChunkArtifact artifact)
```

Constructor

Parameters:

type - type
index - manifest index
dvrStart - start time (ms in DVR time scale)
dvrStop - stop time (ms in DVR time scale)
utcTime
packetTime
artifact - reference to the chunk artifact

(continued from last page)

DvrManifestChunkEntry

```
public DvrManifestChunkEntry(int type,
                             int index,
                             long dvrStart,
                             long dvrStop,
                             long packetTime,
                             long utcTime,
                             DvrChunkArtifact artifact,
                             DvrEncryptionInfoHolder encryptions)
```

Constructor

Parameters:

type - type
index - manifest index
dvrStart - start time (ms in DVR time scale)
dvrStop - stop time (ms in DVR time scale)
utcTime
packetTime
artifact - reference to the chunk artifact
encryptions - associated encryptions

Methods

getDvrArtifact

```
public DvrChunkArtifact getDvrArtifact()
```

Get DVR artifact reference.

Returns:

DVR artifact reference.

setEncryptions

```
public void setEncryptions(DvrEncryptionInfoHolder encryptions)
```

Set associated encryptions.

Parameters:

encryptions - encryptions

getEncryptions

```
public DvrEncryptionInfoHolder getEncryptions()
```

Get associated encryptions.

Returns:

encryptions

serialize

```
public void serialize(java.io.DataOutputStream out)
```

Serialize manifest record.

(continued from last page)

getManifestRepresentation

```
public String getManifestRepresentation()
```

Get textual representation of record for textual manifest usage.

getEncryptionsTextRepresentation

```
protected String getEncryptionsTextRepresentation()
```

getArtifactsTextRepresentation

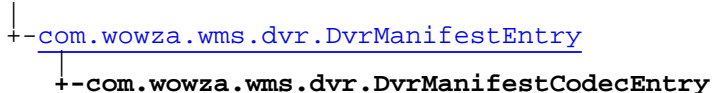
```
protected String getArtifactsTextRepresentation()
```

toString

```
public String toString()
```


com.wowza.wms.dvr Class DvrManifestCodecEntry

java.lang.Object



public class **DvrManifestCodecEntry**
extends [DvrManifestEntry](#)

Entry in DVR manifest that contains codec information

Fields inherited from class [com.wowza.wms.dvr.DvrManifestEntry](#)

[dvrStart](#), [dvrStop](#), [index](#), [MANIFESTFILE_KEY_ARTIFACT](#), [MANIFESTFILE_KEY_AUDIO_CODEC](#), [MANIFESTFILE_KEY_CHUNKINDEX](#), [MANIFESTFILE_KEY_DVRTIME](#), [MANIFESTFILE_KEY_ENCRYPTIONS](#), [MANIFESTFILE_KEY_INDEX](#), [MANIFESTFILE_KEY_METADATA](#), [MANIFESTFILE_KEY_NAME](#), [MANIFESTFILE_KEY_PACKETTIME](#), [MANIFESTFILE_KEY_SIZE](#), [MANIFESTFILE_KEY_START](#), [MANIFESTFILE_KEY_STOP](#), [MANIFESTFILE_KEY_TYPE](#), [MANIFESTFILE_KEY_UTCTIME](#), [MANIFESTFILE_KEY_VIDEO_CODEC](#), [packetTime](#), [SERIALIZE_CURRENT_VERSION](#), [type](#), [utcTime](#)

Constructor Summary

public	DvrManifestCodecEntry (int index, long dvrStart, long packetTime, long utcTime, com.wowza.wms.media.model.MediaCodecInfoAudio audioCodec, com.wowza.wms.media.model.MediaCodecInfoVideo videoCodec) Constructor
--------	--

Method Summary

com.wowza.wms.media.model.MediaCodecInfoAudio	getAudioCodec () Get audio codec information
String	getManifestRepresentation ()
com.wowza.wms.media.model.MediaCodecInfoVideo	getVideoCodec () Get video codec information
void	serialize (java.io.DataOutputStream out)
String	toString ()

Methods inherited from class [com.wowza.wms.dvr.DvrManifestEntry](#)

[encodeBytes](#), [getCommonInitialTextRepString](#), [getDuration](#), [getIndex](#), [getManifestRepresentation](#), [getPacketStartTime](#), [getStartTimecode](#), [getStopTimecode](#), [getType](#), [getUtcStartTime](#), [serialize](#), [serialize](#)

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

Constructors

DvrManifestCodecEntry

```
public DvrManifestCodecEntry(int index,
                             long dvrStart,
                             long packetTime,
                             long utcTime,
                             com.wowza.wms.media.model.MediaCodecInfoAudio audioCodec,
                             com.wowza.wms.media.model.MediaCodecInfoVideo videoCodec)
```

Constructor

Parameters:

index - manifest index
dvrStart - start time (ms in DVR time scale)
audioCodec - audio codec info
videoCodec - video codec info

Methods

getVideoCodec

```
public com.wowza.wms.media.model.MediaCodecInfoVideo getVideoCodec()
```

Get video codec information

Returns:

video codec information

getAudioCodec

```
public com.wowza.wms.media.model.MediaCodecInfoAudio getAudioCodec()
```

Get audio codec information

Returns:

audio codec information

serialize

```
public void serialize(java.io.DataOutputStream out)
```

Serialize manifest record.

getManifestRepresentation

```
public String getManifestRepresentation()
```

Get textual representation of record for textual manifest usage.

(continued from last page)

toString

```
public String toString()
```

com.wowza.wms.dvr Class DvrManifestEntry

java.lang.Object

└─com.wowza.wms.dvr.DvrManifestEntry

Direct Known Subclasses:

[DvrManifestTimeMapEntry](#), [DvrManifestOnMetadataEntry](#), [DvrManifestCodecEntry](#), [DvrManifestChunkEntry](#)

public abstract class **DvrManifestEntry**
extends Object

Abstract class representing entry in DVR manifest.

Field Summary

protected	dvrStart
protected	dvrStop
protected	index
public static final	MANIFESTFILE_KEY_ARTIFACT Value: artifact
public static final	MANIFESTFILE_KEY_AUDIO_CODEC Value: aCodec
public static final	MANIFESTFILE_KEY_CHUNKINDEX Value: chunkIndex
public static final	MANIFESTFILE_KEY_DVRTIME Value: dvrTime
public static final	MANIFESTFILE_KEY_ENCRYPTIONS Value: enc
public static final	MANIFESTFILE_KEY_INDEX Value: index
public static final	MANIFESTFILE_KEY_METADATA Value: metadata
public static final	MANIFESTFILE_KEY_NAME Value: name

public static final	MANIFESTFILE_KEY_PACKETTIME Value: packetTime
public static final	MANIFESTFILE_KEY_SIZE Value: size
public static final	MANIFESTFILE_KEY_START Value: start
public static final	MANIFESTFILE_KEY_STOP Value: stop
public static final	MANIFESTFILE_KEY_TYPE Value: type
public static final	MANIFESTFILE_KEY_UTCTIME Value: utcTime
public static final	MANIFESTFILE_KEY_VIDEO_CODEC Value: vCodec
protected	packetTime
protected static final	SERIALIZE_CURRENT_VERSION Value: 2
protected	type
protected	utcTime

Constructor Summary

public	DvrManifestEntry (int type, int index, long dvrStart, long dvrStop, long packetTime, long utcTime) Constructor Valid types include: IVHost.CONTENTTYPE_AUDIO , IVHost.CONTENTTYPE_VIDEO , IVHost.CONTENTTYPE_DATA , IDvrManifest.ON_METADATA_TYPE , IDvrManifest.CODEC_TYPE , or IDvrManifest.TIME_MAP_TYPE
--------	--

Method Summary

String	encodeBytes (byte[] bytes) Encode string of bytes as Base64.
String	getCommonInitialTextRepString ()
long	getDuration () Get duration
int	getIndex () Get manifest index.

abstract String	<code>getManifestRepresentation()</code> Get textual representation of record for textual manifest usage.
long	<code>getPacketStartTime()</code> Get chunk start timecode in packetTime units.
long	<code>getStartTimecode()</code> Get start timecode.
long	<code>getStopTimecode()</code> Get stop timecode.
int	<code>getType()</code> Get type of manifest record.
long	<code>getUtcStartTime()</code> Get chunk start timecode in UTC units.
byte[]	<code>serialize()</code> Serialize manifest record.
abstract void	<code>serialize(java.io.DataOutputStream out)</code> Serialize manifest record.

Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Fields

MANIFESTFILE_KEY_INDEX

```
public static final java.lang.String MANIFESTFILE_KEY_INDEX
```

Constant value: **index**

MANIFESTFILE_KEY_TYPE

```
public static final java.lang.String MANIFESTFILE_KEY_TYPE
```

Constant value: **type**

MANIFESTFILE_KEY_START

```
public static final java.lang.String MANIFESTFILE_KEY_START
```

Constant value: **start**

MANIFESTFILE_KEY_STOP

```
public static final java.lang.String MANIFESTFILE_KEY_STOP
```

(continued from last page)

Constant value: **stop**

MANIFESTFILE_KEY_SIZE

```
public static final java.lang.String MANIFESTFILE_KEY_SIZE
```

Constant value: **size**

MANIFESTFILE_KEY_ARTIFACT

```
public static final java.lang.String MANIFESTFILE_KEY_ARTIFACT
```

Constant value: **artifact**

MANIFESTFILE_KEY_DVRTIME

```
public static final java.lang.String MANIFESTFILE_KEY_DVRTIME
```

Constant value: **dvrTime**

MANIFESTFILE_KEY_CHUNKINDEX

```
public static final java.lang.String MANIFESTFILE_KEY_CHUNKINDEX
```

Constant value: **chunkIndex**

MANIFESTFILE_KEY_PACKETTIME

```
public static final java.lang.String MANIFESTFILE_KEY_PACKETTIME
```

Constant value: **packetTime**

MANIFESTFILE_KEY_UTCTIME

```
public static final java.lang.String MANIFESTFILE_KEY_UTCTIME
```

Constant value: **utcTime**

MANIFESTFILE_KEY_NAME

```
public static final java.lang.String MANIFESTFILE_KEY_NAME
```

Constant value: **name**

MANIFESTFILE_KEY_AUDIO_CODEC

```
public static final java.lang.String MANIFESTFILE_KEY_AUDIO_CODEC
```

Constant value: **aCodec**

(continued from last page)

MANIFESTFILE_KEY_VIDEO_CODEC

```
public static final java.lang.String MANIFESTFILE_KEY_VIDEO_CODEC
```

Constant value: **vCodec**

MANIFESTFILE_KEY_ENCRYPTIONS

```
public static final java.lang.String MANIFESTFILE_KEY_ENCRYPTIONS
```

Constant value: **enc**

MANIFESTFILE_KEY_METADATA

```
public static final java.lang.String MANIFESTFILE_KEY_METADATA
```

Constant value: **metadata**

SERIALIZE_CURRENT_VERSION

```
protected static final byte SERIALIZE_CURRENT_VERSION
```

Constant value: **2**

dvrStart

```
protected long dvrStart
```

dvrStop

```
protected long dvrStop
```

packetTime

```
protected long packetTime
```

utcTime

```
protected long utcTime
```

index

```
protected int index
```

(continued from last page)

type

protected int **type**

Constructors

DvrManifestEntry

```
public DvrManifestEntry(int type,  
                        int index,  
                        long dvrStart,  
                        long dvrStop,  
                        long packetTime,  
                        long utcTime)
```

Constructor Valid types include: [IVHost.CONTENTTYPE_AUDIO](#), [IVHost.CONTENTTYPE_VIDEO](#), [IVHost.CONTENTTYPE_DATA](#), [IDvrManifest.ON_METADATA_TYPE](#), [IDvrManifest.CODEC_TYPE](#), or [IDvrManifest.TIME_MAP_TYPE](#)

Parameters:

type - type
index - manifest index
dvrStart - start time (ms in DVR time scale)
dvrStop - stop time (ms in DVR time scale)
packetTime
utcTime

Methods

getStartTimecode

```
public long getStartTimecode()
```

Get start timecode. In milliseconds, DVR time base.

Returns:

start time

getStopTimecode

```
public long getStopTimecode()
```

Get stop timecode. In milliseconds, DVR time base.

Returns:

stop time

getPacketStartTime

```
public long getPacketStartTime()
```

Get chunk start timecode in packetTime units. In milliseconds.

Returns:

packet start time

(continued from last page)

getUtcStartTime

```
public long getUtcStartTime()
```

Get chunk start timecode in UTC units. In milliseconds.

Returns:

UTC start time

getDuration

```
public long getDuration()
```

Get duration

Returns:

duration in ms

getType

```
public int getType()
```

Get type of manifest record. Valid types include: [IVHost.CONTENTTYPE_AUDIO](#), [IVHost.CONTENTTYPE_VIDEO](#), [IVHost.CONTENTTYPE_DATA](#), [IDvrManifest.ON_METADATA_TYPE](#), [IDvrManifest.CODEC_TYPE](#), or [IDvrManifest.TIME_MAP_TYPE](#)

Returns:

type

getIndex

```
public int getIndex()
```

Get manifest index.

Returns:

index

serialize

```
public abstract void serialize(java.io.DataOutputStream out)
```

Serialize manifest record.

Parameters:

out - output stream

serialize

```
public byte[] serialize()
```

Serialize manifest record.

Returns:

serialized data

(continued from last page)

getManifestRepresentation

```
public abstract String getManifestRepresentation()
```

Get textual representation of record for textual manifest usage.

Returns:

textual representation of entry

getCommonInitialTextRepString

```
protected String getCommonInitialTextRepString()
```

encodeBytes

```
protected String encodeBytes(byte[] bytes)
```

Encode string of bytes as Base64. Provides check for null buffer or empty buffer.

Parameters:

bytes - buffer to encode.

Returns:

Base64 encoding or "".

com.wowza.wms.dvr

Class DvrManifestEntryFactory



```
public class DvrManifestEntryFactory
extends Object
```

Deserialize manifest entry blobs.

Constructor Summary

public	DvrManifestEntryFactory()
--------	---

Method Summary

static DvrManifestEntry	deserialize (byte[] data)
static DvrManifestEntry	deserialize (java.nio.ByteBuffer buffer)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

DvrManifestEntryFactory

```
public DvrManifestEntryFactory()
```

Methods

deserialize

```
public static DvrManifestEntry deserialize(byte[] data)
```

deserialize

```
public static DvrManifestEntry deserialize(java.nio.ByteBuffer buffer)
```

com.wowza.wms.dvr Class DvrManifestEntryRange

java.lang.Object

└─com.wowza.wms.dvr.DvrManifestEntryRange

public class **DvrManifestEntryRange**
extends Object

Class that compactly represents a range of indices of a certain manifest type.

Constructor Summary

public	DvrManifestEntryRange (int type, int startIndex, int endIndex) Construct a range of a given type.
--------	--

Method Summary

void	deserialize (byte[] data) Deserialize.
int	getEndIndex () Get end index.
IndexRange	getRange () Get index range.
int	getStartIndex () Get start index.
int	getType () Get range type
boolean	isEmpty ()
boolean	isInRange (int index) Is index in range.
byte[]	serialize () Serialize range.
void	setRange (IndexRange range) Set index range.
String	toString ()

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

(continued from last page)

Constructors

DvrManifestEntryRange

```
public DvrManifestEntryRange(int type,  
                             int startIndex,  
                             int endIndex)
```

Construct a range of a given type.

Parameters:

type - manifest type

startIndex - initial member of range

endIndex - final member of range

Methods

getType

```
public int getType()
```

Get range type

Returns:

type.

getRange

```
public IndexRange getRange()
```

Get index range.

Returns:

index range

setRange

```
public void setRange(IndexRange range)
```

Set index range.

Parameters:

range - index range.

getStartIndex

```
public int getStartIndex()
```

Get start index.

Returns:

start index (or -1 if undefined)

getEndIndex

```
public int getEndIndex()
```

Get end index.

(continued from last page)

Returns:

end index (or -1 if undefined)

isInRange

```
public boolean isInRange(int index)
```

Is index in range.

Parameters:

index - index in question.

Returns:

true if part of range.

serialize

```
public byte[] serialize()
```

Serialize range.

Returns:

out buffer containing serialized range.

deserialize

```
public void deserialize(byte[] data)
```

Deserialize. Called after no-op constructor.

Parameters:

data - buffer to deserialize.

isEmpty

```
public boolean isEmpty()
```

toString

```
public String toString()
```

com.wowza.wms.dvr Class DvrManifestEntryRangeGroup

java.lang.Object

└─com.wowza.wms.dvr.DvrManifestEntryRangeGroup

```
public class DvrManifestEntryRangeGroup
    extends Object
```

A group of Manifest Ranges. The group may contain ranges of different types.

Field Summary

public	ranges
--------	------------------------

Constructor Summary

public	DvrManifestEntryRangeGroup()
--------	--

Method Summary

void	addRange(DvrManifestEntryRange range) Add an index range to the group.
void	deserialize(byte[] data) Deserialize.
boolean	isEmpty() Determine if group of ranges is empty
boolean	isInRange(int type, int index) Determine if index of given type is contained in the group of ranges.
byte[]	serialize() Serialize range group
String	toString()

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

ranges

```
public java.util.List ranges
```


Constructors

DvrManifestEntryRangeGroup

```
public DvrManifestEntryRangeGroup()
```

Methods

addRange

```
public void addRange(DvrManifestEntryRange range)
```

Add an index range to the group.

Parameters:

range

isInRange

```
public boolean isInRange(int type,  
    int index)
```

Determine if index of given type is contained in the group of ranges.

Parameters:

type - range type

index - index in question

Returns:

true iof in range.

isEmpty

```
public boolean isEmpty()
```

Determine if group of ranges is empty

Returns:

true if empty

serialize

```
public byte[] serialize()
```

Serialize range group

Returns:

bytes representing range group

deserialize

```
public void deserialize(byte[] data)
```

Deserialize. Called after no-op constructor.

(continued from last page)

Parameters:

data - buffer to deserialize.

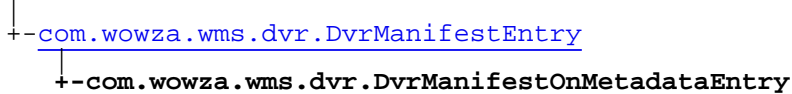
toString

```
public String toString()
```

com.wowza.wms.dvr

Class DvrManifestOnMetadataEntry

java.lang.Object



public class **DvrManifestOnMetadataEntry**
 extends [DvrManifestEntry](#)

Entry in DVR manifest that contains onMetadata information and associated onMetadata chunk.

Field Summary

protected	artifact
protected	data

Fields inherited from class [com.wowza.wms.dvr.DvrManifestEntry](#)

[dvrStart](#), [dvrStop](#), [index](#), [MANIFESTFILE_KEY_ARTIFACT](#), [MANIFESTFILE_KEY_AUDIO_CODEC](#), [MANIFESTFILE_KEY_CHUNKINDEX](#), [MANIFESTFILE_KEY_DVRTIME](#), [MANIFESTFILE_KEY_ENCRYPTIONS](#), [MANIFESTFILE_KEY_INDEX](#), [MANIFESTFILE_KEY_METADATA](#), [MANIFESTFILE_KEY_NAME](#), [MANIFESTFILE_KEY_PACKETTIME](#), [MANIFESTFILE_KEY_SIZE](#), [MANIFESTFILE_KEY_START](#), [MANIFESTFILE_KEY_STOP](#), [MANIFESTFILE_KEY_TYPE](#), [MANIFESTFILE_KEY_UTCTIME](#), [MANIFESTFILE_KEY_VIDEO_CODEC](#), [packetTime](#), [SERIALIZE_CURRENT_VERSION](#), [type](#), [utcTime](#)

Constructor Summary

public	DvrManifestOnMetadataEntry (int index, long start, long packetTime, long utcTime, DvrChunkArtifact artifact, byte[] data) Constructor
--------	--

Method Summary

String	getArtifactsTextRepresentation()
byte[]	getData() Get onMetadata information.
DvrChunkArtifact	getDvrArtifact()
String	getManifestRepresentation()
void	serialize (java.io.DataOutputStream out)
String	toString()

Methods inherited from class [com.wowza.wms.dvr.DvrManifestEntry](#)

[getBytes](#), [getCommonInitialTextRepString](#), [getDuration](#), [getIndex](#),
[getManifestRepresentation](#), [getPacketStartTime](#), [getStartTimecode](#), [getStopTimecode](#),
[getType](#), [getUtcStartTime](#), [serialize](#), [serialize](#)

Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`,
`wait`

Fields

artifact

protected `com.wowza.wms.dvr.DvrChunkArtifact` **artifact**

data

protected `byte` **data**

Constructors

DvrManifestOnMetadataEntry

```
public DvrManifestOnMetadataEntry(int index,  
                                   long start,  
                                   long packetTime,  
                                   long utcTime,  
                                   DvrChunkArtifact artifact,  
                                   byte[] data)
```

Constructor

Parameters:

`index` - manifest index
`start` - start time (ms in DVR time scale)
`utcTime`
`packetTime`
`artifact` - reference to the chunk artifact
`data` - buffer of onMetadata information

Methods

getDvrArtifact

```
public DvrChunkArtifact getDvrArtifact()
```

getData

```
public byte[] getData()
```

(continued from last page)

Get onMetadata information.

Returns:

onMetadata information.

serialize

```
public void serialize(java.io.DataOutputStream out)
```

Serialize manifest record.

getManifestRepresentation

```
public String getManifestRepresentation()
```

Get textual representation of record for textual manifest usage.

getArtifactsTextRepresentation

```
protected String getArtifactsTextRepresentation()
```

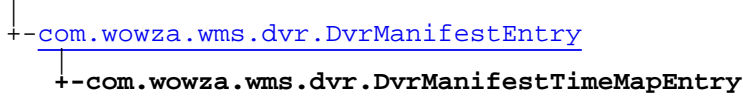
toString

```
public String toString()
```

com.wowza.wms.dvr

Class DvrManifestTimeMapEntry

java.lang.Object



public class **DvrManifestTimeMapEntry**
 extends [DvrManifestEntry](#)

Entry in DVR manifest that contains time map information.

Fields inherited from class [com.wowza.wms.dvr.DvrManifestEntry](#)

[dvrStart](#), [dvrStop](#), [index](#), [MANIFESTFILE_KEY_ARTIFACT](#), [MANIFESTFILE_KEY_AUDIO_CODEC](#), [MANIFESTFILE_KEY_CHUNKINDEX](#), [MANIFESTFILE_KEY_DVRTIME](#), [MANIFESTFILE_KEY_ENCRYPTIONS](#), [MANIFESTFILE_KEY_INDEX](#), [MANIFESTFILE_KEY_METADATA](#), [MANIFESTFILE_KEY_NAME](#), [MANIFESTFILE_KEY_PACKETTIME](#), [MANIFESTFILE_KEY_SIZE](#), [MANIFESTFILE_KEY_START](#), [MANIFESTFILE_KEY_STOP](#), [MANIFESTFILE_KEY_TYPE](#), [MANIFESTFILE_KEY_UTCTIME](#), [MANIFESTFILE_KEY_VIDEO_CODEC](#), [packetTime](#), [SERIALIZE_CURRENT_VERSION](#), [type](#), [utcTime](#)

Constructor Summary

public	DvrManifestTimeMapEntry (int index, int chunkIndex, long start, long packetTime, long utcTime, TimeMapRecord timeMap) Constructor
--------	--

Method Summary

int	getChunkIndex () Get chunk index that correlates to this time mapping
String	getManifestRepresentation ()
TimeMapRecord	getTimeMapping () Get time map record.
void	serialize (java.io.DataOutputStream out)
String	toString ()

Methods inherited from class [com.wowza.wms.dvr.DvrManifestEntry](#)

[encodeBytes](#), [getCommonInitialTextRepString](#), [getDuration](#), [getIndex](#), [getManifestRepresentation](#), [getPacketStartTime](#), [getStartTimecode](#), [getStopTimecode](#), [getType](#), [getUtcStartTime](#), [serialize](#), [serialize](#)

Methods inherited from class java.lang.Object

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

Constructors

DvrManifestTimeMapEntry

```
public DvrManifestTimeMapEntry(int index,
                               int chunkIndex,
                               long start,
                               long packetTime,
                               long utcTime,
                               TimeMapRecord timeMap)
```

Constructor

Parameters:

index - manifest index
start - start time (ms in DVR time scale)
timeMap - time map record

Methods

getTimeMapping

```
public TimeMapRecord getTimeMapping()
```

Get time map record.

Returns:

time map record

getChunkIndex

```
public int getChunkIndex()
```

Get chunk index that correlates to this time mapping

Returns:

chunk index

getManifestRepresentation

```
public String getManifestRepresentation()
```

Get textual representation of record for textual manifest usage.

serialize

```
public void serialize(java.io.DataOutputStream out)
```

Serialize manifest record.

toString

```
public String toString()
```

com.wowza.wms.dvr Interface IDvrChannelManifest

All Subinterfaces:

[IDvrTimeMap](#)

public interface **IDvrChannelManifest**
extends

Method Summary

long	expandEndTime (long dvrEndTime)
long	expandStartTime (long dvrStartTime)
long	getClosestStartTime (long t)
DvrManifestEntry	getFirstEntry ()
int	getFirstIndex ()
java.util.Map	getIndexMap ()
DvrManifestEntry	getLastLiveEntry ()
DvrManifestEntry	getLastRecordedEntry ()
int	getLastRecordedIndex ()
long	getLiveDuration ()
java.util.List	getLiveEntries ()
java.util.List	getLiveEntries (long startTime)
java.util.List	getLiveEntriesWithLimit (long t, int limit)
DvrManifestEntryRange	getLiveRangeEndingBeforeTime (long time)
DvrManifestEntryRange	getLiveRangeEndingBeforeTime (long time, boolean skipFirst)
java.util.List	getLiveTailEntries (int index)
int	getNumberLiveEntries (long dvrStart)
int	getNumberLiveEntries (long dvrStart, long dvrEnd)

int	<u>getNumberRecordedEntries</u> (long dvrStart)
int	<u>getNumberRecordedEntries</u> (long dvrStart, long dvrEnd)
long	<u>getRecordedDuration</u> ()
java.util.List	<u>getRecordedEntries</u> ()
java.util.List	<u>getRecordedEntries</u> (long dvrStartTime)
java.util.List	<u>getRecordedEntries</u> (long dvrStartTime, long dvrEndTime)
java.util.List	<u>getRecordedEntriesInRange</u> (int startIndex, int endIndex)
java.util.List	<u>getRecordedEntriesWithLimit</u> (long t, int limit)
<u>DvrManifestEntry</u>	<u>getRecordedEntryByIndex</u> (int index)
<u>DvrManifestEntry</u>	<u>getRecordedEntryByTimeKey</u> (long t)
<u>DvrManifestEntry</u>	<u>getRecordedEntryStartingBeforeTime</u> (long t, boolean inclusive)
int	<u>getType</u> ()
boolean	<u>isEmpty</u> ()

Methods

isEmpty

```
public boolean isEmpty()
```

getType

```
public int getType()
```

getLastRecordedIndex

```
public int getLastRecordedIndex()
```

getRecordedEntries

```
public java.util.List getRecordedEntries()
```

(continued from last page)

getRecordedEntriesInRange

```
public java.util.List getRecordedEntriesInRange(int startIndex,  
int endIndex)
```

getIndexMap

```
public java.util.Map getIndexMap()
```

getRecordedEntryByTimeKey

```
public DvrManifestEntry getRecordedEntryByTimeKey(long t)
```

getRecordedEntryByIndex

```
public DvrManifestEntry getRecordedEntryByIndex(int index)
```

getRecordedEntryStartingBeforeTime

```
public DvrManifestEntry getRecordedEntryStartingBeforeTime(long t,  
boolean inclusive)
```

getRecordedEntries

```
public java.util.List getRecordedEntries(long dvrStartTime)
```

getRecordedEntries

```
public java.util.List getRecordedEntries(long dvrStartTime,  
long dvrEndTime)
```

getRecordedEntriesWithLimit

```
public java.util.List getRecordedEntriesWithLimit(long t,  
int limit)
```

getRecordedDuration

```
public long getRecordedDuration()
```

getLiveDuration

```
public long getLiveDuration()
```

getClosestStartTime

```
public long getClosestStartTime(long t)
```

getLiveEntries

```
public java.util.List getLiveEntries()
```

getLiveTailEntries

```
public java.util.List getLiveTailEntries(int index)
```

getNumberLiveEntries

```
public int getNumberLiveEntries(long dvrStart)
```

getNumberLiveEntries

```
public int getNumberLiveEntries(long dvrStart,  
                                long dvrEnd)
```

getNumberRecordedEntries

```
public int getNumberRecordedEntries(long dvrStart)
```

getNumberRecordedEntries

```
public int getNumberRecordedEntries(long dvrStart,  
                                long dvrEnd)
```

getLiveEntries

```
public java.util.List getLiveEntries(long startTime)
```

(continued from last page)

getLiveEntriesWithLimit

```
public java.util.List getLiveEntriesWithLimit(long t,  
int limit)
```

getFirstIndex

```
public int getFirstIndex()
```

getFirstEntry

```
public DvrManifestEntry getFirstEntry()
```

getLastLiveEntry

```
public DvrManifestEntry getLastLiveEntry()
```

getLastRecordedEntry

```
public DvrManifestEntry getLastRecordedEntry()
```

getLiveRangeEndingBeforeTime

```
public DvrManifestEntryRange getLiveRangeEndingBeforeTime(long time)
```

getLiveRangeEndingBeforeTime

```
public DvrManifestEntryRange getLiveRangeEndingBeforeTime(long time,  
boolean skipFirst)
```

expandStartTime

```
public long expandStartTime(long dvrStartTime)
```

expandEndTime

```
public long expandEndTime(long dvrEndTime)
```

com.wowza.wms.dvr Interface IDvrChunker

public interface **IDvrChunker**
extends

Method Summary

long	<code>calculateChunkGroupTime</code> (long dvrTime)
String	<code>determineChunkGroupIdentifier</code> (long dvrTime)
int	<code>getChunkGroupDuration</code> ()
void	<code>setChunkGroupDuration</code> (int chunkGrouping)

Methods

`getChunkGroupDuration`

public int **getChunkGroupDuration**()

`determineChunkGroupIdentifier`

public String **determineChunkGroupIdentifier**(long dvrTime)

`calculateChunkGroupTime`

public long **calculateChunkGroupTime**(long dvrTime)

`setChunkGroupDuration`

public void **setChunkGroupDuration**(int chunkGrouping)

com.wowza.wms.dvr Interface IDvrChunkMemoryCache

All Superinterfaces:

[IDvrRawChunkProvider](#)

public interface **IDvrChunkMemoryCache**
extends [IDvrRawChunkProvider](#)

Interface for managing in-memory cache of chunks. The implementor is responsible for implementing the desired algorithm for keeping DVR chunks in memory.

Method Summary

void	addToCache (DvrManifestChunkEntry entry, DvrChunk chunk) Provide in-memory cache the opportunity to add the DVR chunk to its cache.
void	init (IDvrStreamStore store) Called to initialize the in-memory cache.
DvrChunk	retrieveRawChunk (DvrManifestChunkEntry entry)

Methods inherited from interface [com.wowza.wms.dvr.IDvrRawChunkProvider](#)

[retrieveRawChunk](#)

Methods

init

public void **init**([IDvrStreamStore](#) store)

Called to initialize the in-memory cache.

Parameters:

store - The DVR stream store associated with this cache.

retrieveRawChunk

public DvrChunk **retrieveRawChunk**([DvrManifestChunkEntry](#) entry)

addToCache

public void **addToCache**([DvrManifestChunkEntry](#) entry,
DvrChunk chunk)

Provide in-memory cache the opportunity to add the DVR chunk to its cache. The implementation may choose to cache the chunk or ignore it.

Parameters:

(continued from last page)

entry - DVR manifest entry
chunk - the chunk.

com.wowza.wms.dvr Interface IDvrChunkUtcDelegate

public interface **IDvrChunkUtcDelegate**
extends

Delegate class for determining utc time of DVR chunk.

Method Summary

long	<code>determineUtcTime</code> (long currentUtc, long aChunkStartUtc, long aFirstPacketUtc, AMFPacket aFirstPacket, long vChunkStartUtc, long vFirstPacketUtc, AMFPacket vFirstPacket)
void	<code>init</code> (IDvrStreamStore store)

Methods

init

public void **init**([IDvrStreamStore](#) store)

determineUtcTime

```
public long determineUtcTime(long currentUtc,  
    long aChunkStartUtc,  
    long aFirstPacketUtc,  
    AMFPacket aFirstPacket,  
    long vChunkStartUtc,  
    long vFirstPacketUtc,  
    AMFPacket vFirstPacket)
```


com.wowza.wms.dvr Interface IDvrConstants

All Superinterfaces:
[IDvrPrivateConstants](#)

public interface **IDvrConstants**
extends [IDvrPrivateConstants](#)

Nested Class Summary

class	IDvrConstants.DvrTimeScale IDvrConstants.DvrTimeScale
-------	--

Field Summary

public static final	ARCHIVE_STRATEGY_APPEND Value: append
public static final	ARCHIVE_STRATEGY_DELETE Value: delete
public static final	ARCHIVE_STRATEGY_VERSION Value: version
public static final	DEFAULT_CHUNK_MEMORY_CACHESIZE Default value for DVR Property "chunkMemoryCacheSize". Value: 10
public static final	DEFAULT_CUPERTINO_PLAYLIST_GZIP_THRESHOLD Default value for property "dvrCupertinoPlaylistGzipThreshold" #see PROPERTY_CUPERTINO_PLAYLIST_GZIP_THRESHOLD Value: 4000
public static final	DEFAULT_PLAYLIST_REQUEST_DELEGATE Value: com.wowza.wms.dvr.impl.DvrStartDurationPlaylistRequestDelegate
public static final	DEFAULT_PROPERTY_ALLOWABLE_AV_PACKET_DELTA Default value for DVR Property "dvrAllowableAVPacketDelta". Value: 2000
public static final	DEFAULT_PROPERTY_APPEND_DISCONTINUITY_DELTA Default value for DVR Property "dvrAppendDiscontinuityDelta". Value: 0
public static final	DEFAULT_PROPERTY_AUDIO_ONLY_CHUNK_TARGET_DURATION Default value for DVR Property "dvrAudioOnlyChunkTargetDuration". Value: 2000

public static final	DEFAULT_PROPERTY_BREAK_ON_PTS Default value for DVR Property "dvrChunkBreakOnPTS". Value: true
public static final	DEFAULT_PROPERTY_CHUNK_CACHE_CLASS Default value for DVR Property "dvrChunkMemoryCacheClass". Value: com.wowza.wms.dvr.impl.DvrDefaultChunkMemoryCache
public static final	DEFAULT_PROPERTY_CHUNK_DURATION_MINIMUM Default value for DVR Property "dvrChunkDurationMinimum". Value: 1500
public static final	DEFAULT_PROPERTY_CHUNK_GROUPING_SECONDS Default value for DVR Property "dvrChunkGroupingSeconds". Value: 600
public static final	DEFAULT_PROPERTY_CHUNK_READER_CLASS Default value for DVR Property "dvrChunkReaderClass". Value: com.wowza.wms.dvr.impl.io.DvrFileChunkReader
public static final	DEFAULT_PROPERTY_CHUNK_WRITER_CLASS Default value for DVR Property "dvrChunkWriterClass". Value: com.wowza.wms.dvr.impl.io.DvrFileChunkWriter
public static final	DEFAULT_PROPERTY_DEBUG_MAX_INVALID_CHUNKS_LOGGED Default value for DVR Property "dvrMaxInvalidChunksLogged". Value: 10
public static final	DEFAULT_PROPERTY_DEBUG_MAX_RAW_PACKETS Default value for DVR Property "dvrDebugMaximumRawPackets". Value: 200
public static final	DEFAULT_PROPERTY_DEBUG_MAX_VALID_CHUNKS_LOGGED Default value for DVR Property "dvrMaxValidChunksLogged". Value: 10
public static final	DEFAULT_PROPERTY_DEBUG_RAW_PACKETS Default value for DVR Property "dvrDebugRawPackets". Value: false
public static final	DEFAULT_PROPERTY_DVR_MAX_CHUNK_LOG Default value for DVR Property "dvrMaxChunkLogCount". Value: 10
public static final	DEFAULT_PROPERTY_FILE_SYSTEM_CLASS Default value for DVR Property "dvrFileSystemClass". Value: com.wowza.wms.dvr.impl.io.DvrDefaultFileSystem
public static final	DEFAULT_PROPERTY_MANIFEST_PERSISTER_CLASS Default value for DVR Property "dvrManifestPersisterClass". Value: com.wowza.wms.dvr.impl.DvrManifestPersister
public static final	DEFAULT_PROPERTY_MAX_RECALC_DURATION_LOG Default value for DVR Property "dvrMaxRecountDurationLogCount". Value: 30

public static final	DEFAULT_PROPERTY_MAX_RECORDING_LENGTH Default value for DVR Property "dvrMaximumRecordingLength". Value: 108000
public static final	DEFAULT_PROPERTY_MBR_ALTERNATIVE_MATCH_DELTA Default value for DVR Property "dvrMbrAlternateMatchDelta". Value: 1000
public static final	DEFAULT_PROPERTY_MBR_MINIMUM_PACKETTIME_GAP_SIZE Default value for DVR Property "dvrMbrMinimumPacketTimeGapSize". Value: 100
public static final	DEFAULT_PROPERTY_MBR_MINIMUM_UTCTIME_GAP_SIZE Default value for DVR Property "dvrMbrMinimumPacketTimeGapSize". Value: 750
public static final	DEFAULT_PROPERTY_MEDIACACHE_READER_CLASS Default value for DVR Property "dvrMediaCacheReaderClass". Value: com.wowza.wms.mediache.impl.MediaCacheRandomAccessReader
public static final	DEFAULT_PROPERTY_PACKET_DELTA_TO_NOTIFY Value: 200
public static final	DEFAULT_PROPERTY_PACKET_DELTA_TO_RESET_TIME Default value for DVR Property "dvrResetTimePacketDelta". Value: 200
public static final	DEFAULT_PROPERTY_PACKET_DURATION_MAXIMUM Default value for DVR Property "dvrChunkDurationMinimum". Value: 5000
public static final	DEFAULT_PROPERTY_PACKET_SORT_TIME Default value for DVR Property "dvrPacketSortTime". Value: 0
public static final	DEFAULT_PROPERTY_RECORDINGS_LOADER_CLASS Default value for DVR Property "dvrRecordingsLoaderClass". Value: com.wowza.wms.dvr.DvrRecordingsLoader
public static final	DEFAULT_PROPERTY_REPEATER_HEARTBEAT_DURATION Value: 4000
public static final	DEFAULT_PROPERTY_SANJOSE_ABST_DURATION_TOLERANCE Default value for property "dvrSanJosePlaylistAbstDurationTolerance" #see PROPERTY_SANJOSE_ABST_DURATION_TOLERANCE Value: 4
public static final	DEFAULT_PROPERTY_SANJOSE_ABST_TIMESCALE Default value for property "dvrSanJosePlaylistAbstTimescale" #see PROPERTY_SANJOSE_ABST_TIMESCALE Value: 1000
public static final	DEFAULT_PROPERTY_SMOOTH_MANIFEST_LIVE_CAN_PAUSE Value: true

public static final	<u>DEFAULT_PROPERTY_SMOOTH_MANIFEST_LIVE_CAN_SEEK</u> Value: true
public static final	<u>DEFAULT_PROPERTY_SMOOTH_MANIFEST_MAJOR_VERSION</u> Value: 2
public static final	<u>DEFAULT_PROPERTY_SMOOTH_MANIFEST_MINOR_VERSION</u> Value: 1
public static final	<u>DEFAULT_PROPERTY_SMOOTH_MANIFEST_RECORDED_CAN_PAUSE</u> Value: true
public static final	<u>DEFAULT_PROPERTY_SMOOTH_MANIFEST_RECORDED_CAN_SEEK</u> Value: true
public static final	<u>DEFAULT_PROPERTY_SMOOTH_MANIFEST_RECORDED_SPECIFY_DURATION</u> Value: true
public static final	<u>DEFAULT_PROPERTY_STORAGE_DIRECTORY</u> Default value for DVR Property "dvrStorageDirectory". Value: <code>\${com.wowza.wms.context.VHostConfigHome}/dvr</code>
public static final	<u>DEFAULT_PROPERTY_TEXT_READER_CLASS</u> Default value for DVR Property "dvrTextReaderClass". Value: <code>com.wowza.wms.dvr.impl.io.DvrTextFileReader</code>
public static final	<u>DEFAULT_PROPERTY_TEXT_WRITER_CLASS</u> Default value for DVR Property "dvrTextWriterClass". Value: <code>com.wowza.wms.dvr.impl.io.DvrTextFileWriter</code>
public static final	<u>DVR_DEFAULT_FILESTORE</u> The default DVR store ID: "dvrfilestorage". Value: <code>dvrfilestorage</code>
public static final	<u>DVR_DEFAULT_RECORDER_ID</u> The default DVR recorder ID: "dvrrecorder". Value: <code>dvrrecorder</code>
public static final	<u>DVR_REPEATER_PACKETIZER_ID</u> The default DVR streaming repeater ID: "dvrstreamingrepeater". Value: <code>dvrstreamingrepeater</code>
public static final	<u>DVR_STREAMING_PACKETIZER_ID</u> The default DVR streaming packetizer ID: "dvrstreamingpacketizer". Value: <code>dvrstreamingpacketizer</code>
public static final	<u>DVR_WINDOW_DURATION_UNLIMITED</u> Value: 0
public static final	<u>MEDIACACHE_PREFIX</u> The MediaCache prefix for dvr repeater Value: <code>dvrorigin</code>

public static final	<u>MIMETYPE_VIDEO_MP4</u> Constant for mime type "video/mp4" Value: video/mp4
public static final	<u>PROPERTY_ALLOWABLE_AV_PACKET_DELTA</u> DVR Property "dvrAllowableAVPacketDelta": for specifying how much audio and video packets may diverge before triggering an error. Value: dvrAllowableAVPacketDelta
public static final	<u>PROPERTY_APPEND_DISCONTINUITY_DELTA</u> DVR Property "dvrAppendDiscontinuityDelta": for specifying how many milliseconds of empty space are put between individual recordings that when in append mode. Value: dvrAppendDiscontinuityDelta
public static final	<u>PROPERTY_ARCHIVE_STRATEGY</u> DVR Property "dvrArchiveStrategy": for specifying the DVR archive strategy. Value: dvrArchiveStrategy
public static final	<u>PROPERTY_AUDIO_ONLY_CHUNK_TARGET_DURATION</u> DVR Property "dvrAudioOnlyChunkTargetDuration": for target duration when recording audio-only. Value: dvrAudioOnlyChunkTargetDuration
public static final	<u>PROPERTY_BREAK_ON_PTS</u> DVR Property "dvrChunkBreakOnPTS": for specifying that DVR should be broken on PTS. Value: dvrChunkBreakOnPTS
public static final	<u>PROPERTY_CHUNK_CACHE_CLASS</u> DVR Property "dvrChunkMemoryCacheClass": for controlling the class responsible for caching DVR chunks in memory. Value: dvrChunkMemoryCacheClass
public static final	<u>PROPERTY_CHUNK_DURATION_MINIMUM</u> DVR Property "dvrChunkDurationMinimum": for minimum chunk duration, in milliseconds. Value: dvrChunkDurationMinimum
public static final	<u>PROPERTY_CHUNK_GROUPING_SECONDS</u> DVR Property "dvrChunkGroupingSeconds": for determining how many seconds of DVR are stored in each stores sub-folder. Value: dvrChunkGroupingSeconds
public static final	<u>PROPERTY_CHUNK_MEMORY_CACHESIZE</u> DVR Property "chunkMemoryCacheSize": used by DvrDefaultChunkMemoryCache to set number of chunks stored in DVR in-memory cache. Value: chunkMemoryCacheSize
public static final	<u>PROPERTY_CHUNK_READER_CLASS</u> DVR Property "dvrChunkReaderClass": for controlling the class responsible for reading DVR Chunks. Value: dvrChunkReaderClass
public static final	<u>PROPERTY_CHUNK_WRITER_CLASS</u> DVR Property "dvrChunkWriterClass": for controlling the class responsible for writing DVR Chunks. Value: dvrChunkWriterClass

public static final	<u>PROPERTY_CUPERTINO_ON_CHUNK_START_RESET_COUNTER</u> DVR Property "dvrCupertinoOnChunkStartResetCounter": when a new chunk starts, reset internal tsPacketizer counters Value: dvrCupertinoOnChunkStartResetCounter
public static final	<u>PROPERTY_CUPERTINO_PLAYLIST_ALLOW_CACHING</u> DVR Property "dvrCupertinoPlaylistAllowCaching": used to force Cupertino playlist to set #EXT-X-ALLOW-CACHE: value Value: dvrCupertinoPlaylistAllowCaching
public static final	<u>PROPERTY_CUPERTINO_PLAYLIST_FORCE_LIVE</u> DVR Property "dvrCupertinoPlaylistForceLive": used to override playlist request delegate logic that determines if playlist is live. Value: dvrCupertinoPlaylistForceLive
public static final	<u>PROPERTY_CUPERTINO_PLAYLIST_FORCE_NONLIVE</u> DVR Property "dvrCupertinoPlaylistForceLive": used to override playlist request delegate logic that determines if playlist is live versus non-live. Value: dvrCupertinoPlaylistForceNonLive
public static final	<u>PROPERTY_CUPERTINO_PLAYLIST_GZIP_THRESHOLD</u> DVR Property "dvrCupertinoPlaylistGzipThreshold": when playlist is larger than this number of bytes, and gzip is enabled and accepted, the playlist will be compressed Value: dvrCupertinoPlaylistGzipThreshold
public static final	<u>PROPERTY_CUPERTINO_PLAYLIST_USE_GZIP</u> DVR Property "dvrCupertinoPlaylistUseGzip": used to force Cupertino playlist to use gzip if it is accepted Value: dvrCupertinoPlaylistUseGzip
public static final	<u>PROPERTY_CUPERTINO_PUT_END_TAG_BEFORE_CHUNKS</u> DVR Property "dvrCupertinoPutEndTagBeforeChunks": if set to true, puts the EXT-X-ENDLIST before the DVR chunks. Value: dvrCupertinoPutEndTagBeforeChunks
public static final	<u>PROPERTY_DEBUG_CHUNK_RETRIEVALS</u> DVR Property "dvrDebugChunkRetrievals": for logging each chunk retrieval Value: dvrDebugChunkRetrievals
public static final	<u>PROPERTY_DEBUG_CUPERTINO_PLAYER_ADAPTER</u> HTTP Streamer Property "dvrDebugCupertinoPlayerAdapter": for turning on DVR Player Cupertino Adapter debug logging. Value: dvrDebugCupertinoPlayerAdapter
public static final	<u>PROPERTY_DEBUG_FAILED_CHUNK_RETRIEVALS</u> DVR Property "dvrDebugFailedChunkRetrievals": for logging info about each failed chunk retrieval Value: dvrDebugFailedChunkRetrievals
public static final	<u>PROPERTY_DEBUG_LOG_INVALID_CHUNK_DETAILS</u> DVR Property "dvrLogInvalidChunkDetails": to control detailed logging information of DVR invalid chunks Value: dvrLogInvalidChunkDetails
public static final	<u>PROPERTY_DEBUG_LOG_INVALID_CHUNK_MATCHER</u> DVR Property "dvrLogInvalidChunkMatcher": for matching stream names that will log chunk packets Value: dvrLogInvalidChunkMatcher

public static final	<u>PROPERTY_DEBUG_LOG_VALID_CHUNK_DETAILS</u> DVR Property "dvrLogValidChunkDetails": to control detailed logging information of DVR valid chunks Value: dvrLogValidChunkDetails
public static final	<u>PROPERTY_DEBUG_LOG_VALID_CHUNK_MATCHER</u> DVR Property "dvrLogValidChunkMatcher": for matching stream names that will log chunk packets Value: dvrLogValidChunkMatcher
public static final	<u>PROPERTY_DEBUG_MAX_INVALID_CHUNKS_LOGGED</u> DVR Property "dvrMaxInvalidChunksLogged": for controlling maximum number of invalid DVR chunks logged. Value: dvrMaxInvalidChunksLogged
public static final	<u>PROPERTY_DEBUG_MAX_VALID_CHUNKS_LOGGED</u> DVR Property "dvrMaxValidChunksLogged": for controlling maximum number of valid DVR chunks logged. Value: dvrMaxValidChunksLogged
public static final	<u>PROPERTY_DEBUG_MAXIMUM_RAW_PACKETS</u> DVR Property "dvrDebugMaximumRawPackets": for setting maximum number of logged raw packets. Value: dvrDebugMaximumRawPackets
public static final	<u>PROPERTY_DEBUG_MBR_ALIGNMENT</u> HTTP Streamer or DVR Property "dvrDebugMbrAlignment": for turning on logging of mbr alignment Value: dvrDebugMbrAlignment
public static final	<u>PROPERTY_DEBUG_MBR_ALIGNMENT_RESOLUTION</u> DVR Property "dvrDebugMbrAlignmentResolution": for turning on logging of mbr alignment resolution (requested to actual) Value: dvrDebugMbrAlignmentResolution
public static final	<u>PROPERTY_DEBUG_MBR_DETAILS_WHEN_RECORD_DNE</u> HTTP Streamer or DVR Property "dvrDebugMbrDetailsWhenRecordDNE": for turning on DVR MBR Player Adapter debug logging. Value: dvrDebugMbrDetailsWhenRecordDNE
public static final	<u>PROPERTY_DEBUG_MBR_PLAYER_ADAPTER</u> HTTP Streamer or DVR Property "dvrDebugMbrPlayerAdapter": for turning on DVR MBR Player Adapter debug logging. Value: dvrDebugMbrPlayerAdapter
public static final	<u>PROPERTY_DEBUG_MBR_PLAYER_ADAPTER_IF_SHIFTED_EMPTY</u> HTTP Streamer or DVR Property "dvrDebugMbrPlayerAdapter": for turning on DVR MBR Player Adapter debug logging when null shifted entries is returned. Value: dvrDebugMbrPlayerAdapterWhenShiftedEmpty
public static final	<u>PROPERTY_DEBUG_METHODS</u> DVR Property "dvrDebugManagerLogMethods": for turning on DVR Manager debug logging. Value: dvrDebugManagerLogMethods
public static final	<u>PROPERTY_DEBUG_PLAYER_ADAPTER</u> HTTP Streamer Property "dvrDebugPlayerAdapter": for turning on DVR Player Adapter debug logging for all streamer types. Value: dvrDebugPlayerAdapter

public static final	<u>PROPERTY_DEBUG_PLAYLIST_REQUEST</u> DVR Property "dvrDebugPlaylistRequest": for turning on logging of DVR playlist requests. Value: dvrDebugPlaylistRequest
public static final	<u>PROPERTY_DEBUG_RAW_PACKETS</u> DVR Property "dvrDebugRawPackets": for turning on logging of incoming raw packets. Value: dvrDebugRawPackets
public static final	<u>PROPERTY_DEBUG_RAW_PACKETS_MATCHER</u> DVR Property "dvrDebugRawPacketsMatcher": for matching stream names that will dump raw packet. Value: dvrDebugRawPacketsMatcher
public static final	<u>PROPERTY_DEBUG_REPEATER</u> DVR Property "dvrDebugRepeater": for turning on logging of DVR repeater Value: dvrDebugRepeater
public static final	<u>PROPERTY_DEBUG_SANJOSE_PLAYER_ADAPTER</u> HTTP Streamer Property "dvrDebugSanJosePlayerAdapter": for turning on DVR Player San Jose Adapter debug logging. Value: dvrDebugSanJosePlayerAdapter
public static final	<u>PROPERTY_DEBUG_SMOOTH_PLAYER_ADAPTER</u> HTTP Streamer Property "dvrDebugSmoothPlayerAdapter": for turning on DVR Player Smooth Adapter debug logging. Value: dvrDebugSmoothPlayerAdapter
public static final	<u>PROPERTY_DEBUG_STATE_CHANGE</u> DVR Property "dvrDebugStateChange": for logging state changes of DVR store. Value: dvrDebugStateChange
public static final	<u>PROPERTY_DEBUG_TOSSED_HOLDERS</u> DVR Property "dvrDebugTossedHolders": for turning on logging of packets that are being tossed. Value: dvrDebugTossedHolders
public static final	<u>PROPERTY_ENCRYPTION_INFO_DELEGATE</u> DVR Property "dvrPlaylistEncryptionInfoDelegate": used to over-ride the encryption info on the playback side. Value: dvrPlaylistEncryptionInfoDelegate
public static final	<u>PROPERTY_FILE_SYSTEM_CLASS</u> DVR Property "dvrFileSystemClass": for controlling the class responsible for managing the DVR File System. Value: dvrFileSystemClass
public static final	<u>PROPERTY_MANIFEST_PERSISTER_CLASS</u> DVR Property "dvrManifestPersisterClass": for controlling the class responsible for persisting the manifest files. Value: dvrManifestPersisterClass
public static final	<u>PROPERTY_MAX_CHUNK_LOG</u> DVR Property "dvrMaxChunkLogCount": for maximum number of DVR chunks to log. Value: dvrMaxChunkLogCount

public static final	<u>PROPERTY_MAX_RECALC_DURATION_LOG</u> DVR Property "dvrMaxRecountDurationLogCount": for maximum number of messages to log regarding recalculating DVR chunk duration Value: dvrMaxRecountDurationLogCount
public static final	<u>PROPERTY_MAX_RECORDING_LENGTH</u> DVR Property "dvrMaximumRecordingLength": The maximum recording length in seconds. Value: dvrMaximumRecordingLength
public static final	<u>PROPERTY_MBR_ALTERNATIVE_MATCH_DELTA</u> DVR Property "dvrMbrAlternateMatchDelta": when finding alternative for a MBR gap, an alternate is considered a match is within this many ms of the gap in question. Value: dvrMbrAlternateMatchDelta
public static final	<u>PROPERTY_MBR_MINIMUM_PACKETTIME_GAP_SIZE</u> DVR Property "dvrMbrMinimumPacketTimeGapSize": when doing mbr alignment gaps smaller than this are ignored. Value: dvrMbrMinimumPacketTimeGapSize
public static final	<u>PROPERTY_MBR_MINIMUM_UTCTIME_GAP_SIZE</u> DVR Property "dvrMbrMinimumUtcTimeGapSize": when doing mbr alignment gaps smaller than this are ignored. Value: dvrMbrMinimumUtcTimeGapSize
public static final	<u>PROPERTY_MBR_USE_UTC_FOR_ALIGNMENT</u> DVR Property "dvrMbrUseUtcForAlignment": when doing mbr alignment use utc time for alignment Value: dvrMbrUseUtcForAlignment
public static final	<u>PROPERTY_MEDIACACHE_ENABLED</u> DVR Property "dvrMediaCacheEnabled" Value: dvrMediaCacheEnabled
public static final	<u>PROPERTY_MEDIACACHE_READER_CLASS</u> DVR Property "dvrMediaCacheReaderClass" Value: dvrMediaCacheReaderClass
public static final	<u>PROPERTY_PACKET_DELTA_TO_NOTIFY</u> Value: dvrPacketDeltaToNotify
public static final	<u>PROPERTY_PACKET_DELTA_TO_RESET_TIME</u> DVR Property "dvrResetTimePacketDelta": for controlling how much Chunk start times must diverge from last chunk's end time before the DVR time gets reset. Value: dvrResetTimePacketDelta
public static final	<u>PROPERTY_PACKET_DURATION_MAXIMUM</u> DVR Property "dvrPacketDurationMaximum": for maximum packet duration, in milliseconds. Value: dvrPacketDurationMaximum
public static final	<u>PROPERTY_PACKET_SORT_TIME</u> DVR Property "dvrPacketSortTime": for sorting incoming packets. Value: dvrPacketSortTime

public static final	PROPERTY_PLAYLIST_REQUEST_DELEGATE DVR Property "dvrPlaylistRequestDelegate": used to over-ride the class that generates a playlist request. Value: dvrPlaylistRequestDelegate
public static final	PROPERTY_RECORD_AUDIO DVR Property "recordAudio": for determining if audio should be recorded. Value: recordAudio
public static final	PROPERTY_RECORD_DATA DVR Property "recordData": for determining if data should be recorded. Value: recordData
public static final	PROPERTY_RECORD_VIDEO DVR Property "recordVideo": for determining if video should be recorded. Value: recordVideo
public static final	PROPERTY_RECORDINGS_LOADER_CLASS DVR Property "dvrRecordingsLoaderClass": for controlling the class responsible for loading DVR recordings. Value: dvrRecordingsLoaderClass
public static final	PROPERTY_REPEATER_HEARTBEAT_DURATION DVR Property "dvrRepeaterHeartbeatDuration": for time in ms that origin pings edges Value: dvrRepeaterHeartbeatDuration
public static final	PROPERTY_REPEATER_SHARED_SECRET DVR Property "dvrEncryptionSharedSecret": for encryption shared secret between origins and edges. Value: dvrEncryptionSharedSecret
public static final	PROPERTY_SANJOSE_ABST_DURATION_TOLERANCE DVR Property "dvrSanJosePlaylistAbstDurationEqualityTolerance": used to over-ride the tolerance when determining equal chunks lengths. Value: dvrSanJosePlaylistAbstDurationTolerance
public static final	PROPERTY_SANJOSE_ABST_TIMESCALE DVR Property "dvrSanJosePlaylistAbstTimescale": used to over-ride the time-scale for abst files. Value: dvrSanJosePlaylistAbstTimescale
public static final	PROPERTY_SANJOSE_PLAYLIST_DELIVERYTYPE DVR Property "dvrSanJosePlaylistDeliveryType": used to over-ride the mime type for DVR San Jose F4m playlists. Value: dvrSanJosePlaylistDeliveryType
public static final	PROPERTY_SANJOSE_PLAYLIST_LIVE_STREAMTYPE DVR Property "dvrSanJosePlaylistLiveStreamType": used to over-ride the stream type for live DVR in San Jose f4m playlists. Value: dvrSanJosePlaylistLiveStreamType
public static final	PROPERTY_SANJOSE_PLAYLIST_MIMETYPE DVR Property "dvrSanJosePlaylistMimeType": used to over-ride the mime type for DVR San Jose F4m playlists. Value: dvrSanJosePlaylistMimeType

public static final	<u>PROPERTY_SANJOSE_PLAYLIST_RECORDED_STREAMTYPE</u> DVR Property "dvrSanJosePlaylistRecordedStreamType": used to over-ride the stream type for non-live (recorded) DVR in San Jose f4m playlists. Value: dvrSanJosePlaylistRecordedStreamType
public static final	<u>PROPERTY_SANJOSE_PLAYLIST_VERSION</u> DVR Property "dvrSanJosePlaylistVersion": used to over-ride the version of the San Jose f4m playlist. Value: dvrSanJosePlaylistVersion
public static final	<u>PROPERTY_SMOOTH_MANIFEST_H264_CODEC</u> DVR Property "dvrSmoothManifestH264Codec": used to force Smooth Manifest to use this as its FourCC H264 codec info. Value: dvrSmoothManifestH264Codec
public static final	<u>PROPERTY_SMOOTH_MANIFEST_LIVE_CAN_PAUSE</u> DVR Property "dvrSmoothManifestLiveCanPause": used to determine is CanPause is enabled in smooth Manifest. Value: dvrSmoothManifestLiveCanPause
public static final	<u>PROPERTY_SMOOTH_MANIFEST_LIVE_CAN_SEEK</u> DVR Property "dvrSmoothManifestLiveCanSeek": used to determine is CanSeek is enabled in smooth Manifest. Value: dvrSmoothManifestLiveCanSeek
public static final	<u>PROPERTY_SMOOTH_MANIFEST_MAJOR_VERSION</u> DVR Property "dvrSmoothManifestMajorVersion": used to determine Smooth manifest major version Value: dvrSmoothManifestMajorVersion
public static final	<u>PROPERTY_SMOOTH_MANIFEST_MINOR_VERSION</u> DVR Property "dvrSmoothManifestMinorVersion": used to determine Smooth manifest major version Value: dvrSmoothManifestMinorVersion
public static final	<u>PROPERTY_SMOOTH_MANIFEST_RECORDED_CAN_PAUSE</u> DVR Property "dvrSmoothManifestRecordedCanPause": used to determine is CanPause is enabled in smooth Manifest. Value: dvrSmoothManifestRecordedCanPause
public static final	<u>PROPERTY_SMOOTH_MANIFEST_RECORDED_CAN_SEEK</u> DVR Property "dvrSmoothManifestRecordedCanSeek": used to determine is CanSeek is enabled in smooth Manifest. Value: dvrSmoothManifestRecordedCanSeek
public static final	<u>PROPERTY_SMOOTH_MANIFEST_RECORDED_SPECIFY_DURATION</u> DVR Property "dvrSmoothManifestRecordedSpecifyDuration": used to determine if duration is specified. Value: dvrSmoothManifestRecordedSpecifyDuration
public static final	<u>PROPERTY_SMOOTH_MANIFEST_VERBOSE_DURATION</u> DVR Property "dvrSmoothManifestVerboseDuration": used to force Smooth Manifest to include durations for each record Value: dvrSmoothManifestVerboseDuration
public static final	<u>PROPERTY_START_RECORDING_ON_STARTUP</u> DVR Property "startRecordingOnStartup": for determining if DVR recorder should start recording immediately. Value: startRecordingOnStartup

public static final	<u>PROPERTY_STORAGE_DIRECTORY</u> DVR Property "dvrStorageDirectory": for overriding the DVR storage directory location Typically this is defined application-wide in Application.xml under Application/DVR/StorageDirectory. Value: dvrStorageDirectory
public static final	<u>PROPERTY_STREAM_DATA</u> Value: dvrStreamRecordedData
public static final	<u>PROPERTY_TEXT_READER_CLASS</u> DVR Property "dvrTextReaderClass": for controlling the class responsible for reading DVR text files. Value: dvrTextReaderClass
public static final	<u>PROPERTY_TEXT_WRITER_CLASS</u> DVR Property "dvrTextWriterClass": for controlling the class responsible for writing DVR text files. Value: dvrTextWriterClass
public static final	<u>PROPERTY_WINDOW_DURATION</u> DVR Property "dvrWindowDuration": for specifying the DVR window duration, in seconds. Value: dvrWindowDuration
public static final	<u>SANJOSE_F4M_STREAMINGTYPE_STREAMING</u> Constant for San Jose streaming type "streaming" Value: streaming
public static final	<u>SANJOSE_F4M_STREAMTYPE_DVR</u> Constant for San Jose stream type "dvr". Value: dvr
public static final	<u>SANJOSE_F4M_STREAMTYPE_LIVE</u> Constant for San Jose stream type "live". Value: live
public static final	<u>SANJOSE_F4M_STREAMTYPE_LIVEORRECORDED</u> Constant for San Jose stream type "liveOrRecorded". Value: liveOrRecorded
public static final	<u>SANJOSE_F4M_STREAMTYPE_RECORDED</u> Constant for San Jose stream type "recorded". Value: recorded
public static final	<u>SANJOSE_F4M_VERSION_1_0</u> Constant for San Jose f4m version "1.0". Value: 1.0
public static final	<u>SANJOSE_F4M_VERSION_2_0</u> Constant for San Jose f4m version "2.0". Value: 2.0

Fields inherited from interface [com.wowza.wms.dvr.IDvrPrivateConstants](#)

[CHUNK_FILENAME_FORMAT_AUDIO](#), [CHUNK_FILENAME_FORMAT_DATA](#), [CHUNK_FILENAME_FORMAT_METADATA](#),
[CHUNK_FILENAME_FORMAT_VIDEO](#), [CHUNK_FOLDER_FORMAT](#), [DEFAULT_PROPERTY_AUDIO_GROUP_COUNT](#),
[DEFAULT_PROPERTY_MANIFEST_PURGE_SIZE](#), [DEFAULT_PROPERTY_MAX_ALLOWABLE_CHUNK_DURATION](#),
[DEFAULT_PROPERTY_MAX_SAVE HOLDER_SIZE](#), [DEFAULT_PROPERTY_MINIMUM_AVAILABLE_CHUNKS](#),
[DEFAULT_PROPERTY_PURGE_CONTROL_CLASS](#),
[DEFAULT_PROPERTY_RECALC_DURATION_MAX_AV_DIFFERENCE_TRIGGER](#),
[DEFAULT_PROPERTY_RECALC_DURATION_TRIGGER_SIZE](#), [DEFAULT_PROPERTY_STREAM_IDLE_TIMEOUT](#),
[DEFAULT_PROPERTY_STREAM_STARTUP_TIMEOUT](#), [DEFAULT_PROPERTY_UTC_ALIGN_TO_AUDIO](#),
[DEFAULT_PROPERTY_UTC_USE_WOWZA_PACKET_ARRIVAL](#), [DEFAULT_PROPERTY_WAIT_FOR_CODEC_TIME](#),
[PROPERTY_AUDIO_GROUP_COUNT](#), [PROPERTY_DEBUG_MBR_RESOLVER](#),
[PROPERTY_DEBUG_MBR_RESOLVER_GAP_DETAILS](#),
[PROPERTY_DEBUG_MBR_RESOLVER_INTERSECTION_DETAILS](#), [PROPERTY_INTERCEPTOR_CHUNK_DELEGATE](#),
[PROPERTY_MANIFEST_PURGE_CACHE_SIZE](#), [PROPERTY_MAX_ALLOWABLE_CHUNK_DURATION](#),
[PROPERTY_MAX_SAVE HOLDER_SIZE](#), [PROPERTY_MBR_USE_SIMPLEALIGNMENT](#),
[PROPERTY_MINIMUM_AVAILABLE_CHUNKS](#), [PROPERTY_PURGE_CONTROL_CLASS](#),
[PROPERTY_RECALC_DURATION_MAX_AV_DIFFERENCE_TRIGGER](#),
[PROPERTY_RECALC_DURATION_TRIGGER_SIZE](#), [PROPERTY_UTC_ALIGN_TO_AUDIO](#),
[PROPERTY_UTC_DELEGATE_CLASS](#), [PROPERTY_UTC_USE_WOWZA_PACKET_ARRIVAL](#),
[PROPERTY_VERIFY_CHUNK_EXISTENCE_ON_MANIFEST_LOAD](#), [PROPERTY_WAIT_FOR_CODEC_TIME](#)

Fields

DVR_STREAMING_PACKETIZER_ID

```
public static final java.lang.String DVR_STREAMING_PACKETIZER_ID
```

The default DVR streaming packetizer ID: "dvrstreamingpacketizer".

See DVR.xml and Application.xml.

Constant value: **dvrstreamingpacketizer**

DVR_REPEATER_PACKETIZER_ID

```
public static final java.lang.String DVR_REPEATER_PACKETIZER_ID
```

The default DVR streaming repeater ID: "dvrstreamingrepeater".

See DVR.xml and Application.xml.

Constant value: **dvrstreamingrepeater**

DVR_DEFAULT_RECORDER_ID

```
public static final java.lang.String DVR_DEFAULT_RECORDER_ID
```

The default DVR recorder ID: "dvrrecorder".

See DVR.xml and Application.xml.

Constant value: **dvrrecorder**

DVR_DEFAULT_FILESTORE

```
public static final java.lang.String DVR_DEFAULT_FILESTORE
```

The default DVR store ID: "dvrfilestorage".

See DVR.xml and Application.xml.

Constant value: **dvrfilestorage**

MEDIACACHE_PREFIX

```
public static final java.lang.String MEDIACACHE_PREFIX
```

The MediaCache prefix for dvr repeater

See MediaCache.xml

Constant value: **dvrorigin**

PROPERTY_CHUNK_GROUPING_SECONDS

```
public static final java.lang.String PROPERTY_CHUNK_GROUPING_SECONDS
```

DVR Property "dvrChunkGroupingSeconds": for determining how many seconds of DVR are stored in each stores sub-folder.

The directory naming convention is HHHH_MM_SS with H = hours, M = minutes, S = seconds. Using the defaults, first directory would be named 0000_00_00 and the second directory would be named 0000_10_00, etc.

Valid values are integers greater than 60. Add the property to Application/DVR/Properties section of Application.xml.

Constant value: **dvrChunkGroupingSeconds**

See Also:

[DEFAULT_PROPERTY_CHUNK_GROUPING_SECONDS](#)

DEFAULT_PROPERTY_CHUNK_GROUPING_SECONDS

```
public static final int DEFAULT_PROPERTY_CHUNK_GROUPING_SECONDS
```

Default value for DVR Property "dvrChunkGroupingSeconds".

Default value is 600 seconds (10 minutes).

Constant value: **600**

See Also:

[PROPERTY_CHUNK_GROUPING_SECONDS](#)

PROPERTY_APPEND_DISCONTINUITY_DELTA

```
public static final java.lang.String PROPERTY_APPEND_DISCONTINUITY_DELTA
```

DVR Property "dvrAppendDiscontinuityDelta": for specifying how many milliseconds of empty space are put between individual recordings that when in append mode.

Valid values are integers greater than or equal to 0. Add the property to Application/DVR/Properties section of Application.xml.

Constant value: **dvrAppendDiscontinuityDelta**

See Also:

[DEFAULT_PROPERTY_APPEND_DISCONTINUITY_DELTA](#)

DEFAULT_PROPERTY_APPEND_DISCONTINUITY_DELTA

```
public static final int DEFAULT_PROPERTY_APPEND_DISCONTINUITY_DELTA
```

Default value for DVR Property "dvrAppendDiscontinuityDelta".

Default value is 0.

Constant value: **0**

See Also:

(continued from last page)

[PROPERTY_APPEND_DISCONTINUITY_DELTA](#)

PROPERTY_WINDOW_DURATION

```
public static final java.lang.String PROPERTY_WINDOW_DURATION
```

DVR Property "dvrWindowDuration": for specifying the DVR window duration, in seconds.

Typically this is defined application-wide in Application.xml under Application/DVR/WindowDuration. However, to override this on a per stream basis, the property may be set on the IDvrStreamManager after creation and before initialization. Valid values are integers greater than or equal to 0. Add the property to Application/DVR/Properties section of Application.xml. The default is 0, meaning the window size is unlimited.

Constant value: **dvrWindowDuration**

See Also:

[DVR_WINDOW_DURATION_UNLIMITED](#)

DVR_WINDOW_DURATION_UNLIMITED

```
public static final int DVR_WINDOW_DURATION_UNLIMITED
```

Constant value: **0**

PROPERTY_STORAGE_DIRECTORY

```
public static final java.lang.String PROPERTY_STORAGE_DIRECTORY
```

DVR Property "dvrStorageDirectory": for overriding the DVR storage directory location. Typically this is defined application-wide in Application.xml under Application/DVR/StorageDirectory. However, to override this on a per stream basis, the property may be set on the IDvrStreamManager after creation and before initialization.

Constant value: **dvrStorageDirectory**

See Also:

[DEFAULT_PROPERTY_STORAGE_DIRECTORY](#)

DEFAULT_PROPERTY_STORAGE_DIRECTORY

```
public static final java.lang.String DEFAULT_PROPERTY_STORAGE_DIRECTORY
```

Default value for DVR Property "dvrStorageDirectory".

Default value is "\${com.wowza.wms.context.VHostConfigHome}/dvr".

Constant value: **\${com.wowza.wms.context.VHostConfigHome}/dvr**

See Also:

[PROPERTY_STORAGE_DIRECTORY](#)

PROPERTY_ARCHIVE_STRATEGY

```
public static final java.lang.String PROPERTY_ARCHIVE_STRATEGY
```

DVR Property "dvrArchiveStrategy": for specifying the DVR archive strategy.

Typically this is defined application-wide in Application.xml under Application/DVR/ArchiveStrategy. However, to override this on a per stream basis, the property may be set on the IDvrStreamManager after creation and before initialization. Valid values are:

Constant value: **dvrArchiveStrategy**

See Also:

[ARCHIVE_STRATEGY_APPEND](#)

(continued from last page)

[ARCHIVE_STRATEGY_VERSION](#)[ARCHIVE_STRATEGY_APPEND](#)

ARCHIVE_STRATEGY_DELETE

```
public static final java.lang.String ARCHIVE_STRATEGY_DELETE
```

Constant value: **delete**

ARCHIVE_STRATEGY_VERSION

```
public static final java.lang.String ARCHIVE_STRATEGY_VERSION
```

Constant value: **version**

ARCHIVE_STRATEGY_APPEND

```
public static final java.lang.String ARCHIVE_STRATEGY_APPEND
```

Constant value: **append**

PROPERTY_CHUNK_READER_CLASS

```
public static final java.lang.String PROPERTY_CHUNK_READER_CLASS
```

DVR Property "dvrChunkReaderClass": for controlling the class responsible for reading DVR Chunks.

The class must implement IDvrChunkReader

Add the property to Application/DVR/Properties section of Application.xml.

Constant value: **dvrChunkReaderClass**

See Also:

`com.wowza.wms.dvr.io.IDvrChunkReader`

[DEFAULT_PROPERTY_CHUNK_READER_CLASS](#)

DEFAULT_PROPERTY_CHUNK_READER_CLASS

```
public static final java.lang.String DEFAULT_PROPERTY_CHUNK_READER_CLASS
```

Default value for DVR Property "dvrChunkReaderClass".

Default value is "com.wowza.wms.dvr.impl.io.DvrFileChunkReader".

Constant value: **com.wowza.wms.dvr.impl.io.DvrFileChunkReader**

See Also:

[PROPERTY_CHUNK_READER_CLASS](#)

PROPERTY_CHUNK_WRITER_CLASS

```
public static final java.lang.String PROPERTY_CHUNK_WRITER_CLASS
```

DVR Property "dvrChunkWriterClass": for controlling the class responsible for writing DVR Chunks.

The class must implement IDvrChunkWriter

Add the property to Application/DVR/Properties section of Application.xml.

Constant value: **dvrChunkWriterClass**

(continued from last page)

See Also:

`com.wowza.wms.dvr.io.IDvrChunkWriter`
[DEFAULT_PROPERTY_CHUNK_WRITER_CLASS](#)

DEFAULT_PROPERTY_CHUNK_WRITER_CLASS

```
public static final java.lang.String DEFAULT_PROPERTY_CHUNK_WRITER_CLASS
```

Default value for DVR Property "dvrChunkWriterClass".

Default value is "com.wowza.wms.dvr.impl.io.DvrFileChunkWriter".

Constant value: **com.wowza.wms.dvr.impl.io.DvrFileChunkWriter**

See Also:

[PROPERTY_CHUNK_WRITER_CLASS](#)

PROPERTY_FILE_SYSTEM_CLASS

```
public static final java.lang.String PROPERTY_FILE_SYSTEM_CLASS
```

DVR Property "dvrFileSystemClass": for controlling the class responsible for managing the DVR File System.

The class must implement `com.wowza.wms.dvr.io.IDvrFileSystem` and may sub-class `com.wowza.wms.dvr.impl.io.DvrDefaultFileSystem`

Add the property to Application/DVR/Properties section of Application.xml.

Constant value: **dvrFileSystemClass**

See Also:

`com.wowza.wms.dvr.io.IDvrFileSystem`
[DEFAULT_PROPERTY_FILE_SYSTEM_CLASS](#)

DEFAULT_PROPERTY_FILE_SYSTEM_CLASS

```
public static final java.lang.String DEFAULT_PROPERTY_FILE_SYSTEM_CLASS
```

Default value for DVR Property "dvrFileSystemClass".

Default value is "com.wowza.wms.dvr.impl.io.DvrDefaultFileSystem".

Constant value: **com.wowza.wms.dvr.impl.io.DvrDefaultFileSystem**

See Also:

[PROPERTY_FILE_SYSTEM_CLASS](#)

PROPERTY_MANIFEST_PERSISTER_CLASS

```
public static final java.lang.String PROPERTY_MANIFEST_PERSISTER_CLASS
```

DVR Property "dvrManifestPersisterClass": for controlling the class responsible for persisting the manifest files.

The class must implement `com.wowza.wms.dvr.io.IDvrManifestPersister`

Add the property to Application/DVR/Properties section of Application.xml.

Constant value: **dvrManifestPersisterClass**

See Also:

`com.wowza.wms.dvr.io.IDvrManifestPersister`
[DEFAULT_PROPERTY_MANIFEST_PERSISTER_CLASS](#)

(continued from last page)

DEFAULT_PROPERTY_MANIFEST_PERSISTER_CLASS

```
public static final java.lang.String DEFAULT_PROPERTY_MANIFEST_PERSISTER_CLASS
```

Default value for DVR Property "dvrManifestPersisterClass".

Default value is "com.wowza.wms.dvr.impl.DvrManifestPersister".

Constant value: **com.wowza.wms.dvr.impl.DvrManifestPersister**

See Also:

[PROPERTY_MANIFEST_PERSISTER_CLASS](#)

PROPERTY_CHUNK_CACHE_CLASS

```
public static final java.lang.String PROPERTY_CHUNK_CACHE_CLASS
```

DVR Property "dvrChunkMemoryCacheClass": for controlling the class responsible for caching DVR chunks in memory.

The class must implement [IDvrChunkMemoryCache](#) and may sub-class `com.wowza.wms.dvr.impl.DvrDefaultChunkMemoryCache`

Add the property to Application/DVR/Properties section of Application.xml.

Constant value: **dvrChunkMemoryCacheClass**

See Also:

[IDvrChunkMemoryCache](#)

[DEFAULT_PROPERTY_CHUNK_CACHE_CLASS](#)

DEFAULT_PROPERTY_CHUNK_CACHE_CLASS

```
public static final java.lang.String DEFAULT_PROPERTY_CHUNK_CACHE_CLASS
```

Default value for DVR Property "dvrChunkMemoryCacheClass".

Default value is "com.wowza.wms.dvr.impl.DvrDefaultChunkMemoryCache".

Constant value: **com.wowza.wms.dvr.impl.DvrDefaultChunkMemoryCache**

See Also:

[PROPERTY_CHUNK_CACHE_CLASS](#)

PROPERTY_RECORDINGS_LOADER_CLASS

```
public static final java.lang.String PROPERTY_RECORDINGS_LOADER_CLASS
```

DVR Property "dvrRecordingsLoaderClass": for controlling the class responsible for loading DVR recordings.

The class must implement [IDvrRecordingsLoader](#)

Add the property to Application/DVR/Properties section of Application.xml.

Constant value: **dvrRecordingsLoaderClass**

See Also:

[IDvrRecordingsLoader](#)

[DEFAULT_PROPERTY_RECORDINGS_LOADER_CLASS](#)

DEFAULT_PROPERTY_RECORDINGS_LOADER_CLASS

```
public static final java.lang.String DEFAULT_PROPERTY_RECORDINGS_LOADER_CLASS
```

Default value for DVR Property "dvrRecordingsLoaderClass".

Default value is "com.wowza.wms.dvr.DvrRecordingsLoader".

(continued from last page)

Constant value: **com.wowza.wms.dvr.DvrRecordingsLoader**

See Also:

[PROPERTY_RECORDINGS_LOADER_CLASS](#)

PROPERTY_TEXT_WRITER_CLASS

```
public static final java.lang.String PROPERTY_TEXT_WRITER_CLASS
```

DVR Property "dvrTextWriterClass": for controlling the class responsible for writing DVR text files.

The class must implement [IDvrTextWriter](#)

Add the property to Application/DVR/Properties section of Application.xml.

Constant value: **dvrTextWriterClass**

See Also:

[IDvrTextWriter](#)

[DEFAULT_PROPERTY_TEXT_WRITER_CLASS](#)

DEFAULT_PROPERTY_TEXT_WRITER_CLASS

```
public static final java.lang.String DEFAULT_PROPERTY_TEXT_WRITER_CLASS
```

Default value for DVR Property "dvrTextWriterClass".

Default value is "com.wowza.wms.dvr.impl.io.DvrTextFileWriter".

Constant value: **com.wowza.wms.dvr.impl.io.DvrTextFileWriter**

See Also:

[PROPERTY_TEXT_WRITER_CLASS](#)

PROPERTY_TEXT_READER_CLASS

```
public static final java.lang.String PROPERTY_TEXT_READER_CLASS
```

DVR Property "dvrTextReaderClass": for controlling the class responsible for reading DVR text files.

The class must implement [IDvrTextReader](#)

Add the property to Application/DVR/Properties section of Application.xml.

Constant value: **dvrTextReaderClass**

See Also:

[IDvrTextReader](#)

[DEFAULT_PROPERTY_TEXT_READER_CLASS](#)

DEFAULT_PROPERTY_TEXT_READER_CLASS

```
public static final java.lang.String DEFAULT_PROPERTY_TEXT_READER_CLASS
```

Default value for DVR Property "dvrTextReaderClass".

Default value is "com.wowza.wms.dvr.impl.io.DvrTextFileReader".

Constant value: **com.wowza.wms.dvr.impl.io.DvrTextFileReader**

See Also:

[PROPERTY_TEXT_READER_CLASS](#)

(continued from last page)

PROPERTY_ALLOWABLE_AV_PACKET_DELTA

```
public static final java.lang.String PROPERTY_ALLOWABLE_AV_PACKET_DELTA
```

DVR Property "dvrAllowableAVPacketDelta": for specifying how much audio and video packets may diverge before triggering an error.

Wowza nDVR expects the incoming audio and video to be aligned. Through this parameter, Wowza nDVR will try to compensate for out of alignment issues, but it cannot resolve them. This setting safeguards against audio and data video packets that are not closely aligned. The units are in ms. The default value is 2000 ms. If audio and video are out of alignment by more than this value, the audio and video chunks will be ignored in an attempt to get the streams back into alignment. If you already have an out of alignment issue, increasing the default value is likely to cause more problems. Increasing this value will increase the number of chunks not recorded which may cause the overall quality to be unacceptable.

Add the property to Application/DVR/Properties section of Application.xml.
Constant value: **dvrAllowableAVPacketDelta**

See Also:

[DEFAULT_PROPERTY_ALLOWABLE_AV_PACKET_DELTA](#)

DEFAULT_PROPERTY_ALLOWABLE_AV_PACKET_DELTA

```
public static final int DEFAULT_PROPERTY_ALLOWABLE_AV_PACKET_DELTA
```

Default value for DVR Property "dvrAllowableAVPacketDelta".

Default value is 2000 ms.
Constant value: **2000**

See Also:

[PROPERTY_ALLOWABLE_AV_PACKET_DELTA](#)

PROPERTY_PACKET_DELTA_TO_RESET_TIME

```
public static final java.lang.String PROPERTY_PACKET_DELTA_TO_RESET_TIME
```

DVR Property "dvrResetTimePacketDelta": for controlling how much Chunk start times must diverge from last chunk's end time before the DVR time gets reset.

Add the property to Application/DVR/Properties section of Application.xml
Constant value: **dvrResetTimePacketDelta**

See Also:

[DEFAULT_PROPERTY_PACKET_DELTA_TO_RESET_TIME](#)

DEFAULT_PROPERTY_PACKET_DELTA_TO_RESET_TIME

```
public static final int DEFAULT_PROPERTY_PACKET_DELTA_TO_RESET_TIME
```

Default value for DVR Property "dvrResetTimePacketDelta".

Default value is 200 ms.
Constant value: **200**

See Also:

[PROPERTY_PACKET_DELTA_TO_RESET_TIME](#)

PROPERTY_PACKET_DELTA_TO_NOTIFY

```
public static final java.lang.String PROPERTY_PACKET_DELTA_TO_NOTIFY
```

(continued from last page)

Constant value: **dvrPacketDeltaToNotify**

DEFAULT_PROPERTY_PACKET_DELTA_TO_NOTIFY

```
public static final int DEFAULT_PROPERTY_PACKET_DELTA_TO_NOTIFY
```

Constant value: **200**

PROPERTY_RECORD_DATA

```
public static final java.lang.String PROPERTY_RECORD_DATA
```

DVR Property "recordData": for determining if data should be recorded.

Set property to: true to record data of incoming stream. Default is true. Set property to: false to ignore data packets during recording.

Constant value: **recordData**

PROPERTY_RECORD_VIDEO

```
public static final java.lang.String PROPERTY_RECORD_VIDEO
```

DVR Property "recordVideo": for determining if video should be recorded.

Set property to: true to record video of incoming stream. Default is true. Set property to: false to ignore video packets during recording. Either recordVideo or recordAudio property must be set to: true

Constant value: **recordVideo**

PROPERTY_RECORD_AUDIO

```
public static final java.lang.String PROPERTY_RECORD_AUDIO
```

DVR Property "recordAudio": for determining if audio should be recorded.

Set property to: true to record audio of incoming stream. Default is true. Set property to: false to ignore audio packets during recording. Either recordVideo or recordAudio property must be set to: true

Constant value: **recordAudio**

PROPERTY_START_RECORDING_ON_STARTUP

```
public static final java.lang.String PROPERTY_START_RECORDING_ON_STARTUP
```

DVR Property "startRecordingOnStartup": for determining if DVR recorder should start recording immediately.

Set property to: true (default) to start recording immediately when stream is detected. Set property to: false to init the DVR recorder but to not start recording immediately when stream starts.

Constant value: **startRecordingOnStartup**

PROPERTY_AUDIO_ONLY_CHUNK_TARGET_DURATION

```
public static final java.lang.String PROPERTY_AUDIO_ONLY_CHUNK_TARGET_DURATION
```

DVR Property "dvrAudioOnlyChunkTargetDuration": for target duration when recording audio-only.

Applies to recording an audio-only stream and is the target chunk duration. The unit is in ms. The default value is 2000 ms. This setting is ignored if the stream contains video and audio, in which case the keyframe determines the chunk size.

Add the property to Application/DVR/Properties section of Application.xml

Constant value: **dvrAudioOnlyChunkTargetDuration**

See Also:

(continued from last page)

[DEFAULT_PROPERTY_AUDIO_ONLY_CHUNK_TARGET_DURATION](#)

DEFAULT_PROPERTY_AUDIO_ONLY_CHUNK_TARGET_DURATION

```
public static final int DEFAULT_PROPERTY_AUDIO_ONLY_CHUNK_TARGET_DURATION
```

Default value for DVR Property "dvrAudioOnlyChunkTargetDuration".

Default value is 2000 ms.

Constant value: **2000**

See Also:

[PROPERTY_AUDIO_ONLY_CHUNK_TARGET_DURATION](#)

PROPERTY_PACKET_SORT_TIME

```
public static final java.lang.String PROPERTY_PACKET_SORT_TIME
```

DVR Property "dvrPacketSortTime": for sorting incoming packets. Specified in milliseconds.

Valid values are a integer greater than equal to 0. Zero means no sorting occurs. Add the property to Application/DVR/Properties section of Application.xml

Constant value: **dvrPacketSortTime**

See Also:

[DEFAULT_PROPERTY_PACKET_SORT_TIME](#)

DEFAULT_PROPERTY_PACKET_SORT_TIME

```
public static final int DEFAULT_PROPERTY_PACKET_SORT_TIME
```

Default value for DVR Property "dvrPacketSortTime".

Default value is 0 ms (i.e. no sorting).

Constant value: **0**

See Also:

[PROPERTY_PACKET_SORT_TIME](#)

PROPERTY_BREAK_ON_PTS

```
public static final java.lang.String PROPERTY_BREAK_ON_PTS
```

DVR Property "dvrChunkBreakOnPTS": for specifying that DVR should be broken on PTS.

Valid values are "true" or "false" Add the property to Application/DVR/Properties section of Application.xml

Constant value: **dvrChunkBreakOnPTS**

See Also:

[DEFAULT_PROPERTY_BREAK_ON_PTS](#)

DEFAULT_PROPERTY_BREAK_ON_PTS

```
public static final boolean DEFAULT_PROPERTY_BREAK_ON_PTS
```

Default value for DVR Property "dvrChunkBreakOnPTS".

Default value is true.

Constant value: **true**

See Also:

(continued from last page)

[PROPERTY_BREAK_ON_PTS](#)

PROPERTY_REPEATER_SHARED_SECRET

```
public static final java.lang.String PROPERTY_REPEATER_SHARED_SECRET
```

DVR Property "dvrEncryptionSharedSecret": for encryption shared secret between origins and edges. In Origin-Edge scenarios when Wowza is using encryption, a shared secret must be defined for both origin and edges to encrypt the information sent between origin and edge so that encryption information is not revealed.

A String values.

Add the property to Application/DVR/Properties section of Application.xml
Constant value: **dvrEncryptionSharedSecret**

PROPERTY_REPEATER_HEARTBEAT_DURATION

```
public static final java.lang.String PROPERTY_REPEATER_HEARTBEAT_DURATION
```

DVR Property "dvrRepeaterHeartbeatDuration": for time in ms that origin pings edges

A Integer Value.

Add the property to Application/DVR/Properties section of Application.xml
Constant value: **dvrRepeaterHeartbeatDuration**

DEFAULT_PROPERTY_REPEATER_HEARTBEAT_DURATION

```
public static final int DEFAULT_PROPERTY_REPEATER_HEARTBEAT_DURATION
```

Constant value: **4000**

PROPERTY_CHUNK_DURATION_MINIMUM

```
public static final java.lang.String PROPERTY_CHUNK_DURATION_MINIMUM
```

DVR Property "dvrChunkDurationMinimum": for minimum chunk duration, in milliseconds.

Add the property to Application/DVR/Properties section of Application.xml
Constant value: **dvrChunkDurationMinimum**

See Also:

[DEFAULT_PROPERTY_CHUNK_DURATION_MINIMUM](#)

DEFAULT_PROPERTY_CHUNK_DURATION_MINIMUM

```
public static final int DEFAULT_PROPERTY_CHUNK_DURATION_MINIMUM
```

Default value for DVR Property "dvrChunkDurationMinimum".

Default value is 1500 ms.
Constant value: **1500**

See Also:

[PROPERTY_CHUNK_DURATION_MINIMUM](#)

PROPERTY_CHUNK_MEMORY_CACHESIZE

```
public static final java.lang.String PROPERTY_CHUNK_MEMORY_CACHESIZE
```

(continued from last page)

DVR Property "chunkMemoryCacheSize": used by DvrDefaultChunkMemoryCache to set number of chunks stored in DVR in-memory cache.

Default value is [DEFAULT_CHUNK_MEMORY_CACHESIZE](#)

Add this to Application/DVR/Properties section of Application.xml
Constant value: **chunkMemoryCacheSize**

DEFAULT_CHUNK_MEMORY_CACHESIZE

```
public static final int DEFAULT_CHUNK_MEMORY_CACHESIZE
```

Default value for DVR Property "chunkMemoryCacheSize".

Default value is 10 chunks.
Constant value: **10**

See Also:

[PROPERTY_CHUNK_MEMORY_CACHESIZE](#)

PROPERTY_MBR_USE_UTC_FOR_ALIGNMENT

```
public static final java.lang.String PROPERTY_MBR_USE_UTC_FOR_ALIGNMENT
```

DVR Property "dvrMbrUseUtcForAlignment": when doing mbr alignment use utc time for alignment

Add the property to Application/DVR/Properties section of Application.xml
Constant value: **dvrMbrUseUtcForAlignment**

PROPERTY_MBR_MINIMUM_PACKETTIME_GAP_SIZE

```
public static final java.lang.String PROPERTY_MBR_MINIMUM_PACKETTIME_GAP_SIZE
```

DVR Property "dvrMbrMinimumPacketTimeGapSize": when doing mbr alignment gaps smaller than this are ignored.

Add the property to Application/DVR/Properties section of Application.xml
Constant value: **dvrMbrMinimumPacketTimeGapSize**

See Also:

[DEFAULT_PROPERTY_MBR_MINIMUM_PACKETTIME_GAP_SIZE](#)

PROPERTY_MBR_ALTERNATIVE_MATCH_DELTA

```
public static final java.lang.String PROPERTY_MBR_ALTERNATIVE_MATCH_DELTA
```

DVR Property "dvrMbrAlternateMatchDelta": when finding alternative for a MBR gap, an alternate is considered a match is within this many ms of the gap in question.

Add the property to Application/DVR/Properties section of Application.xml
Constant value: **dvrMbrAlternateMatchDelta**

See Also:

[DEFAULT_PROPERTY_MBR_ALTERNATIVE_MATCH_DELTA](#)

DEFAULT_PROPERTY_MBR_ALTERNATIVE_MATCH_DELTA

```
public static final int DEFAULT_PROPERTY_MBR_ALTERNATIVE_MATCH_DELTA
```

Default value for DVR Property "dvrMbrAlternateMatchDelta".

Default value is 1000 ms.
Constant value: **1000**

(continued from last page)

See Also:

[PROPERTY_MBR_ALTERNATIVE_MATCH_DELTA](#)

PROPERTY_PACKET_DURATION_MAXIMUM

```
public static final java.lang.String PROPERTY_PACKET_DURATION_MAXIMUM
```

DVR Property "dvrPacketDurationMaximum": for maximum packet duration, in milliseconds.

Add the property to Application/DVR/Properties section of Application.xml
Constant value: **dvrPacketDurationMaximum**

See Also:

[DEFAULT_PROPERTY_PACKET_DURATION_MAXIMUM](#)

DEFAULT_PROPERTY_PACKET_DURATION_MAXIMUM

```
public static final int DEFAULT_PROPERTY_PACKET_DURATION_MAXIMUM
```

Default value for DVR Property "dvrChunkDurationMinimum".

Default value is 5000 ms.
Constant value: **5000**

See Also:

[PROPERTY_PACKET_DURATION_MAXIMUM](#)

DEFAULT_PROPERTY_MBR_MINIMUM_PACKETTIME_GAP_SIZE

```
public static final int DEFAULT_PROPERTY_MBR_MINIMUM_PACKETTIME_GAP_SIZE
```

Default value for DVR Property "dvrMbrMinimumPacketTimeGapSize".

Default value is 100 ms.
Constant value: **100**

See Also:

[PROPERTY_MBR_MINIMUM_PACKETTIME_GAP_SIZE](#)

PROPERTY_MBR_MINIMUM_UTCTIME_GAP_SIZE

```
public static final java.lang.String PROPERTY_MBR_MINIMUM_UTCTIME_GAP_SIZE
```

DVR Property "dvrMbrMinimumUtcTimeGapSize": when doing mbr alignment gaps smaller than this are ignored.

Add the property to Application/DVR/Properties section of Application.xml
Constant value: **dvrMbrMinimumUtcTimeGapSize**

See Also:

[DEFAULT_PROPERTY_MBR_MINIMUM_UTCTIME_GAP_SIZE](#)

DEFAULT_PROPERTY_MBR_MINIMUM_UTCTIME_GAP_SIZE

```
public static final int DEFAULT_PROPERTY_MBR_MINIMUM_UTCTIME_GAP_SIZE
```

Default value for DVR Property "dvrMbrMinimumPacketTimeGapSize".

Default value is 750 ms.
Constant value: **750**

(continued from last page)

See Also:

[PROPERTY_MBR_MINIMUM_UTCTIME_GAP_SIZE](#)

PROPERTY_MAX_RECORDING_LENGTH

```
public static final java.lang.String PROPERTY_MAX_RECORDING_LENGTH
```

DVR Property "dvrMaximumRecordingLength": The maximum recording length in seconds. Recording stops when it reaches this value.

Add the property to Application/DVR/Properties section of Application.xml A value of zero means no maximum recording length is enforced. If used in conjunction with DVRWindow, should be slightly larger (1-2%) than DVRWindow.
Constant value: **dvrMaximumRecordingLength**

See Also:

[DEFAULT_PROPERTY_MAX_RECORDING_LENGTH](#)

DEFAULT_PROPERTY_MAX_RECORDING_LENGTH

```
public static final long DEFAULT_PROPERTY_MAX_RECORDING_LENGTH
```

Default value for DVR Property "dvrMaximumRecordingLength".

Default value is 108000 (30 hours)

Constant value: **108000**

See Also:

[PROPERTY_MAX_RECORDING_LENGTH](#)

PROPERTY_MEDIACACHE_ENABLED

```
public static final java.lang.String PROPERTY_MEDIACACHE_ENABLED
```

DVR Property "dvrMediaCacheEnabled"

Add the property to Application/DVR/Properties section of Application.xml Default is false.
Constant value: **dvrMediaCacheEnabled**

PROPERTY_MEDIACACHE_READER_CLASS

```
public static final java.lang.String PROPERTY_MEDIACACHE_READER_CLASS
```

DVR Property "dvrMediaCacheReaderClass"

Add the property to Application/DVR/Properties section of Application.xml
Constant value: **dvrMediaCacheReaderClass**

See Also:

[DEFAULT_PROPERTY_MEDIACACHE_READER_CLASS](#)

DEFAULT_PROPERTY_MEDIACACHE_READER_CLASS

```
public static final java.lang.String DEFAULT_PROPERTY_MEDIACACHE_READER_CLASS
```

Default value for DVR Property "dvrMediaCacheReaderClass".

Constant value: **com.wowza.wms.mediacache.impl.MediaCacheRandomAccessReader**

See Also:

[PROPERTY_MEDIACACHE_READER_CLASS](#)

(continued from last page)

PROPERTY_DEBUG_LOG_INVALID_CHUNK_DETAILS

```
public static final java.lang.String PROPERTY_DEBUG_LOG_INVALID_CHUNK_DETAILS
```

DVR Property "dvrLogInvalidChunkDetails": to control detailed logging information of DVR invalid chunks

The default is false, meaning do not log invalid chunks. Add the property to Application/DVR/Properties section of Application.xml

Constant value: **dvrLogInvalidChunkDetails**

See Also:

[PROPERTY_DEBUG_MAX_INVALID_CHUNKS_LOGGED](#)

PROPERTY_DEBUG_LOG_INVALID_CHUNK_MATCHER

```
public static final java.lang.String PROPERTY_DEBUG_LOG_INVALID_CHUNK_MATCHER
```

DVR Property "dvrLogInvalidChunkMatcher": for matching stream names that will log chunk packets

Only affects logging if [PROPERTY_DEBUG_MAX_INVALID_CHUNKS_LOGGED](#) is true.

Constant value: **dvrLogInvalidChunkMatcher**

See Also:

[PROPERTY_DEBUG_MAX_INVALID_CHUNKS_LOGGED](#)

PROPERTY_DEBUG_MAX_INVALID_CHUNKS_LOGGED

```
public static final java.lang.String PROPERTY_DEBUG_MAX_INVALID_CHUNKS_LOGGED
```

DVR Property "dvrMaxInvalidChunksLogged": for controlling maximum number of invalid DVR chunks logged.

The default is 10. Logging only occurs if property [PROPERTY_DEBUG_LOG_INVALID_CHUNK_DETAILS](#) is true. Add the property to Application/DVR/Properties section of Application.xml

Constant value: **dvrMaxInvalidChunksLogged**

See Also:

[DEFAULT_PROPERTY_DEBUG_MAX_INVALID_CHUNKS_LOGGED](#)

[PROPERTY_DEBUG_LOG_INVALID_CHUNK_DETAILS](#)

DEFAULT_PROPERTY_DEBUG_MAX_INVALID_CHUNKS_LOGGED

```
public static final int DEFAULT_PROPERTY_DEBUG_MAX_INVALID_CHUNKS_LOGGED
```

Default value for DVR Property "dvrMaxInvalidChunksLogged".

Default value is 10.

Constant value: **10**

See Also:

[PROPERTY_DEBUG_LOG_INVALID_CHUNK_DETAILS](#)

[PROPERTY_DEBUG_MAX_INVALID_CHUNKS_LOGGED](#)

PROPERTY_DEBUG_LOG_VALID_CHUNK_DETAILS

```
public static final java.lang.String PROPERTY_DEBUG_LOG_VALID_CHUNK_DETAILS
```

DVR Property "dvrLogValidChunkDetails": to control detailed logging information of DVR valid chunks

The default is false, meaning do not log valid chunks. Add the property to Application/DVR/Properties section of Application.xml

Constant value: **dvrLogValidChunkDetails**

(continued from last page)

See Also:[PROPERTY_DEBUG_MAX_VALID_CHUNKS_LOGGED](#)

PROPERTY_DEBUG_MAX_VALID_CHUNKS_LOGGED

```
public static final java.lang.String PROPERTY_DEBUG_MAX_VALID_CHUNKS_LOGGED
```

DVR Property "dvrMaxValidChunksLogged": for controlling maximum number of valid DVR chunks logged.

The default is 10. Logging only occurs if property [PROPERTY_DEBUG_LOG_VALID_CHUNK_DETAILS](#) is true. Add the property to Application/DVR/Properties section of Application.xml
Constant value: **dvrMaxValidChunksLogged**

See Also:[DEFAULT_PROPERTY_DEBUG_MAX_VALID_CHUNKS_LOGGED](#)[PROPERTY_DEBUG_LOG_VALID_CHUNK_DETAILS](#)

DEFAULT_PROPERTY_DEBUG_MAX_VALID_CHUNKS_LOGGED

```
public static final int DEFAULT_PROPERTY_DEBUG_MAX_VALID_CHUNKS_LOGGED
```

Default value for DVR Property "dvrMaxValidChunksLogged".

Default value is 10.
Constant value: **10**

See Also:[PROPERTY_DEBUG_LOG_VALID_CHUNK_DETAILS](#)[PROPERTY_DEBUG_MAX_VALID_CHUNKS_LOGGED](#)

PROPERTY_DEBUG_LOG_VALID_CHUNK_MATCHER

```
public static final java.lang.String PROPERTY_DEBUG_LOG_VALID_CHUNK_MATCHER
```

DVR Property "dvrLogValidChunkMatcher": for matching stream names that will log chunk packets

Only affects logging if [PROPERTY_DEBUG_MAX_VALID_CHUNKS_LOGGED](#) is true.
Constant value: **dvrLogValidChunkMatcher**

See Also:[PROPERTY_DEBUG_MAX_VALID_CHUNKS_LOGGED](#)

PROPERTY_DEBUG_TOSSED_HOLDERS

```
public static final java.lang.String PROPERTY_DEBUG_TOSSED_HOLDERS
```

DVR Property "dvrDebugTossedHolders": for turning on logging of packets that are being tossed.

Valid values are "true" or "false". Default is false.
Constant value: **dvrDebugTossedHolders**

PROPERTY_DEBUG_RAW_PACKETS

```
public static final java.lang.String PROPERTY_DEBUG_RAW_PACKETS
```

DVR Property "dvrDebugRawPackets": for turning on logging of incoming raw packets.

Valid values are "true" or "false".
Constant value: **dvrDebugRawPackets**

See Also:

(continued from last page)

[PROPERTY_DEBUG_MAXIMUM_RAW_PACKETS](#)

DEFAULT_PROPERTY_DEBUG_RAW_PACKETS

```
public static final boolean DEFAULT_PROPERTY_DEBUG_RAW_PACKETS
```

Default value for DVR Property "dvrDebugRawPackets".

Default value is false.

Constant value: **false**

See Also:

[PROPERTY_DEBUG_RAW_PACKETS](#)

[PROPERTY_DEBUG_RAW_PACKETS_MATCHER](#)

PROPERTY_DEBUG_RAW_PACKETS_MATCHER

```
public static final java.lang.String PROPERTY_DEBUG_RAW_PACKETS_MATCHER
```

DVR Property "dvrDebugRawPacketsMatcher": for matching stream names that will dump raw packet.

Only affects logging if [PROPERTY_DEBUG_RAW_PACKETS](#) is true.

Constant value: **dvrDebugRawPacketsMatcher**

See Also:

[PROPERTY_DEBUG_RAW_PACKETS](#)

PROPERTY_DEBUG_MAXIMUM_RAW_PACKETS

```
public static final java.lang.String PROPERTY_DEBUG_MAXIMUM_RAW_PACKETS
```

DVR Property "dvrDebugMaximumRawPackets": for setting maximum number of logged raw packets.

Valid values is an integer. 0 means there is no limit.

Only affects logging if [PROPERTY_DEBUG_RAW_PACKETS](#) is true.

Constant value: **dvrDebugMaximumRawPackets**

See Also:

[PROPERTY_DEBUG_RAW_PACKETS](#)

DEFAULT_PROPERTY_DEBUG_MAX_RAW_PACKETS

```
public static final int DEFAULT_PROPERTY_DEBUG_MAX_RAW_PACKETS
```

Default value for DVR Property "dvrDebugMaximumRawPackets".

Default value is 200 log statements.

Constant value: **200**

See Also:

[PROPERTY_DEBUG_MAXIMUM_RAW_PACKETS](#)

PROPERTY_DEBUG_STATE_CHANGE

```
public static final java.lang.String PROPERTY_DEBUG_STATE_CHANGE
```

DVR Property "dvrDebugStateChange": for logging state changes of DVR store.

Constant value: **dvrDebugStateChange**

(continued from last page)

PROPERTY_DEBUG_CHUNK_RETRIEVALS

```
public static final java.lang.String PROPERTY_DEBUG_CHUNK_RETRIEVALS
```

DVR Property "dvrDebugChunkRetrievals": for logging each chunk retrieval
Constant value: **dvrDebugChunkRetrievals**

PROPERTY_DEBUG_FAILED_CHUNK_RETRIEVALS

```
public static final java.lang.String PROPERTY_DEBUG_FAILED_CHUNK_RETRIEVALS
```

DVR Property "dvrDebugFailedChunkRetrievals": for logging info about each failed chunk retrieval
Constant value: **dvrDebugFailedChunkRetrievals**

PROPERTY_MAX_CHUNK_LOG

```
public static final java.lang.String PROPERTY_MAX_CHUNK_LOG
```

DVR Property "dvrMaxChunkLogCount ": for maximum number of DVR chunks to log.

Maximum number of chunks to log recording information about in the wowzamedia_access.log file. As recording is continued past this value, there is no feedback for normal operation in the log. View directories and files created in [install-dir]/dvr for on-going feedback that recording is occurring. Add the property to Application/DVR/Properties section of Application.xml
Constant value: **dvrMaxChunkLogCount**

See Also:

[DEFAULT_PROPERTY_DVR_MAX_CHUNK_LOG](#)

DEFAULT_PROPERTY_DVR_MAX_CHUNK_LOG

```
public static final int DEFAULT_PROPERTY_DVR_MAX_CHUNK_LOG
```

Default value for DVR Property "dvrMaxChunkLogCount ".

Default value is 10 logging statements.
Constant value: **10**

See Also:

[PROPERTY_MAX_CHUNK_LOG](#)

PROPERTY_MAX_RECALC_DURATION_LOG

```
public static final java.lang.String PROPERTY_MAX_RECALC_DURATION_LOG
```

DVR Property "dvrMaxRecountDurationLogCount ": for maximum number of messages to log regarding recalculating DVR chunk duration

Maximum number of log statements recorded regarding chunk duration being recalculated. A value less than 0 means all statements will be logged with no limit.
Constant value: **dvrMaxRecountDurationLogCount**

See Also:

[DEFAULT_PROPERTY_MAX_RECALC_DURATION_LOG](#)

DEFAULT_PROPERTY_MAX_RECALC_DURATION_LOG

```
public static final int DEFAULT_PROPERTY_MAX_RECALC_DURATION_LOG
```

Default value for DVR Property "dvrMaxRecountDurationLogCount ".

Default value is 30 logging statements.

(continued from last page)

Constant value: **30**

See Also:

[PROPERTY_MAX_RECALC_DURATION_LOG](#)

PROPERTY_DEBUG_METHODS

```
public static final java.lang.String PROPERTY_DEBUG_METHODS
```

DVR Property "dvrDebugManagerLogMethods": for turning on DVR Manager debug logging.

Valid values are "true" or "false".

Add the property to Application/DVR/Properties section of Application.xml

Constant value: **dvrDebugManagerLogMethods**

PROPERTY_DEBUG_REPEATER

```
public static final java.lang.String PROPERTY_DEBUG_REPEATER
```

DVR Property "dvrDebugRepeater": for turning on logging of DVR repeater

Valid values are "true" or "false".

Constant value: **dvrDebugRepeater**

PROPERTY_DEBUG_PLAYER_ADAPTER

```
public static final java.lang.String PROPERTY_DEBUG_PLAYER_ADAPTER
```

HTTP Streamer Property "dvrDebugPlayerAdapter": for turning on DVR Player Adapter debug logging for all streamer types.

Valid values are "true" or "false". (Default is false)

Add this property to Application/HTTPStreamer/Properties.

Constant value: **dvrDebugPlayerAdapter**

PROPERTY_DEBUG_SMOOTH_PLAYER_ADAPTER

```
public static final java.lang.String PROPERTY_DEBUG_SMOOTH_PLAYER_ADAPTER
```

HTTP Streamer Property "dvrDebugSmoothPlayerAdapter": for turning on DVR Player Smooth Adapter debug logging.

Valid values are "true" or "false". (Default is false)

Add this property to Application/HTTPStreamer/Properties.

Constant value: **dvrDebugSmoothPlayerAdapter**

PROPERTY_DEBUG_SANJOSE_PLAYER_ADAPTER

```
public static final java.lang.String PROPERTY_DEBUG_SANJOSE_PLAYER_ADAPTER
```

HTTP Streamer Property "dvrDebugSanJosePlayerAdapter": for turning on DVR Player San Jose Adapter debug logging.

Valid values are "true" or "false". (Default is false)

Add this property to Application/HTTPStreamer/Properties.

Constant value: **dvrDebugSanJosePlayerAdapter**

(continued from last page)

PROPERTY_DEBUG_CUPERTINO_PLAYER_ADAPTER

```
public static final java.lang.String PROPERTY_DEBUG_CUPERTINO_PLAYER_ADAPTER
```

HTTP Streamer Property "dvrDebugCupertinoPlayerAdapter": for turning on DVR Player Cupertino Adapter debug logging.

Valid values are "true" or "false". (Default is false)

Add this property to Application/HTTPStreamer/Properties.
Constant value: **dvrDebugCupertinoPlayerAdapter**

PROPERTY_DEBUG_MBR_PLAYER_ADAPTER

```
public static final java.lang.String PROPERTY_DEBUG_MBR_PLAYER_ADAPTER
```

HTTP Streamer or DVR Property "dvrDebugMbrPlayerAdapter": for turning on DVR MBR Player Adapter debug logging.

Valid values are "true" or "false". (Default is false)

Add this property to Application/HTTPStreamer/Properties.
Constant value: **dvrDebugMbrPlayerAdapter**

PROPERTY_DEBUG_MBR_PLAYER_ADAPTER_IF_SHIFTED_EMPTY

```
public static final java.lang.String  
PROPERTY_DEBUG_MBR_PLAYER_ADAPTER_IF_SHIFTED_EMPTY
```

HTTP Streamer or DVR Property "dvrDebugMbrPlayerAdapter": for turning on DVR MBR Player Adapter debug logging when null shifted entries is returned.

Valid values are "true" or "false". (Default is false)

Add this property to Application/HTTPStreamer/Properties.
Constant value: **dvrDebugMbrPlayerAdapterWhenShiftedEmpty**

PROPERTY_DEBUG_MBR_DETAILS_WHEN_RECORD_DNE

```
public static final java.lang.String PROPERTY_DEBUG_MBR_DETAILS_WHEN_RECORD_DNE
```

HTTP Streamer or DVR Property "dvrDebugMbrDetailsWhenRecordDNE": for turning on DVR MBR Player Adapter debug logging.

Valid values are "true" or "false". (Default is false)

Add this property to Application/HTTPStreamer/Properties.
Constant value: **dvrDebugMbrDetailsWhenRecordDNE**

PROPERTY_DEBUG_MBR_ALIGNMENT

```
public static final java.lang.String PROPERTY_DEBUG_MBR_ALIGNMENT
```

HTTP Streamer or DVR Property "dvrDebugMbrAlignment": for turning on logging of mbr alignment

Valid values are "true" or "false". Default is false.
Constant value: **dvrDebugMbrAlignment**

PROPERTY_DEBUG_MBR_ALIGNMENT_RESOLUTION

```
public static final java.lang.String PROPERTY_DEBUG_MBR_ALIGNMENT_RESOLUTION
```

(continued from last page)

DVR Property "dvrDebugMbrAlignmentResolution": for turning on logging of mbr alignment resolution (requested to actual)

Valid values are "true" or "false". Default is false.

Constant value: **dvrDebugMbrAlignmentResolution**

PROPERTY_DEBUG_PLAYLIST_REQUEST

```
public static final java.lang.String PROPERTY_DEBUG_PLAYLIST_REQUEST
```

DVR Property "dvrDebugPlaylistRequest": for turning on logging of DVR playlist requests.

Valid values are "true" or "false". Default is false.

Constant value: **dvrDebugPlaylistRequest**

PROPERTY_PLAYLIST_REQUEST_DELEGATE

```
public static final java.lang.String PROPERTY_PLAYLIST_REQUEST_DELEGATE
```

DVR Property "dvrPlaylistRequestDelegate": used to over-ride the class that generates a playlist request.

The delegate should extend DvrBasePlaylistRequestFactory

If unspecified, the standard DVR-time based delegate [DEFAULT_PLAYLIST_REQUEST_DELEGATE](#) is used

Add this to Application/DVR/Properties section of Application.xml

Constant value: **dvrPlaylistRequestDelegate**

DEFAULT_PLAYLIST_REQUEST_DELEGATE

```
public static final java.lang.String DEFAULT_PLAYLIST_REQUEST_DELEGATE
```

Constant value: **com.wowza.wms.dvr.impl.DvrStartDurationPlaylistRequestDelegate**

PROPERTY_ENCRYPTION_INFO_DELEGATE

```
public static final java.lang.String PROPERTY_ENCRYPTION_INFO_DELEGATE
```

DVR Property "dvrPlaylistEncryptionInfoDelegate": used to over-ride the encryption info on the playback side.

The specified delegate should extend com.wowza.wms.dvr.DvrBaseEncryptionInfoDelegate

Add this to Application/DVR/Properties section of Application.xml

Constant value: **dvrPlaylistEncryptionInfoDelegate**

PROPERTY_SANJOSE_PLAYLIST_MIMETYPE

```
public static final java.lang.String PROPERTY_SANJOSE_PLAYLIST_MIMETYPE
```

DVR Property "dvrSanJosePlaylistMimeType": used to over-ride the mime type for DVR San Jose F4m playlists.

Default value is "video/mp4"

Add this to Application/DVR/Properties section of Application.xml

Constant value: **dvrSanJosePlaylistMimeType**

PROPERTY_SANJOSE_PLAYLIST_DELIVERYTYPE

```
public static final java.lang.String PROPERTY_SANJOSE_PLAYLIST_DELIVERYTYPE
```

(continued from last page)

DVR Property "dvrSanJosePlaylistDeliveryType": used to over-ride the mime type for DVR San Jose F4m playlists.

Default value is "streaming"

Add this to Application/DVR/Properties section of Application.xml

Constant value: **dvrSanJosePlaylistDeliveryType**

PROPERTY_SANJOSE_PLAYLIST_VERSION

```
public static final java.lang.String PROPERTY_SANJOSE_PLAYLIST_VERSION
```

DVR Property "dvrSanJosePlaylistVersion": used to over-ride the version of the San Jose f4m playlist.

Default value is "2.0". Valid values are "1.0" and "2.0".

Constant value: **dvrSanJosePlaylistVersion**

See Also:

[SANJOSE_F4M_VERSION_1_0](#)

[Add this to Application/DVR/Properties section of Application.xml](#)

PROPERTY_SANJOSE_PLAYLIST_RECORDED_STREAMTYPE

```
public static final java.lang.String PROPERTY_SANJOSE_PLAYLIST_RECORDED_STREAMTYPE
```

DVR Property "dvrSanJosePlaylistRecordedStreamType": used to over-ride the stream type for non-live (recorded) DVR in San Jose f4m playlists.

Default value is #SANJOSE_F4M_STREAMTYPE_RECORDED.*

Constant value: **dvrSanJosePlaylistRecordedStreamType**

See Also:

[SANJOSE_F4M_STREAMTYPE_DVR](#)

[SANJOSE_F4M_STREAMTYPE_LIVE](#)

[SANJOSE_F4M_STREAMTYPE_LIVEORRECORDED](#)

Add this to Application/DVR/Properties section of Application.xml

PROPERTY_SANJOSE_PLAYLIST_LIVE_STREAMTYPE

```
public static final java.lang.String PROPERTY_SANJOSE_PLAYLIST_LIVE_STREAMTYPE
```

DVR Property "dvrSanJosePlaylistLiveStreamType": used to over-ride the stream type for live DVR in San Jose f4m playlists.

Default value is #SANJOSE_F4M_STREAMTYPE_DVR.

Constant value: **dvrSanJosePlaylistLiveStreamType**

See Also:

[SANJOSE_F4M_STREAMTYPE_DVR](#)

[SANJOSE_F4M_STREAMTYPE_LIVE](#)

[SANJOSE_F4M_STREAMTYPE_LIVEORRECORDED](#)

Add this to Application/DVR/Properties section of Application.xml

SANJOSE_F4M_STREAMTYPE_DVR

```
public static final java.lang.String SANJOSE_F4M_STREAMTYPE_DVR
```

Constant for San Jose stream type "dvr".

(continued from last page)

Constant value: **dvr**

SANJOSE_F4M_STREAMTYPE_RECORDED

```
public static final java.lang.String SANJOSE_F4M_STREAMTYPE_RECORDED
```

Constant for San Jose stream type "recorded".
Constant value: **recorded**

SANJOSE_F4M_STREAMTYPE_LIVE

```
public static final java.lang.String SANJOSE_F4M_STREAMTYPE_LIVE
```

Constant for San Jose stream type "live".
Constant value: **live**

SANJOSE_F4M_STREAMTYPE_LIVEORRECORDED

```
public static final java.lang.String SANJOSE_F4M_STREAMTYPE_LIVEORRECORDED
```

Constant for San Jose stream type "liveOrRecorded".
Constant value: **liveOrRecorded**

SANJOSE_F4M_VERSION_1_0

```
public static final java.lang.String SANJOSE_F4M_VERSION_1_0
```

Constant for San Jose f4m version "1.0".
Constant value: **1.0**

SANJOSE_F4M_VERSION_2_0

```
public static final java.lang.String SANJOSE_F4M_VERSION_2_0
```

Constant for San Jose f4m version "2.0".
Constant value: **2.0**

SANJOSE_F4M_STREAMINGTYPE_STREAMING

```
public static final java.lang.String SANJOSE_F4M_STREAMINGTYPE_STREAMING
```

Constant for San Jose streaming type "streaming".
Constant value: **streaming**

MIMETYPE_VIDEO_MP4

```
public static final java.lang.String MIMETYPE_VIDEO_MP4
```

Constant for mime type "video/mp4".
Constant value: **video/mp4**

PROPERTY_SANJOSE_ABST_TIMESCALE

```
public static final java.lang.String PROPERTY_SANJOSE_ABST_TIMESCALE
```

DVR Property "dvrSanJosePlaylistAbstTimescale": used to over-ride the time-scale for abst files.

Default value is #SANJOSE_ABST_DEFAULT_TIMESCALE (1000), which means milliseconds.

Add this to Application/DVR/Properties section of Application.xml

(continued from last page)

Constant value: **dvrSanJosePlaylistAbstTimescale**

PROPERTY_SANJOSE_ABST_DURATION_TOLERANCE

```
public static final java.lang.String PROPERTY_SANJOSE_ABST_DURATION_TOLERANCE
```

DVR Property "dvrSanJosePlaylistAbstDurationEqualityTolerance": used to over-ride the tolerance when determining equal chunks lengths.

Default value is #SANJOSE_ABST_DEFAULT_DURATION_EQUALITY_TOLERANCE (50), which means 50 milliseconds.

Add this to Application/DVR/Properties section of Application.xml

Constant value: **dvrSanJosePlaylistAbstDurationTolerance**

DEFAULT_PROPERTY_SANJOSE_ABST_TIMESCALE

```
public static final int DEFAULT_PROPERTY_SANJOSE_ABST_TIMESCALE
```

Default value for property "dvrSanJosePlaylistAbstTimescale" #see [PROPERTY_SANJOSE_ABST_TIMESCALE](#)

Constant value: **1000**

DEFAULT_PROPERTY_SANJOSE_ABST_DURATION_TOLERANCE

```
public static final int DEFAULT_PROPERTY_SANJOSE_ABST_DURATION_TOLERANCE
```

Default value for property "dvrSanJosePlaylistAbstDurationTolerance" #see [PROPERTY_SANJOSE_ABST_DURATION_TOLERANCE](#)

Constant value: **4**

PROPERTY_CUPERTINO_PLAYLIST_FORCE_LIVE

```
public static final java.lang.String PROPERTY_CUPERTINO_PLAYLIST_FORCE_LIVE
```

DVR Property "dvrCupertinoPlaylistForceLive": used to override playlist request delegate logic that determines if playlist is live.

Default value is false

Add this to Application/DVR/Properties section of Application.xml

Constant value: **dvrCupertinoPlaylistForceLive**

PROPERTY_CUPERTINO_PLAYLIST_FORCE_NONLIVE

```
public static final java.lang.String PROPERTY_CUPERTINO_PLAYLIST_FORCE_NONLIVE
```

DVR Property "dvrCupertinoPlaylistForceLive": used to override playlist request delegate logic that determines if playlist is live versus non-live.

Default value is false

Add this to Application/DVR/Properties section of Application.xml

Constant value: **dvrCupertinoPlaylistForceNonLive**

PROPERTY_CUPERTINO_PLAYLIST_USE_GZIP

```
public static final java.lang.String PROPERTY_CUPERTINO_PLAYLIST_USE_GZIP
```

DVR Property "dvrCupertinoPlaylistUseGzip": used to force Cupertino playlist to use gzip if it is accepted

Default value is true

Add this to Application/DVR/Properties section of Application.xml

(continued from last page)

Constant value: **dvrCupertinoPlaylistUseGzip**

PROPERTY_CUPERTINO_PLAYLIST_GZIP_THRESHOLD

```
public static final java.lang.String PROPERTY_CUPERTINO_PLAYLIST_GZIP_THRESHOLD
```

DVR Property "dvrCupertinoPlaylistGzipThreshold": when playlist is larger than this number of bytes, and gzip is enabled and accepted, the playlist will be compressed

Default value is [DEFAULT_CUPERTINO_PLAYLIST_GZIP_THRESHOLD](#)

Add this to Application/DVR/Properties section of Application.xml

Constant value: **dvrCupertinoPlaylistGzipThreshold**

DEFAULT_CUPERTINO_PLAYLIST_GZIP_THRESHOLD

```
public static final int DEFAULT_CUPERTINO_PLAYLIST_GZIP_THRESHOLD
```

Default value for property "dvrCupertinoPlaylistGzipThreshold" #see [PROPERTY_CUPERTINO_PLAYLIST_GZIP_THRESHOLD](#)

Constant value: **4000**

PROPERTY_CUPERTINO_ON_CHUNK_START_RESET_COUNTER

```
public static final java.lang.String PROPERTY_CUPERTINO_ON_CHUNK_START_RESET_COUNTER
```

DVR Property "dvrCupertinoOnChunkStartResetCounter": when a new chunk starts, reset internal tsPacketizer counters

Default value is false

Add this to Application/DVR/Properties section of Application.xml

Constant value: **dvrCupertinoOnChunkStartResetCounter**

PROPERTY_CUPERTINO_PLAYLIST_ALLOW_CACHING

```
public static final java.lang.String PROPERTY_CUPERTINO_PLAYLIST_ALLOW_CACHING
```

DVR Property "dvrCupertinoPlaylistAllowCaching": used to force Cupertino playlist to set #EXT-X-ALLOW-CACHE: value

Default value is false

Add this to Application/DVR/Properties section of Application.xml

Constant value: **dvrCupertinoPlaylistAllowCaching**

PROPERTY_CUPERTINO_PUT_END_TAG_BEFORE_CHUNKS

```
public static final java.lang.String PROPERTY_CUPERTINO_PUT_END_TAG_BEFORE_CHUNKS
```

DVR Property "dvrCupertinoPutEndTagBeforeChunks": if set to true, puts the EXT-X-ENDLIST before the DVR chunks.

Default value is false

Add this to Application/DVR/Properties section of Application.xml

Constant value: **dvrCupertinoPutEndTagBeforeChunks**

PROPERTY_SMOOTH_MANIFEST_LIVE_CAN_SEEK

```
public static final java.lang.String PROPERTY_SMOOTH_MANIFEST_LIVE_CAN_SEEK
```

(continued from last page)

DVR Property "dvrSmoothManifestLiveCanSeek": used to determine is CanSeek is enabled in smooth Manifest.

Default value is #DEFAULT_PROPERTY_SMOOTH_LIVE_CAN_SEEK, which is set to true.

Add this to Application/DVR/Properties section of Application.xml

Constant value: **dvrSmoothManifestLiveCanSeek**

DEFAULT_PROPERTY_SMOOTH_MANIFEST_LIVE_CAN_SEEK

```
public static final boolean DEFAULT_PROPERTY_SMOOTH_MANIFEST_LIVE_CAN_SEEK
```

Constant value: **true**

PROPERTY_SMOOTH_MANIFEST_LIVE_CAN_PAUSE

```
public static final java.lang.String PROPERTY_SMOOTH_MANIFEST_LIVE_CAN_PAUSE
```

DVR Property "dvrSmoothManifestLiveCanPause": used to determine is CanPause is enabled in smooth Manifest.

Default value is #DEFAULT_PROPERTY_SMOOTH_MANIFEST_LIVE_CAN_PAUSE, which is set to true.

Add this to Application/DVR/Properties section of Application.xml

Constant value: **dvrSmoothManifestLiveCanPause**

DEFAULT_PROPERTY_SMOOTH_MANIFEST_LIVE_CAN_PAUSE

```
public static final boolean DEFAULT_PROPERTY_SMOOTH_MANIFEST_LIVE_CAN_PAUSE
```

Constant value: **true**

PROPERTY_SMOOTH_MANIFEST_RECORDED_CAN_SEEK

```
public static final java.lang.String PROPERTY_SMOOTH_MANIFEST_RECORDED_CAN_SEEK
```

DVR Property "dvrSmoothManifestRecordedCanSeek": used to determine is CanSeek is enabled in smooth Manifest.

Default value is #DEFAULT_PROPERTY_SMOOTH_MANIFEST_RECORDED_CAN_SEEK, which is set to true.

Add this to Application/DVR/Properties section of Application.xml

Constant value: **dvrSmoothManifestRecordedCanSeek**

DEFAULT_PROPERTY_SMOOTH_MANIFEST_RECORDED_CAN_SEEK

```
public static final boolean DEFAULT_PROPERTY_SMOOTH_MANIFEST_RECORDED_CAN_SEEK
```

Constant value: **true**

PROPERTY_SMOOTH_MANIFEST_RECORDED_CAN_PAUSE

```
public static final java.lang.String PROPERTY_SMOOTH_MANIFEST_RECORDED_CAN_PAUSE
```

DVR Property "dvrSmoothManifestRecordedCanPause": used to determine is CanPause is enabled in smooth Manifest.

Default value is #DEFAULT_PROPERTY_SMOOTH_MANIFEST_RECORDED_CAN_PAUSE, which is set to true.

Add this to Application/DVR/Properties section of Application.xml

Constant value: **dvrSmoothManifestRecordedCanPause**

(continued from last page)

DEFAULT_PROPERTY_SMOOTH_MANIFEST_RECORDED_CAN_PAUSE

```
public static final boolean DEFAULT_PROPERTY_SMOOTH_MANIFEST_RECORDED_CAN_PAUSE
```

Constant value: **true**

PROPERTY_SMOOTH_MANIFEST_RECORDED_SPECIFY_DURATION

```
public static final java.lang.String  
PROPERTY_SMOOTH_MANIFEST_RECORDED_SPECIFY_DURATION
```

DVR Property "dvrSmoothManifestRecordedSpecifyDuration": used to determine if duration is specified. If not duration of zero is shown.

Default value is #DEFAULT_PROPERTY_SMOOTH_MANIFEST_RECORDED_SPECIFY_DURATION, which is set to true.

Add this to Application/DVR/Properties section of Application.xml
Constant value: **dvrSmoothManifestRecordedSpecifyDuration**

DEFAULT_PROPERTY_SMOOTH_MANIFEST_RECORDED_SPECIFY_DURATION

```
public static final boolean DEFAULT_PROPERTY_SMOOTH_MANIFEST_RECORDED_SPECIFY_DURATION
```

Constant value: **true**

PROPERTY_SMOOTH_MANIFEST_MAJOR_VERSION

```
public static final java.lang.String PROPERTY_SMOOTH_MANIFEST_MAJOR_VERSION
```

DVR Property "dvrSmoothManifestMajorVersion": used to determine Smooth manifest major version

Default value is #DEFAULT_PROPERTY_SMOOTH_MANIFEST_MAJOR_VERSION.

Add this to Application/DVR/Properties section of Application.xml
Constant value: **dvrSmoothManifestMajorVersion**

DEFAULT_PROPERTY_SMOOTH_MANIFEST_MAJOR_VERSION

```
public static final int DEFAULT_PROPERTY_SMOOTH_MANIFEST_MAJOR_VERSION
```

Constant value: **2**

PROPERTY_SMOOTH_MANIFEST_MINOR_VERSION

```
public static final java.lang.String PROPERTY_SMOOTH_MANIFEST_MINOR_VERSION
```

DVR Property "dvrSmoothManifestMinorVersion": used to determine Smooth manifest major version

Default value is #DEFAULT_PROPERTY_SMOOTH_MANIFEST_MINOR_VERSION.

Add this to Application/DVR/Properties section of Application.xml
Constant value: **dvrSmoothManifestMinorVersion**

DEFAULT_PROPERTY_SMOOTH_MANIFEST_MINOR_VERSION

```
public static final int DEFAULT_PROPERTY_SMOOTH_MANIFEST_MINOR_VERSION
```

(continued from last page)

Constant value: **1**

PROPERTY_SMOOTH_MANIFEST_VERBOSE_DURATION

```
public static final java.lang.String PROPERTY_SMOOTH_MANIFEST_VERBOSE_DURATION
```

DVR Property "dvrSmoothManifestVerboseDuration": used to force Smooth Manifest to include durations for each record

Default value is false

Add this to Application/DVR/Properties section of Application.xml

Constant value: **dvrSmoothManifestVerboseDuration**

PROPERTY_SMOOTH_MANIFEST_H264_CODEC

```
public static final java.lang.String PROPERTY_SMOOTH_MANIFEST_H264_CODEC
```

DVR Property "dvrSmoothManifestH264Codec": used to force Smooth Manifest to use this as its FourCC H264 codec info.

Default value is set by Smooth Streaming

Add this to Application/DVR/Properties section of Application.xml

Constant value: **dvrSmoothManifestH264Codec**

PROPERTY_STREAM_DATA

```
public static final java.lang.String PROPERTY_STREAM_DATA
```

Constant value: **dvrStreamRecordedData**

com.wowza.wms.dvr Class IDvrConstants.DvrTimeScale

```

java.lang.Object
  |
  +- java.lang.Enum
        |
        +- com.wowza.wms.dvr.IDvrConstants.DvrTimeScale
  
```

All Implemented Interfaces:

java.io.Serializable, Comparable

public static final class **IDvrConstants.DvrTimeScale**
extends Enum

Field Summary

public static final	DVR_TIME
public static final	PACKET_TIME
public static final	UTC_TIME

Method Summary

static IDvrConstants.DvrTimeScale	valueOf (String name)
static IDvrConstants.DvrTimeScale[]	values ()

Methods inherited from class java.lang.Enum

clone, compareTo, equals, finalize, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface java.lang.Comparable

compareTo

Fields

(continued from last page)

DVR_TIME

```
public static final com.wowza.wms.dvr.IDvrConstants.DvrTimeScale DVR_TIME
```

PACKET_TIME

```
public static final com.wowza.wms.dvr.IDvrConstants.DvrTimeScale PACKET_TIME
```

UTC_TIME

```
public static final com.wowza.wms.dvr.IDvrConstants.DvrTimeScale UTC_TIME
```

Methods

values

```
public static IDvrConstants.DvrTimeScale\[\] values()
```

valueOf

```
public static IDvrConstants.DvrTimeScale valueOf(String name)
```

com.wowza.wms.dvr Interface IDvrEncryptionInfo

public interface **IDvrEncryptionInfo**
extends Cloneable

Interface for persisting encryption information for DVR.

Field Summary

public static final	<code>ENCRYPTION_TYPE_CENC</code> Common Encryption encryption type. Value: 3
public static final	<code>ENCRYPTION_TYPE_CUPERTINO</code> Cupertino AES encryption type. Value: 1
public static final	<code>ENCRYPTION_TYPE_PLAYREADY</code> Playready encryption type. Value: 2

Method Summary

Object	<code>clone()</code>
int	<code>getEncryptionType()</code> Serialize encryption info.
int	<code>getSerializeSize()</code> Return number of bytes required to serialize this encryption info.
byte[]	<code>serialize()</code> Serialize the encryption info, returning a byte buffer containing the serialized data.
int	<code>serialize(byte[] buffer, int pos)</code> Serialize the encryption info into buffer starting at position 'pos'.

Fields

ENCRYPTION_TYPE_CUPERTINO

public static final int **ENCRYPTION_TYPE_CUPERTINO**

Cupertino AES encryption type.
Constant value: **1**

ENCRYPTION_TYPE_PLAYREADY

public static final int **ENCRYPTION_TYPE_PLAYREADY**

Playready encryption type.

(continued from last page)

Constant value: **2**

ENCRYPTION_TYPE_CENC

```
public static final int ENCRYPTION_TYPE_CENC
```

Common Encryption encryption type.
Constant value: **3**

Methods

getSerializeSize

```
public int getSerializeSize()
```

Return number of bytes required to serialize this encryption info.

Returns:

number of bytes to serialize.

getEncryptionType

```
public int getEncryptionType()
```

Serialize encryption info.

The encryption should consist of the following:

- version (1 byte)
- encryption type (1 byte)
- length of 'blob' (2 bytes)
- byte blob (n bytes)

Returns:

encryption type (constants defined in this class)

serialize

```
public int serialize(byte[] buffer,  
int pos)
```

Serialize the encryption info into buffer starting at position 'pos'.

Parameters:

`buffer` - buffer to serialize to
`pos` - position to start serializing

Returns:

new position into buffer

serialize

```
public byte[] serialize()
```

Serialize the encryption info, returning a byte buffer containing the serialized data.

Returns:

(continued from last page)

byte buffer of serialized data

clone

```
public Object clone()  
    throws CloneNotSupportedException
```

com.wowza.wms.dvr Interface IDvrManifest

public interface **IDvrManifest**
extends

Interface for a DVR Store manifest. A DVR manifest collects type-specific manifest info, including:

- Audio manifest
- Video manifest
- data manifest
- time map manifest, to map DVR, packet and real time
- codec manifest to track codec information
- onMetadata manifest

The difference between retrieving 'recorded entries' and 'live entries' is an important one. Recorded Entries means every current manifest entry. Live Entries refers to a subset of these because a playlist will not include the very last entries, as some chunks must be available for buffering.

Field Summary	
public static final	CODEC_TYPE Constant for codec manifest type. Value: 999
public static final	MANIFEST_TAGNAME_CAN_PLAY Value: CanPlay
public static final	MANIFEST_TAGNAME_CAN_RECORD Value: CanRecord
public static final	MANIFEST_TAGNAME_CHUNK_GROUPING Value: ChunkGrouping
public static final	MANIFEST_TAGNAME_CURRENT_TIME Value: CurrentTime
public static final	MANIFEST_TAGNAME_HAS_ENCRYPTION Value: HasEncryption
public static final	MANIFEST_TAGNAME_PURGE_TIME Value: PurgeTime
public static final	ON_METADATA_TYPE Constant for 'onMetadata' manifest type. Value: 0
public static final	TIME_MAP_TYPE Constant for time map manifest type. Value: 998

Method Summary

void	<code>addToManifest</code> (java.util.List entries) Add manifest entries to the manifest
void	<code>deserialize</code> (byte[] bytes) Deserialize the manifest.
long	<code>expandEndTime</code> (int type, long dvrEndTime)
long	<code>expandStartTime</code> (int type, long dvrStartTime)
long	<code>getClosestStartTime</code> (int type, long dvrTime) Given a dvrTime and a manifest type, find the closest chunk starting time.
<code>DvrManifestCodecEntry</code>	<code>getCodecEntryForTime</code> (long dvrTime) Return most recent codec entry for a given DVR time.
long	<code>getDvrTime</code> (int type)
<code>DvrManifestEntryRangeGroup</code>	<code>getEntriesToPurge</code> (long purgeTime) Given a purge time, return a group of manifest ranges to purge.
<code>DvrManifestEntry</code>	<code>getFirstEntry</code> (int type) Given manifest type, get the first playlist manifest entry.
<code>DvrManifestEntry</code>	<code>getLastLiveEntry</code> (int type) Given manifest type, get the last playlist manifest entry.
<code>DvrManifestEntry</code>	<code>getLastRecordedEntry</code> (int type) Given manifest type and DVR time, get last entry.
int	<code>getLastRecordedIndex</code> (int type) Return last index of DVR entry for given type.
long	<code>getLiveDuration</code> (int type) Given manifest type, return DVR live duration in seconds
java.util.List	<code>getLiveEntries</code> (int type, long dvrStart) Given manifest type and DVR time, get a list of the live entries.
java.util.List	<code>getLiveEntriesWithLimit</code> (int type, long dvrTime, int limit) Given manifest type and DVR time, get a list of the live entries limiting number of returned items.
DvrChannelManifest	<code>getManifestChannel</code> (int type) For a given manifest type, return the channel manifest.
<code>DvrManifestOnMetadataEntry</code>	<code>getMetadataEntryForTime</code> (long dvrTime) Return most recent metadataEntry for a given DVR time.
int	<code>getNextChunkIndex</code> ()
int	<code>getNextCodecIndex</code> ()
int	<code>getNextMetadataIndex</code> ()

int	<u>getNextTimeMapIndex()</u>
int	<u>getNumberLiveEntries</u> (int type, long dvrStart) Given manifest type and DVR time, get the number of live entries at the given time.
int	<u>getNumberLiveEntries</u> (int type, long dvrStart, long dvrEnd) Given manifest type and DVR time, get the number of live entries at the given time.
int	<u>getNumberRecordedEntries</u> (int type, long dvrStart) Given manifest type and DVR time, get the number of entries at the given time.
int	<u>getNumberRecordedEntries</u> (int type, long dvrStart, long dvrEnd) Given manifest type and DVR time, get the number of live entries at the given time.
long	<u>getRecordedDuration</u> (int type) Given manifest type, return DVR recorded duration in seconds
java.util.List	<u>getRecordedEntries</u> (int type) Get a copy of all manifest entries of a given type.
java.util.List	<u>getRecordedEntries</u> (int type, long dvrStartTime) Given manifest type and DVR time, get a list of recorded entries.
java.util.List	<u>getRecordedEntries</u> (int type, long dvrStartTime, long dvrEndTime) Given manifest type and DVR time, get a list of recorded entries.
java.util.List	<u>getRecordedEntriesInRange</u> (<u>DvrManifestEntryRange</u> range) Given a DvrManifestEntryRange, get a list of recorded entries in this range.
java.util.Map	<u>getRecordedEntriesMap</u> (int type) Get a copy of all manifest entries of a given type as a Map of indices.
java.util.List	<u>getRecordedEntriesWithLimit</u> (int type, long dvrTime, int limit) Given manifest type and DVR time, get a list of recorded entries limiting number of returned items.
<u>DvrManifestEntry</u>	<u>getRecordedEntryByIndex</u> (int type, int index) Get the manifest entry given a manifest type and an index.
<u>DvrManifestEntry</u>	<u>getRecordedEntryByTimeKey</u> (int type, long dvrTime) Get the manifest entry given a manifest type and a time (in DVR units)
<u>IDvrTimeMap</u>	<u>getTimeMap</u> ()
boolean	<u>hasAudio</u> () Does manifest contain audio.
boolean	<u>hasCodecData</u> () Does manifest contain codec data.
boolean	<u>hasData</u> () Does manifest contain data.
boolean	<u>hasOnMetadata</u> () Does manifest contain onMetadata.
boolean	<u>hasTimeMapData</u> () Does manifest contain time map info.

boolean	<code>hasVideo()</code> Does manifest contain video.
void	<code>importManifest(IDvrManifest manifest, boolean persist)</code> Import the specified manifest into this manifest
void	<code>initialize()</code> Initialize the manifest.
java.util.List	<code>purgeEntries(DvrManifestEntryRangeGroup ranges)</code> Given a group of ranges, purge the manifest entries.
void	<code>refreshManifest()</code> Refresh the manifest.
byte[]	<code>serialize(boolean ignoreEntries)</code> Serialize the manifest.

Fields

ON_METADATA_TYPE

public static final int **ON_METADATA_TYPE**

Constant for 'onMetadata' manifest type.
Constant value: **0**

CODEC_TYPE

public static final int **CODEC_TYPE**

Constant for codec manifest type.
Constant value: **999**

TIME_MAP_TYPE

public static final int **TIME_MAP_TYPE**

Constant for time map manifest type.
Constant value: **998**

MANIFEST_TAGNAME_CAN_RECORD

public static final java.lang.String **MANIFEST_TAGNAME_CAN_RECORD**

Constant value: **CanRecord**

MANIFEST_TAGNAME_CAN_PLAY

public static final java.lang.String **MANIFEST_TAGNAME_CAN_PLAY**

Constant value: **CanPlay**

(continued from last page)

MANIFEST_TAGNAME_HAS_ENCRYPTION

```
public static final java.lang.String MANIFEST_TAGNAME_HAS_ENCRYPTION
```

Constant value: **HasEncryption**

MANIFEST_TAGNAME_CHUNK_GROUPING

```
public static final java.lang.String MANIFEST_TAGNAME_CHUNK_GROUPING
```

Constant value: **ChunkGrouping**

MANIFEST_TAGNAME_PURGE_TIME

```
public static final java.lang.String MANIFEST_TAGNAME_PURGE_TIME
```

Constant value: **PurgeTime**

MANIFEST_TAGNAME_CURRENT_TIME

```
public static final java.lang.String MANIFEST_TAGNAME_CURRENT_TIME
```

Constant value: **CurrentTime**

Methods

initialize

```
public void initialize()
```

Initialize the manifest. Called after the previous storage has been loaded but before any chunks are added.

refreshManifest

```
public void refreshManifest()
```

Refresh the manifest.

addToManifest

```
public void addToManifest(java.util.List entries)
```

Add manifest entries to the manifest

Parameters:

entries - list of entries.

getRecordedEntries

```
public java.util.List getRecordedEntries(int type)
```

Get a copy of all manifest entries of a given type. Valid types include: [IVHost.CONTENTTYPE_AUDIO](#), [IVHost.CONTENTTYPE_VIDEO](#), [IVHost.CONTENTTYPE_DATA](#), [ON_METADATA_TYPE](#), [CODEC_TYPE](#), or [TIME_MAP_TYPE](#)

Parameters:

(continued from last page)

type - The manifest type.

Returns:

list of entries

getRecordedEntriesMap

```
public java.util.Map getRecordedEntriesMap(int type)
```

Get a copy of all manifest entries of a given type as a Map of indices. Valid types include: [IVHost.CONTENTTYPE_AUDIO](#), [IVHost.CONTENTTYPE_VIDEO](#), [IVHost.CONTENTTYPE_DATA](#), [ON_METADATA_TYPE](#), [CODEC_TYPE](#), or [TIME_MAP_TYPE](#)

Parameters:

type - the manifest type.

Returns:

map of entries

getRecordedEntriesInRange

```
public java.util.List getRecordedEntriesInRange(DvrManifestEntryRange range)
```

Given a DvrManifestEntryRange, get a list of recorded entries in this range.

Parameters:

range - a range (which consists of a manifest type and a start and end index)

Returns:

list of entries

getLastRecordedEntry

```
public DvrManifestEntry getLastRecordedEntry(int type)
```

Given manifest type and DVR time, get last entry.

"Recorded Entries" includes those entries after the end of the playlist.

Parameters:

type - the manifest type.

Returns:

manifest entry

getRecordedEntryByTimeKey

```
public DvrManifestEntry getRecordedEntryByTimeKey(int type,  
long dvrTime)
```

Get the manifest entry given a manifest type and a time (in DVR units)

Parameters:

type - the manifest type.
dvrTime - DVR time

Returns:

manifest entry

(continued from last page)

getRecordedEntryByIndex

```
public DvrManifestEntry getRecordedEntryByIndex(int type,  
int index)
```

Get the manifest entry given a manifest type and an index.

Parameters:

type - the manifest type.
index - the index into the manifest

Returns:

manifest entry

getEntriesToPurge

```
public DvrManifestEntryRangeGroup getEntriesToPurge(long purgeTime)
```

Given a purge time, return a group of manifest ranges to purge. This method ensures that the purge entries are aligned by index so that audio and video always remain index aligned.

Parameters:

purgeTime - The DVR time to purge

Returns:

a group of ranges.

purgeEntries

```
public java.util.List purgeEntries(DvrManifestEntryRangeGroup ranges)
```

Given a group of ranges, purge the manifest entries.

Parameters:

ranges - Group of ranges

Returns:

A list of the purged items.

getMetadataEntryForTime

```
public DvrManifestOnMetadataEntry getMetadataEntryForTime(long dvrTime)
```

Return most recent metadataEntry for a given DVR time. The manifest for metadata entries is sparse-- it only stores items periodically. So for a given time t, the returned entry will exist at or before time t.

Parameters:

dvrTime - DVR time

Returns:

onMetadata Entry

getCodecEntryForTime

```
public DvrManifestCodecEntry getCodecEntryForTime(long dvrTime)
```

Return most recent codec entry for a given DVR time. The manifest for codec entries is sparse-- it only stores items periodically. So for a given time t, the returned entry will exist at or before time t.

(continued from last page)

Parameters:

dvrTime - DVR time

Returns:

Codec Entry

getNumberLiveEntries

```
public int getNumberLiveEntries(int type,  
                                long dvrStart)
```

Given manifest type and DVR time, get the number of live entries at the given time.

"Live Entries" means that a couple of the last entries will not be included as they are too close to the live point to be part of the playlist.

Parameters:

type - the manifest type.

dvrStart - DVR time

Returns:

number of entries.

getNumberLiveEntries

```
public int getNumberLiveEntries(int type,  
                                long dvrStart,  
                                long dvrEnd)
```

Given manifest type and DVR time, get the number of live entries at the given time.

"Live Entries" means that a couple of the last entries will not be included as they are too close to the live point to be part of the playlist.

Parameters:

type - the manifest type.

dvrStart - DVR time

dvrEnd - end DVR time

Returns:

number of entries.

getNumberRecordedEntries

```
public int getNumberRecordedEntries(int type,  
                                    long dvrStart)
```

Given manifest type and DVR time, get the number of entries at the given time.

"Recorded Entries" includes those entries after the end of the playlist.

Parameters:

type - the manifest type.

dvrStart - DVR time

Returns:

number of entries.

(continued from last page)

getNumberRecordedEntries

```
public int getNumberRecordedEntries(int type,  
    long dvrStart,  
    long dvrEnd)
```

Given manifest type and DVR time, get the number of live entries at the given time.

"Recorded Entries" includes those entries after the end of the playlist.

Parameters:

type - the manifest type.
dvrStart - DVR time
dvrEnd - end DVR time

Returns:

number of entries.

getFirstEntry

```
public DvrManifestEntry getFirstEntry(int type)
```

Given manifest type, get the first playlist manifest entry.

Parameters:

type - the manifest type.

Returns:

manifest entry

getLastLiveEntry

```
public DvrManifestEntry getLastLiveEntry(int type)
```

Given manifest type, get the last playlist manifest entry.

"Live Entries" means that a couple of the last entries will not be included as they are too close to the live point to be part of the playlist.

Parameters:

type - the manifest type.

Returns:

manifest entry

getLiveEntries

```
public java.util.List getLiveEntries(int type,  
    long dvrStart)
```

Given manifest type and DVR time, get a list of the live entries.

"Live Entries" means that a couple of the last entries will not be included as they are too close to the live point to be part of the playlist.

Parameters:

type - the manifest type.
dvrStart - DVR time

Returns:

(continued from last page)

list of entries.

getLiveEntriesWithLimit

```
public java.util.List getLiveEntriesWithLimit(int type,
        long dvrTime,
        int limit)
```

Given manifest type and DVR time, get a list of the live entries limiting number of returned items.

"Live Entries" means that a couple of the last entries will not be included as they are too close to the live point to be part of the playlist.

Parameters:

type - the manifest type.
dvrTime - DVR time
limit - maximum number of returned items

Returns:

list of entries.

getRecordedEntriesWithLimit

```
public java.util.List getRecordedEntriesWithLimit(int type,
        long dvrTime,
        int limit)
```

Given manifest type and DVR time, get a list of recorded entries limiting number of returned items.

"Recorded Entries" includes those entries after the end of the playlist.

Parameters:

type - the manifest type.
dvrTime - DVR time
limit - maximum number of returned items

Returns:

list of entries.

getRecordedEntries

```
public java.util.List getRecordedEntries(int type,
        long dvrStartTime)
```

Given manifest type and DVR time, get a list of recorded entries.

"Recorded Entries" includes those entries after the end of the playlist.

Parameters:

type - the manifest type.
dvrStartTime - DVR time

Returns:

list of entries.

getRecordedEntries

```
public java.util.List getRecordedEntries(int type,
        long dvrStartTime,
        long dvrEndTime)
```

(continued from last page)

Given manifest type and DVR time, get a list of recorded entries.

"Recorded Entries" includes those entries after the end of the playlist.

Parameters:

type - the manifest type.
dvrStartTime - DVR time
dvrEndTime - DVR end time

Returns:

list of entries.

getRecordedDuration

```
public long getRecordedDuration(int type)
```

Given manifest type, return DVR recorded duration in seconds

"Recorded Entries" includes those entries after the end of the "live" playlist.

Parameters:

type - the manifest type.

Returns:

duration in seconds

getLiveDuration

```
public long getLiveDuration(int type)
```

Given manifest type, return DVR live duration in seconds

"Recorded Entries" includes those entries after the end of the "live" playlist.

Parameters:

type - the manifest type.

Returns:

duration in seconds

getDvrTime

```
public long getDvrTime(int type)
```

getLastRecordedIndex

```
public int getLastRecordedIndex(int type)
```

Return last index of DVR entry for given type.

Parameters:

type

Returns:

index, or -1 if type does not exist

(continued from last page)

getNextChunkIndex

```
public int getNextChunkIndex()
```

getNextMetadataIndex

```
public int getNextMetadataIndex()
```

getNextCodecIndex

```
public int getNextCodecIndex()
```

getNextTimeMapIndex

```
public int getNextTimeMapIndex()
```

getManifestChannel

```
public DvrChannelManifest getManifestChannel(int type)
```

For a given manifest type, return the channel manifest.

Parameters:

type

Returns:

channel manifest

importManifest

```
public void importManifest(IDvrManifest manifest,  
    boolean persist)
```

Import the specified manifest into this manifest

Parameters:

manifest

persist

getClosestStartTime

```
public long getClosestStartTime(int type,  
    long dvrTime)
```

Given a dvrTime and a manifest type, find the closest chunk starting time.

Parameters:

type - manifest type

dvrTime - DVR time

Returns:

(continued from last page)

time that corresponds to the closest start time of the given manifest type

expandEndTime

```
public long expandEndTime(int type,  
                           long dvrEndTime)
```

expandStartTime

```
public long expandStartTime(int type,  
                             long dvrStartTime)
```

hasVideo

```
public boolean hasVideo()
```

Does manifest contain video.

Returns:

true if it contains video.

hasAudio

```
public boolean hasAudio()
```

Does manifest contain audio.

Returns:

true if it contains audio.

hasData

```
public boolean hasData()
```

Does manifest contain data.

Returns:

true if it contains data.

hasOnMetadata

```
public boolean hasOnMetadata()
```

Does manifest contain onMetadata.

Returns:

true if it contains onMetadata.

hasCodecData

```
public boolean hasCodecData()
```

Does manifest contain codec data.

Returns:

(continued from last page)

true if it contains codec data

hasTimeMapData

```
public boolean hasTimeMapData( )
```

Does manifest contain time map info.

Returns:

true if it contains time map info.

serialize

```
public byte[] serialize(boolean ignoreEntries)
```

Serialize the manifest.

Parameters:

`ignoreEntries` - If true, the individual manifest entries are not serialized.

Returns:

byte array of serialized data

deserialize

```
public void deserialize(byte[] bytes)
```

Deserialize the manifest. Should be called after default constructor for manifest has been called.

Parameters:

`bytes` - buffer of bytes to deserialize

getTimeMap

```
public IDvrTimeMap getTimeMap( )
```

com.wowza.wms.dvr Interface IDvrPacketConverter

public interface **IDvrPacketConverter**
extends

Interface for creating DVR chunks from a set of packets.

Method Summary

DvrChunk	createDvrAudioChunk (long dvrTime, int index, int duration, DvrPacketHolder holder) Create audio chunk from set of packets.
DvrChunk	createDvrDataChunk (long dvrTime, int index, int duration, DvrPacketHolder holder) Create data chunk from set of packets.
DvrChunk	createDvrOnMetadataChunk (long dvrTime, long pt, AMFPacket metaPacket) Create onMetadata chunk from set of packets.
DvrChunk	createDvrVideoChunk (long dvrTime, int index, int duration, DvrPacketHolder holder) Create video chunk from set of packets.

Methods

createDvrAudioChunk

```
public DvrChunk createDvrAudioChunk(long dvrTime,  
    int index,  
    int duration,  
    DvrPacketHolder holder)
```

Create audio chunk from set of packets.

Parameters:

dvrTime - DVR start time in milliseconds
index - index of DVR chunk
duration - Duration in milliseconds
holder - packet holder

Returns:

the created chunk or null if chunk not created.

createDvrVideoChunk

```
public DvrChunk createDvrVideoChunk(long dvrTime,  
    int index,  
    int duration,  
    DvrPacketHolder holder)
```

Create video chunk from set of packets.

Parameters:

(continued from last page)

dvrTime - DVR start time in milliseconds
index - index of DVR chunk
duration - Duration in milliseconds
holder - packet holder

Returns:

the created chunk or null if chunk not created.

createDvrDataChunk

```
public DvrChunk createDvrDataChunk(long dvrTime,  
    int index,  
    int duration,  
    DvrPacketHolder holder)
```

Create data chunk from set of packets.

Parameters:

dvrTime - DVR start time in milliseconds
index - index of DVR chunk
duration - Duration in milliseconds
holder - packet holder

Returns:

the created chunk or null if chunk not created.

createDvrOnMetadataChunk

```
public DvrChunk createDvrOnMetadataChunk(long dvrTime,  
    long pt,  
    AMFPacket metaPacket)
```

Create onMetadata chunk from set of packets.

Parameters:

dvrTime - DVR start time in milliseconds
pt - Packet time
metaPacket - metadata packet

Returns:

the created chunk or null if chunk not created.

com.wowza.wms.dvr Interface IDvrPacketWriter

public interface **IDvrPacketWriter**
extends

Interface for chunking and storing packets for DVR.

Method Summary

void	<u>addToChunk</u> (DvrPacketHolder holder)
boolean	<u>canRecordAudio</u> ()
boolean	<u>canRecordData</u> ()
boolean	<u>canRecordVideo</u> ()
void	<u>endChunk</u> (long videoEndTime, long audioEndTime)
void	<u>resetStream</u> ()
void	<u>sendOnMetadata</u> (long pt, <u>AMFPacket</u> metaPacket)
void	<u>setCodecInfoAudio</u> (com.wowza.wms.media.model.MediaCodecInfoAudio codecInfoAudio)
void	<u>setCodecInfoVideo</u> (com.wowza.wms.media.model.MediaCodecInfoVideo codecInfoVideo)
void	<u>startChunk</u> (int streamMode, int videoCodecId, int audioCodecId, long timecode, long utcTimecode)

Methods

sendOnMetadata

```
public void sendOnMetadata(long pt,
    AMFPacket metaPacket)
```

startChunk

```
public void startChunk(int streamMode,
    int videoCodecId,
    int audioCodecId,
    long timecode,
    long utcTimecode)
```

(continued from last page)

addToChunk

```
public void addToChunk(DvrPacketHolder holder)
```

endChunk

```
public void endChunk(long videoEndTime,  
                    long audioEndTime)
```

setCodecInfoAudio

```
public void setCodecInfoAudio(com.wowza.wms.media.model.MediaCodecInfoAudio  
codecInfoAudio)
```

setCodecInfoVideo

```
public void setCodecInfoVideo(com.wowza.wms.media.model.MediaCodecInfoVideo  
codecInfoVideo)
```

resetStream

```
public void resetStream()
```

canRecordAudio

```
public boolean canRecordAudio()
```

canRecordVideo

```
public boolean canRecordVideo()
```

canRecordData

```
public boolean canRecordData()
```

com.wowza.wms.dvr Interface IDvrPlayerAdapter

public interface **IDvrPlayerAdapter**
extends

Field Summary

public static final	QUERY_PARAM_DVR Value: DVR
---------------------	--

Method Summary

int	calcBitrate (IDvrStreamStore store, int type, long t)
String	determineStreamVersion (IDvrStreamManager dvrMgr, String baseStreamName)
DvrPlaylistRequest	getDvrPlaylistRequest (IHTTPStreamerApplicationContext appContext, IDvrStreamStore store, java.util.Map queryMap)
DvrPlaylistRequest	getDvrPlaylistRequest (IHTTPStreamerApplicationContext appContext, java.util.List stores, java.util.Map queryMap)
IDvrMbrPlaylistAlignment	getPlaylistAlignment (IHTTPStreamerSession httpStreamerSession)
boolean	isPlaylistReady (IDvrStreamStore store, DvrPlaylistRequest playlistRequest)
boolean	isPlaylistReady (java.util.List stores, DvrPlaylistRequest dvrPlaylistRequest)

Fields

QUERY_PARAM_DVR

public static final java.lang.String **QUERY_PARAM_DVR**

Constant value: **DVR**

Methods

isPlaylistReady

public boolean **isPlaylistReady**([IDvrStreamStore](#) store, DvrPlaylistRequest playlistRequest)

isPlaylistReady

```
public boolean isPlaylistReady(java.util.List stores,
                                DvrPlaylistRequest dvrPlaylistRequest)
```

determineStreamVersion

```
public String determineStreamVersion(IDvrStreamManager dvrMgr,
                                       String baseStreamName)
```

calcBitrate

```
public int calcBitrate(IDvrStreamStore store,
                       int type,
                       long t)
```

getDvrPlaylistRequest

```
public DvrPlaylistRequest getDvrPlaylistRequest(IHTTPStreamerApplicationContext
appContext,
IDvrStreamStore store,
java.util.Map queryMap)
```

getDvrPlaylistRequest

```
public DvrPlaylistRequest getDvrPlaylistRequest(IHTTPStreamerApplicationContext
appContext,
java.util.List stores,
java.util.Map queryMap)
```

getPlaylistAlignment

```
public IDvrMbrPlaylistAlignment getPlaylistAlignment(IHTTPStreamerSession
httpStreamerSession)
```

com.wowza.wms.dvr Interface IDvrPrivateConstants

All Subinterfaces:

[IDvrConstants](#)

public interface **IDvrPrivateConstants**
extends

Internal and public constants. Public constants are contained in [IDvrConstants](#). Undocumented/Private in this file. To make the constant public (and documented in javadocs, simply move to IDvrConstants)

Field Summary	
public static final	CHUNK_FILENAME_FORMAT_AUDIO Value: A-%04d_%02d_%02d_%03d.m4fa
public static final	CHUNK_FILENAME_FORMAT_DATA Value: D-%04d_%02d_%02d_%03d.m4fd
public static final	CHUNK_FILENAME_FORMAT_METADATA Value: M-%04d_%02d_%02d_%03d.m4fm
public static final	CHUNK_FILENAME_FORMAT_VIDEO Value: V-%04d_%02d_%02d_%03d.m4fv
public static final	CHUNK_FOLDER_FORMAT Value: %04d_%02d_%02d
public static final	DEFAULT_PROPERTY_AUDIO_GROUP_COUNT Default value for DVR Property "dvrAudioGroupCount". Value: 3
public static final	DEFAULT_PROPERTY_MANIFEST_PURGE_SIZE Default value for DVR Property "dvrManifestPurgeCacheSize". Value: 10
public static final	DEFAULT_PROPERTY_MAX_ALLOWABLE_CHUNK_DURATION Default value for DVR Property "dvrMaxAllowableChunkDuration". Value: 120000
public static final	DEFAULT_PROPERTY_MAX_SAVE HOLDER_SIZE Value: 500
public static final	DEFAULT_PROPERTY_MINIMUM_AVAILABLE_CHUNKS Default value for DVR Property "dvrMinimumAvailableChunks". Value: 3
public static final	DEFAULT_PROPERTY_PURGE_CONTROL_CLASS Default value for DVR Property "dvrPurgeControlClass". Value: com.wowza.wms.dvr.impl.DvrPurgeController

public static final	DEFAULT_PROPERTY_RECALC_DURATION_MAX_AV_DIFFERENCE_TRIGGER Default value for DVR Property "dvrRecalcDurationMaxAVStartDifference". Value: 500
public static final	DEFAULT_PROPERTY_RECALC_DURATION_TRIGGER_SIZE Default value for DVR Property "dvrRecalcDurationMaxChunkSize". Value: 60000
public static final	DEFAULT_PROPERTY_STREAM_IDLE_TIMEOUT Default value for DVR Property "streamTimeout". Value: 300000
public static final	DEFAULT_PROPERTY_STREAM_STARTUP_TIMEOUT Default value for DVR Property "streamStartupTimeout". Value: 60000
public static final	DEFAULT_PROPERTY_UTC_ALIGN_TO_AUDIO Default value for DVR Property "dvrUtcAlignToAudio". Value: true
public static final	DEFAULT_PROPERTY_UTC_USE_WOWZA_PACKET_ARRIVAL Default value for DVR Property "dvrUtcUseWowzaPacketArrivalTime". Value: false
public static final	DEFAULT_PROPERTY_WAIT_FOR_CODEC_TIME Default value for DVR Property "dvrWaitForCodecTime". Value: 8000
public static final	PROPERTY_AUDIO_GROUP_COUNT DVR Property "dvrAudioGroupCount": for number of audio packets to group together. Value: dvrAudioGroupCount
public static final	PROPERTY_DEBUG_MBR_RESOLVER Value: dvrDebugMbrResolver
public static final	PROPERTY_DEBUG_MBR_RESOLVER_GAP_DETAILS Value: dvrDebugMbrResolverGapDetails
public static final	PROPERTY_DEBUG_MBR_RESOLVER_INTERSECTION_DETAILS Value: dvrDebugMbrResolverIntersectionDetails
public static final	PROPERTY_INTERCEPTOR_CHUNK_DELEGATE DVR Property "dvrChunkInterceptorDelegate": for specifying a delegate that determines if DVR chunk should be stored or not. Value: dvrChunkInterceptorDelegate
public static final	PROPERTY_MANIFEST_PURGE_CACHE_SIZE DVR Property "dvrManifestPurgeCacheSize": for specifying how many purged chunks are kept around just in case they are requested. Value: dvrManifestPurgeCacheSize
public static final	PROPERTY_MAX_ALLOWABLE_CHUNK_DURATION DVR Property "dvrMaxAllowableChunkDuration": for setting maximum chunk duration. Value: dvrMaxAllowableChunkDuration

public static final	<u>PROPERTY_MAX_SAVE_HOLDER_SIZE</u> Value: dvrMaxSaveHolderSize
public static final	<u>PROPERTY_MBR_USE_SIMPLEALIGNMENT</u> HTTP Streamer Property "dvrMbrUseSimpleAligner": for defaulting to a simple mbr aligner Value: dvrMbrUseSimpleAligner
public static final	<u>PROPERTY_MINIMUM_AVAILABLE_CHUNKS</u> DVR Property "dvrMinimumAvailableChunks": for specifying how many chunks must be available in order to play back the DVR recording. Value: dvrMinimumAvailableChunks
public static final	<u>PROPERTY_PURGE_CONTROL_CLASS</u> DVR Property "dvrPurgeControlClass": for controlling the class responsible for determining which chunks to purge Value: dvrPurgeControlClass
public static final	<u>PROPERTY_RECALC_DURATION_MAX_AV_DIFFERENCE_TRIGGER</u> DVR Property "dvrRecalcDurationMaxAVStartDifference": for triggering recalculation of DVR chunk duration by examining each packet Value: dvrRecalcDurationMaxAVStartDifference
public static final	<u>PROPERTY_RECALC_DURATION_TRIGGER_SIZE</u> DVR Property "dvrRecalcDurationMaxChunkSize": for triggering recalculation of DVR chunk duration by examining each packet Value: dvrRecalcDurationMaxChunkSize
public static final	<u>PROPERTY_UTC_ALIGN_TO_AUDIO</u> DVR Property "dvrUtcAlignToAudio": determines if calculated Utc time gets aligned to audio or video. Value: dvrUtcAlignToAudio
public static final	<u>PROPERTY_UTC_DELEGATE_CLASS</u> DVR Property "dvrUtcTimeDelegate": for specifying a delegate that the utc time for a given chunk. Value: dvrUtcTimeDelegate
public static final	<u>PROPERTY_UTC_USE_WOWZA_PACKET_ARRIVAL</u> DVR Property "dvrUtcUseWowzaPacketArrivalTime": determines if Utc times are based on the time the packet reaches Wowza or the time the chunks reach DVR (default). Value: dvrUtcUseWowzaPacketArrivalTime
public static final	<u>PROPERTY_VERIFY_CHUNK_EXISTENCE_ON_MANIFEST_LOAD</u> Value: dvrVerifyChunkExistenceOnManifestLoad
public static final	<u>PROPERTY_WAIT_FOR_CODEC_TIME</u> DVR Property "dvrWaitForCodecTime": for how long DVR will wait to get the codec information. Value: dvrWaitForCodecTime

Fields

(continued from last page)

PROPERTY_INTERCEPTOR_CHUNK_DELEGATE

```
public static final java.lang.String PROPERTY_INTERCEPTOR_CHUNK_DELEGATE
```

DVR Property "dvrChunkInterceptorDelegate": for specifying a delegate that determines if DVR chunk should be stored or not. It's only here to allow a backdoor method of testing mbr streams that have gaps.
Constant value: **dvrChunkInterceptorDelegate**

PROPERTY_UTC_DELEGATE_CLASS

```
public static final java.lang.String PROPERTY_UTC_DELEGATE_CLASS
```

DVR Property "dvrUtcTimeDelegate": for specifying a delegate that the utc time for a given chunk. Currently in private constants until article is written.
Constant value: **dvrUtcTimeDelegate**

PROPERTY_UTC_ALIGN_TO_AUDIO

```
public static final java.lang.String PROPERTY_UTC_ALIGN_TO_AUDIO
```

DVR Property "dvrUtcAlignToAudio": determines if calculated Utc time gets aligned to audio or video. Note: Currently in private constants until article is written.
Constant value: **dvrUtcAlignToAudio**

DEFAULT_PROPERTY_UTC_ALIGN_TO_AUDIO

```
public static final boolean DEFAULT_PROPERTY_UTC_ALIGN_TO_AUDIO
```

Default value for DVR Property "dvrUtcAlignToAudio". Default value is true.
Constant value: **true**

See Also:

[PROPERTY_UTC_ALIGN_TO_AUDIO](#)

PROPERTY_UTC_USE_WOWZA_PACKET_ARRIVAL

```
public static final java.lang.String PROPERTY_UTC_USE_WOWZA_PACKET_ARRIVAL
```

DVR Property "dvrUtcUseWowzaPacketArrivalTime": determines if Utc times are based on the time the packet reaches Wowza or the time the chunks reach DVR (default). Note: Currently in private constants until article is written.
Constant value: **dvrUtcUseWowzaPacketArrivalTime**

DEFAULT_PROPERTY_UTC_USE_WOWZA_PACKET_ARRIVAL

```
public static final boolean DEFAULT_PROPERTY_UTC_USE_WOWZA_PACKET_ARRIVAL
```

Default value for DVR Property "dvrUtcUseWowzaPacketArrivalTime". Default value is false.
Constant value: **false**

See Also:

[PROPERTY_UTC_USE_WOWZA_PACKET_ARRIVAL](#)

PROPERTY_MINIMUM_AVAILABLE_CHUNKS

```
public static final java.lang.String PROPERTY_MINIMUM_AVAILABLE_CHUNKS
```

(continued from last page)

DVR Property "dvrMinimumAvailableChunks": for specifying how many chunks must be available in order to play back the DVR recording.

It is not recommended to have this set to less than three, as players typically require this many chunks.

Add the property to Application/DVR/Properties section of Application.xml.

Constant value: **dvrMinimumAvailableChunks**

See Also:

[DEFAULT_PROPERTY_MINIMUM_AVAILABLE_CHUNKS](#)

DEFAULT_PROPERTY_MINIMUM_AVAILABLE_CHUNKS

```
public static final int DEFAULT_PROPERTY_MINIMUM_AVAILABLE_CHUNKS
```

Default value for DVR Property "dvrMinimumAvailableChunks".

Default value is 3.

Constant value: **3**

See Also:

[PROPERTY_MINIMUM_AVAILABLE_CHUNKS](#)

PROPERTY_MANIFEST_PURGE_CACHE_SIZE

```
public static final java.lang.String PROPERTY_MANIFEST_PURGE_CACHE_SIZE
```

DVR Property "dvrManifestPurgeCacheSize": for specifying how many purged chunks are kept around just in case they are requested.

When purging chunks occurs, the DVR keeps a few chunks around that normally would have been purged just in case they are requested.

Add the property to Application/DVR/Properties section of Application.xml.

Constant value: **dvrManifestPurgeCacheSize**

See Also:

[DEFAULT_PROPERTY_MANIFEST_PURGE_SIZE](#)

DEFAULT_PROPERTY_MANIFEST_PURGE_SIZE

```
public static final int DEFAULT_PROPERTY_MANIFEST_PURGE_SIZE
```

Default value for DVR Property "dvrManifestPurgeCacheSize".

Default value is 10.

Constant value: **10**

See Also:

[PROPERTY_MANIFEST_PURGE_CACHE_SIZE](#)

PROPERTY_VERIFY_CHUNK_EXISTENCE_ON_MANIFEST_LOAD

```
public static final java.lang.String PROPERTY_VERIFY_CHUNK_EXISTENCE_ON_MANIFEST_LOAD
```

Constant value: **dvrVerifyChunkExistenceOnManifestLoad**

PROPERTY_PURGE_CONTROL_CLASS

```
public static final java.lang.String PROPERTY_PURGE_CONTROL_CLASS
```

(continued from last page)

DVR Property "dvrPurgeControlClass": for controlling the class responsible for determining which chunks to purge

The class must implement [IDvrPurgeController](#) and may sub-class DvrPurgeControllerBase

Add the property to Application/DVR/Properties section of Application.xml.

Constant value: **dvrPurgeControlClass**

See Also:

[IDvrPurgeController](#)

[DEFAULT_PROPERTY_PURGE_CONTROL_CLASS](#)

DEFAULT_PROPERTY_PURGE_CONTROL_CLASS

```
public static final java.lang.String DEFAULT_PROPERTY_PURGE_CONTROL_CLASS
```

Default value for DVR Property "dvrPurgeControlClass".

Default value is "com.wowza.wms.dvr.impl.DvrPurgeController".

Constant value: **com.wowza.wms.dvr.impl.DvrPurgeController**

See Also:

[PROPERTY_PURGE_CONTROL_CLASS](#)

PROPERTY_MBR_USE_SIMPLEALIGNMENT

```
public static final java.lang.String PROPERTY_MBR_USE_SIMPLEALIGNMENT
```

HTTP Streamer Property "dvrMbrUseSimpleAligner": for defaulting to a simple mbr aligner

Add the property to Application/HTTPStreamer/Properties section of Application.xml.

Constant value: **dvrMbrUseSimpleAligner**

PROPERTY_MAX_ALLOWABLE_CHUNK_DURATION

```
public static final java.lang.String PROPERTY_MAX_ALLOWABLE_CHUNK_DURATION
```

DVR Property "dvrMaxAllowableChunkDuration": for setting maximum chunk duration.

Add the property to Application/DVR/Properties section of Application.xml

Constant value: **dvrMaxAllowableChunkDuration**

See Also:

[DEFAULT_PROPERTY_MAX_ALLOWABLE_CHUNK_DURATION](#)

DEFAULT_PROPERTY_MAX_ALLOWABLE_CHUNK_DURATION

```
public static final int DEFAULT_PROPERTY_MAX_ALLOWABLE_CHUNK_DURATION
```

Default value for DVR Property "dvrMaxAllowableChunkDuration".

Default value is 120000 ms (120 seconds).

Constant value: **120000**

See Also:

[PROPERTY_MAX_ALLOWABLE_CHUNK_DURATION](#)

PROPERTY_RECALC_DURATION_MAX_AV_DIFFERENCE_TRIGGER

```
public static final java.lang.String  
PROPERTY_RECALC_DURATION_MAX_AV_DIFFERENCE_TRIGGER
```

(continued from last page)

DVR Property "dvrRecalcDurationMaxAVStartDifference": for triggering recalculation of DVR chunk duration by examining each packet

This is a fallback setting for when the aligned audio and video packets differ their start times. When this happens, fallback logic is triggered which examines the actual packets and tries to adjust the packet times and duration accordingly.

Add the property to Application/DVR/Properties section of Application.xml
Constant value: **dvrRecalcDurationMaxAVStartDifference**

See Also:

[DEFAULT_PROPERTY_RECALC_DURATION_MAX_AV_DIFFERENCE_TRIGGER](#)

DEFAULT_PROPERTY_RECALC_DURATION_MAX_AV_DIFFERENCE_TRIGGER

```
public static final int DEFAULT_PROPERTY_RECALC_DURATION_MAX_AV_DIFFERENCE_TRIGGER
```

Default value for DVR Property "dvrRecalcDurationMaxAVStartDifference".

Default value is 500 ms.
Constant value: **500**

See Also:

[PROPERTY_RECALC_DURATION_MAX_AV_DIFFERENCE_TRIGGER](#)

PROPERTY_RECALC_DURATION_TRIGGER_SIZE

```
public static final java.lang.String PROPERTY_RECALC_DURATION_TRIGGER_SIZE
```

DVR Property "dvrRecalcDurationMaxChunkSize": for triggering recalculation of DVR chunk duration by examining each packet

This is a fallback setting for when the DVR packetizer determines the chunk is quite large, usually because the incoming packet times have done something odd like suddenly diverged. When this happens, asome fallback logic is triggered which examines the acual packets and tries to adjust the duration accordingly.

Add the property to Application/DVR/Properties section of Application.xml
Constant value: **dvrRecalcDurationMaxChunkSize**

See Also:

[DEFAULT_PROPERTY_RECALC_DURATION_TRIGGER_SIZE](#)

DEFAULT_PROPERTY_RECALC_DURATION_TRIGGER_SIZE

```
public static final int DEFAULT_PROPERTY_RECALC_DURATION_TRIGGER_SIZE
```

Default value for DVR Property "dvrRecalcDurationMaxChunkSize".

Default value is 60000 ms (60 seconds).
Constant value: **60000**

See Also:

[PROPERTY_RECALC_DURATION_TRIGGER_SIZE](#)

PROPERTY_MAX_SAVE_HOLDER_SIZE

```
public static final java.lang.String PROPERTY_MAX_SAVE_HOLDER_SIZE
```

Constant value: **dvrMaxSaveHolderSize**

(continued from last page)

DEFAULT_PROPERTY_MAX_SAVE_HOLDER_SIZE

```
public static final int DEFAULT_PROPERTY_MAX_SAVE_HOLDER_SIZE
```

Constant value: **500**

PROPERTY_WAIT_FOR_CODEC_TIME

```
public static final java.lang.String PROPERTY_WAIT_FOR_CODEC_TIME
```

DVR Property "dvrWaitForCodecTime": for how long DVR will wait to get the codec information.

Amount of time Wowza nDVR will wait to receive audio and video codec information after streaming has begun. The unit is in ms. The default value is 8000 ms.

Add the property to Application/DVR/Properties section of Application.xml

Constant value: **dvrWaitForCodecTime**

See Also:

[DEFAULT_PROPERTY_WAIT_FOR_CODEC_TIME](#)

DEFAULT_PROPERTY_WAIT_FOR_CODEC_TIME

```
public static final int DEFAULT_PROPERTY_WAIT_FOR_CODEC_TIME
```

Default value for DVR Property "dvrWaitForCodecTime".

Default value is 8000 ms.

Constant value: **8000**

See Also:

[PROPERTY_WAIT_FOR_CODEC_TIME](#)

PROPERTY_AUDIO_GROUP_COUNT

```
public static final java.lang.String PROPERTY_AUDIO_GROUP_COUNT
```

DVR Property "dvrAudioGroupCount": for number of audio packets to group together.

Default is 3. @see [DEFAULT_PROPERTY_AUDIO_GROUP_COUNT](#)

Constant value: **dvrAudioGroupCount**

DEFAULT_PROPERTY_AUDIO_GROUP_COUNT

```
public static final int DEFAULT_PROPERTY_AUDIO_GROUP_COUNT
```

Default value for DVR Property "dvrAudioGroupCount".

Default value is 3.

Constant value: **3**

See Also:

[PROPERTY_AUDIO_GROUP_COUNT](#)

DEFAULT_PROPERTY_STREAM_IDLE_TIMEOUT

```
public static final int DEFAULT_PROPERTY_STREAM_IDLE_TIMEOUT
```

(continued from last page)

Default value for DVR Property "streamTimeout".

Default value is 300000 ms (5 minutes).
Constant value: **300000**

DEFAULT_PROPERTY_STREAM_STARTUP_TIMEOUT

```
public static final int DEFAULT_PROPERTY_STREAM_STARTUP_TIMEOUT
```

Default value for DVR Property "streamStartupTimeout".

Default value is 60000 ms (60 seconds).
Constant value: **60000**

PROPERTY_DEBUG_MBR_RESOLVER

```
public static final java.lang.String PROPERTY_DEBUG_MBR_RESOLVER
```

Constant value: **dvrDebugMbrResolver**

PROPERTY_DEBUG_MBR_RESOLVER_GAP_DETAILS

```
public static final java.lang.String PROPERTY_DEBUG_MBR_RESOLVER_GAP_DETAILS
```

Constant value: **dvrDebugMbrResolverGapDetails**

PROPERTY_DEBUG_MBR_RESOLVER_INTERSECTION_DETAILS

```
public static final java.lang.String PROPERTY_DEBUG_MBR_RESOLVER_INTERSECTION_DETAILS
```

Constant value: **dvrDebugMbrResolverIntersectionDetails**

CHUNK_FILENAME_FORMAT_METADATA

```
public static final java.lang.String CHUNK_FILENAME_FORMAT_METADATA
```

Constant value: **M-%04d_%02d_%02d_%03d.m4fm**

CHUNK_FILENAME_FORMAT_VIDEO

```
public static final java.lang.String CHUNK_FILENAME_FORMAT_VIDEO
```

Constant value: **V-%04d_%02d_%02d_%03d.m4fv**

CHUNK_FILENAME_FORMAT_AUDIO

```
public static final java.lang.String CHUNK_FILENAME_FORMAT_AUDIO
```

Constant value: **A-%04d_%02d_%02d_%03d.m4fa**

CHUNK_FILENAME_FORMAT_DATA

```
public static final java.lang.String CHUNK_FILENAME_FORMAT_DATA
```

(continued from last page)

Constant value: **D-%04d_%02d_%02d_%03d.m4fd**

CHUNK_FOLDER_FORMAT

```
public static final java.lang.String CHUNK_FOLDER_FORMAT
```

Constant value: **%04d_%02d_%02d**

com.wowza.wms.dvr Interface IDvrPurgeController

public interface **IDvrPurgeController**
extends

Interface for controlling DVR chunk purging. The purge controller is instantiated using a factory class DvrPurgeControllerFactory.
See Also:

[com.wowza.wms.dvr.impl.DvrPurgeControllerFactory](#),
[IDvrPrivateConstants.PROPERTY_PURGE_CONTROL_CLASS](#)

Method Summary

long	getCurrentTime() Get current DVR time.
long	getLastPurgeTime() Get the DVR time when the last purge occurred.
void	init(IDvrStreamStore store) Initialize the controller.
boolean	isPurgingEnabled() Is Purging enabled for this controller.
void	setCurrentDvrTime(long newDvrTime) Set the current DVR time and perform purge if necessary.

Methods

init

public void **init**([IDvrStreamStore](#) store)

Initialize the controller.

Parameters:

store - the DVR stream store

isPurgingEnabled

public boolean **isPurgingEnabled**()

Is Purging enabled for this controller.

Returns:

true if purging is active, false otherwise.

setCurrentDvrTime

public void **setCurrentDvrTime**(long newDvrTime)

Set the current DVR time and perform purge if necessary. This method checks the DVR time against its internal rules for purging and performs the purge. It is also responsible for setting the last purge time and the next purge time.

(continued from last page)

Parameters:newDvrTime

getCurrentTime

```
public long getCurrentTime()
```

Get current DVR time. The store is responsible for setting the DVR time via setCurrentDvrTime.

Returns:current DVR time.

getLastPurgeTime

```
public long getLastPurgeTime()
```

Get the DVR time when the last purge occurred.

Returns:DVR time of last purge

com.wowza.wms.dvr Interface IDvrRawChunkProvider

All Subinterfaces:

[IDvrChunkMemoryCache](#)

public interface **IDvrRawChunkProvider**
extends

Interface for providing raw DVR chunks based on a DVR manifest entry.

Method Summary

DvrChunk	retrieveRawChunk (DvrManifestChunkEntry entry) Given the specified manifest entry, return the raw DVR chunk.
----------	--

Methods

retrieveRawChunk

public DvrChunk **retrieveRawChunk**([DvrManifestChunkEntry](#) entry)

Given the specified manifest entry, return the raw DVR chunk.

Parameters:

entry - The DVR manifest entry

Returns:

the DVR chunk or null if not able to return the chunk.

com.wowza.wms.dvr Interface IDvrRecordingListener

public interface **IDvrRecordingListener**
extends

Listener for DVR Recording events.

See Also:

[IDvrStreamManager.addDvrRecordingListener\(IDvrRecordingListener\)](#),
[IDvrStreamManager.removeDvrRecordingListener\(IDvrRecordingListener\)](#)

Method Summary

void	recordingPaused (IDvrStreamStore store) Callback when DVR moves to paused state.
void	recordingReset (IDvrStreamStore store) Callback when DVR gets reset.
void	recordingResumed (IDvrStreamStore store) Callback when DVR moves out of paused state.
void	recordingStarted (IDvrStreamStore store) Callback when DVR moves to recording state.
void	recordingStopped (IDvrStreamStore store) Callback when DVR recording stops
void	timeReset (IDvrStreamStore store, long oldDvrTime, long oldPacketTime, TimeMapRecord newTime) Callback when DVR time is adjusted.

Methods

recordingStarted

public void **recordingStarted**([IDvrStreamStore](#) store)

Callback when DVR moves to recording state.

Parameters:

store - stream store

recordingPaused

public void **recordingPaused**([IDvrStreamStore](#) store)

Callback when DVR moves to paused state.

Parameters:

store - stream store

(continued from last page)

recordingResumed

```
public void recordingResumed(IDvrStreamStore store)
```

Callback when DVR moves out of paused state.

Parameters:

store - stream store

recordingReset

```
public void recordingReset(IDvrStreamStore store)
```

Callback when DVR gets reset. For example, if the incoming stream resets.

Parameters:

store - stream store

recordingStopped

```
public void recordingStopped(IDvrStreamStore store)
```

Callback when DVR recording stops

Parameters:

store - stream store

timeReset

```
public void timeReset(IDvrStreamStore store,  
    long oldDvrTime,  
    long oldPacketTime,  
    TimeMapRecord newTime)
```

Callback when DVR time is adjusted.

Parameters:

store - stream store

oldDvrTime

oldPacketTime - old packet time associated to oldDvrTime

newTime

com.wowza.wms.dvr Interface IDvrRecordingsLoader

public interface **IDvrRecordingsLoader**
extends

Interface for loading recordings during DVR Stream Manager initialization.

See Also:

[com.wowza.wms.dvr.impl.DvrRecordingsLoaderFactory](#),
[IDvrConstants.PROPERTY_RECORDINGS_LOADER_CLASS](#)

Method Summary

void	init (IDvrStreamManager dvrMgr) Initialize recordings loader
void	loadArchivedRecordings () Discover and load archived recording
boolean	shouldLoadStream (String streamName, java.util.SortedSet versions) Should the given archived streams be laoded.
boolean	shouldLoadStreamVersion (String streamName, Integer version, java.util.SortedSet versions) Should the given archived stream of specific version be laoded.

Methods

init

public void **init**([IDvrStreamManager](#) dvrMgr)

Initialize recordings loader

Parameters:

dvrMgr - the DVR Stream Manager

loadArchivedRecordings

public void **loadArchivedRecordings**()

Discover and load archived recording

shouldLoadStream

public boolean **shouldLoadStream**(String streamName,
java.util.SortedSet versions)

Should the given archived streams be laoded. The method contains logic which determines if all versions of the given stream Name should be loaded or not.

Parameters:

streamName - Stream name (unversioned)

versions - sorted set of available versions of this stream

(continued from last page)

Returns:

true if one or more of the streams should be loaded, false otherwise.

shouldLoadStreamVersion

```
public boolean shouldLoadStreamVersion(String streamName,  
    Integer version,  
    java.util.SortedSet versions)
```

Should the given archived stream of specific version be loaded. The method contains logic which determines if all versions of the given stream Name should be loaded or not.

Parameters:

streamName - Stream name (unversioned)

version - the specific version of the stream we are determining if we should load

versions - sorted set of all available versions of this stream

Returns:

true if one or more of the streams should be loaded, false otherwise.

com.wowza.wms.dvr Interface IDvrStoreChunkListener

public interface **IDvrStoreChunkListener**
extends

Listener for DVR Store chunk events.

Note that this listener traffic is very high.

See Also:

[IDvrStreamStore.addDvrChunkListener\(IDvrStoreChunkListener\)](#),
[IDvrStreamStore.removeDvrChunkListener\(IDvrStoreChunkListener\)](#)

Method Summary

void	postChunkAdded (IDvrStreamStore store) Callback after a chunk is written to DVR store
void	postChunksPurged (IDvrStreamStore store, DvrManifestEntryRangeGroup entries, java.util.List deletedEntries) Callback after DVR store performs a purge
void	preChunkAdded (IDvrStreamStore store) Callback before a chunk is written to DVR store
void	preChunksPurged (IDvrStreamStore store, DvrManifestEntryRangeGroup entries) Callback before DVR store performs a purge

Methods

preChunkAdded

public void **preChunkAdded**([IDvrStreamStore](#) store)

Callback before a chunk is written to DVR store

Parameters:

store - the stream store

postChunkAdded

public void **postChunkAdded**([IDvrStreamStore](#) store)

Callback after a chunk is written to DVR store

Parameters:

store - the stream store

preChunksPurged

public void **preChunksPurged**([IDvrStreamStore](#) store,
[DvrManifestEntryRangeGroup](#) entries)

(continued from last page)

Callback before DVR store performs a purge

Parameters:

store - the stream store

entries - list of DVR chunks to be purged

postChunksPurged

```
public void postChunksPurged(IDvrStreamStore store,  
    DvrManifestEntryRangeGroup entries,  
    java.util.List deletedEntries)
```

Callback after DVR store performs a purge

Parameters:

store - the stream store

entries - list of DVR chunks purged

deletedEntries - list of entries deleted

com.wowza.wms.dvr Interface IDvrStoreListener

public interface **IDvrStoreListener**
extends

Listener for DVR Store lifecycle events.

See Also:

[IDvrStreamManager.addDvrStoreListener\(IDvrStoreListener\)](#),
[IDvrStreamManager.removeDvrStoreListener\(IDvrStoreListener\)](#)

Method Summary

void	dvrStreamStorageDeleted (IDvrStreamStore store, IDvrFileSystem FileSystem , boolean success) Callback after DVR store is deleted from disk.
void	dvrStreamStorageLoaded (IDvrStreamManager dvrMgr, IDvrStreamStore store) Callback after DVR store is loaded from disk.
void	dvrStreamStoreCreate (IDvrStreamStore store) Callback after DVR stream store is created.
void	dvrStreamStoreDestroy (IDvrStreamStore store) Callback after DVR stream store is destroyed.
void	dvrStreamStoreInit (IDvrStreamStore store) Callback after DVR stream store is initialized.

Methods

dvrStreamStoreCreate

public void **dvrStreamStoreCreate**([IDvrStreamStore](#) store)

Callback after DVR stream store is created.

Parameters:

store - stream store

dvrStreamStoreInit

public void **dvrStreamStoreInit**([IDvrStreamStore](#) store)

Callback after DVR stream store is initialized.

Parameters:

store - stream store

dvrStreamStoreDestroy

public void **dvrStreamStoreDestroy**([IDvrStreamStore](#) store)

Callback after DVR stream store is destroyed.

(continued from last page)

Parameters:store - stream store

dvrStreamStorageLoaded

```
public void dvrStreamStorageLoaded(IDvrStreamManager dvrMgr,  
    IDvrStreamStore store)
```

Callback after DVR store is loaded from disk.

Parameters:

dvrMgr - DVR Application Store Manager

store - stream store

dvrStreamStorageDeleted

```
public void dvrStreamStorageDeleted(IDvrStreamStore store,  
    IDvrFileSystem fileSystem,  
    boolean success)
```

Callback after DVR store is deleted from disk.

Parameters:

store

fileSystem - stream store file system

success - whether delete succeeded or failed

com.wowza.wms.dvr Interface IDvrStreamManager

All Superinterfaces:

[ILiveStreamPacketizer](#)

public interface **IDvrStreamManager**

extends [ILiveStreamPacketizer](#)

DVR Stream manager. Manages a live stream and associated DVR stores.

Method Summary

void	addDvrRecordingListener (IDvrRecordingListener listener)	Add listener to DVR recording events.
void	addDvrStoreListener (IDvrStoreListener listener)	Add listener to DVR store lifecycle events.
void	addManifestEntries (String vStreamName, java.util.List entries, String serverGuid)	Add to stream stores manifest.
void	addRepeaterHeartBeatItem ()	
boolean	canRecord ()	Is this stream able to record.
void	deleteArchivedStore (String vStreamName)	Delete archived stream store.
String	getArchiveStrategy ()	Get the DVR archive strategy.
String	getContextStr ()	Get stream context string, useful for logging.
IDvrStreamStore	getDefaultStreamingStore ()	Get the store to be used for streaming.
String	getDvrFileSystemClass ()	Get the class used for the DVR file system.
String	getDvrStorageDir ()	Get the storage directory.
int	getDvrStorageWindowSeconds ()	Get DVR window size.
DvrBaseEncryptionInfo Delegate	getEncryptionDelegate ()	Get the encryption delegate for providing streaming side encryption info objects.
String	getEncryptionRepeaterSharedSecret ()	Get the DVR encryption shared secret.

IDvrStreamStore	getHighestVersionedStore() Return stream store that is highest known version.
void	getInitialRepeaterItems (java.util.List items) Get initial repeater items to send to repeater receiver.
int	getMinimumAvailableChunks() Return number of chunks that must be available to stream.
String	getPacketizerName() Get live stream packetizer name.
IDvrStreamStore	getRecordingStreamStore() Get current recording store
void	getRepeaterItemsDvr (java.util.List items, long lastSeq)
String	getStreamBaseName() Get input streamName (no version info).
IDvrStreamStore	getStreamStore (String vStreamName) Given a stream name containing version info, return the associated stream store.
java.util.List	getStreamStores() Get list of all stream stores known to this stream manager.
IDvrStreamVersionHandler	getStreamVersionHandler() Get the stream version handler object.
void	initialAttributes (String vStreamName, int dvrWindow)
void	initialManifest (String vStreamName, IDvrManifest manifest)
void	initialManifestEnd (String vStreamName)
boolean	isRecording() Is this stream currently recording.
boolean	isRecordingPaused() Is this stream currently paused from recording.
IDvrStreamStore	loadArchivedStore (String vStreamName, DvrManifestHolder manifestHolder) Load archived stream store.
void	notifyDvrStreamStorageDeleted (IDvrStreamStore store, IDvrFileSystem fileSystem, boolean success)
void	notifyDvrStreamStoreCreate (IDvrStreamStore store)
void	notifyDvrStreamStoreDestroy (IDvrStreamStore store)
void	notifyDvrStreamStoreInit (IDvrStreamStore store)
void	notifyDvrStreamStoreLoaded (IDvrStreamStore store)

void	notifyTimeReset (IDvrStreamStore store, long oldDvrTime, long oldPacketTime, TimeMapRecord newTimeMapRecord)
IDvrStreamStore	pauseRecording () Request that stream recording pause.
void	purgeManifestEntries (String vStreamName, DvrManifestEntryRangeGroup rangeGroup) Purge entries from store
void	removeDvrRecordingListener (IDvrRecordingListener listener) Remove listener to DVR recording events.
void	removeDvrStoreListener (IDvrStoreListener listener) Remove listener of DVR store lifecycle events.
void	removeStreamStore (String vStreamName)
void	resetStream () Reset the stream.
IDvrStreamStore	resumeRecording () Request that stream recording resume.
void	setDefaultStreamingStore (IDvrStreamStore store) Set the store to be used for streaming.
void	setRecordingStreamStore (IDvrStreamStore store) Set the store used for recording.
void	setStreamVersionHandler (IDvrStreamVersionHandler handler) Set the stream version handler object.
IDvrStreamStore	startRecording () Request that stream recording start.
void	stateChange (String vStreamName, DvrStreamStoreState state)
IDvrStreamStore	stopRecording () Request that stream recording stop.
DvrChunkStorageInfo	storeChunks (int vDuration, DvrPacketHolder vPackets, int aDuration, DvrPacketHolder aPackets, int dDuration, DvrPacketHolder dPackets)
boolean	storeOnMetadata (long pt, long utc, AMFPacket metaPacket)

Methods inherited from interface [com.wowza.wms.stream.livepacketizer.ILiveStreamPacketizer](#)

[getAndSetStartStream](#), [getApplicationInstance](#), [getLiveStreamPacketizerId](#), [getProperties](#), [getRepeaterLastSequence](#), [handlePacket](#), [init](#), [isActive](#), [isPacketizeAudio](#), [isPacketizeData](#), [isPacketizeVideo](#), [isRepeaterEdge](#), [resetStream](#), [setLiveStreamPacketizerId](#), [setPacketizeAudio](#), [setPacketizeData](#), [setPacketizeVideo](#), [setRepeaterEdge](#), [shutdown](#), [startStream](#), [touch](#)

Methods

(continued from last page)

getContextStr

```
public String getContextStr()
```

Get stream context string, useful for logging.

Returns:

stream context string

getStreamBaseName

```
public String getStreamBaseName()
```

Get input streamName (no version info).

Returns:

unversioned incoming stream name

getStreamStore

```
public IDvrStreamStore getStreamStore(String vStreamName)
```

Given a stream name containing version info, return the associated stream store.

Parameters:

vStreamName - versioned stream name

Returns:

store or null

getPacketizerName

```
public String getPacketizerName()
```

Get live stream packetizer name.

Returns:

live stream packetizer name

getStreamStores

```
public java.util.List getStreamStores()
```

Get list of all stream stores known to this stream manager.

Returns:

list of stream stores.

getHighestVersionedStore

```
public IDvrStreamStore getHighestVersionedStore()
```

Return stream store that is highest known version.

Returns:

stream store

getMinimumAvailableChunks

```
public int getMinimumAvailableChunks()
```

Return number of chunks that must be available to stream.

Returns:

minimum available chunks

setRecordingStreamStore

```
public void setRecordingStreamStore(IDvrStreamStore store)
```

Set the store used for recording.

Parameters:

store - stream store, or null

getRecordingStreamStore

```
public IDvrStreamStore getRecordingStreamStore()
```

Get current recording store

Returns:

current recording store (may be null)

setDefaultStreamingStore

```
public void setDefaultStreamingStore(IDvrStreamStore store)
```

Set the store to be used for streaming.

Parameters:

store - store to stream.

getDefaultStreamingStore

```
public IDvrStreamStore getDefaultStreamingStore()
```

Get the store to be used for streaming.

Returns:

store to stream.

getDvrStorageWindowSeconds

```
public int getDvrStorageWindowSeconds()
```

Get DVR window size.

Returns:

window size in seconds (0 means infinite window size)

getDvrStorageDir

```
public String getDvrStorageDir()
```

(continued from last page)

Get the storage directory.

Returns:

storage directory

getDvrFileSystemClass

```
public String getDvrFileSystemClass()
```

Get the class used for the DVR file system.

Returns:

fully qualified class name

getArchiveStrategy

```
public String getArchiveStrategy()
```

Get the DVR archive strategy.

Returns:

The archive strategy

getEncryptionRepeaterSharedSecret

```
public String getEncryptionRepeaterSharedSecret()
```

Get the DVR encryption shared secret.

Returns:

shared secret.

getEncryptionDelegate

```
public DvrBaseEncryptionInfoDelegate getEncryptionDelegate()
```

Get the encryption delegate for providing streaming side encryption info objects.

Returns:

delegate

getStreamVersionHandler

```
public IDvrStreamVersionHandler getStreamVersionHandler()
```

Get the stream version handler object.

Returns:

stream version handler object.

setStreamVersionHandler

```
public void setStreamVersionHandler(IDvrStreamVersionHandler handler)
```

Set the stream version handler object. Must be called after the stream manager is created and before it is initialized.

Parameters:

handler - stream version handler object.

resetStream

```
public void resetStream()
```

Reset the stream.

canRecord

```
public boolean canRecord()
```

Is this stream able to record.

Returns:

true if this stream is recordable.

isRecording

```
public boolean isRecording()
```

Is this stream currently recording. If this stream is not recordable, the method returns false.

Returns:

true if has a recording stream and it is recording.

isRecordingPaused

```
public boolean isRecordingPaused()
```

Is this stream currently paused from recording. If this stream is not recording and not paused, the method returns false.

Returns:

true if has a recording is paused.

startRecording

```
public IDvrStreamStore startRecording()
```

Request that stream recording start.

Used internally. Clients should call [ILiveStreamDvrRecorder.startRecording\(\)](#). Note that this call places the DVR stream store in the recording state. If the stream store is in the paused state, DVR recording will not occur.

If there is not currently a stream store for recording, one will attempt to be created. A successful start will result in registered [IDvrRecordingListeners](#) to have their [IDvrRecordingListener.recordingStarted\(IDvrStreamStore\)](#) method called.

Success only occurs if the stream canRecord [IDvrStreamStore.canRecord\(\)](#) and the stream is not already in the recording state.

Returns:

store if successful. null otherwise.

stopRecording

```
public IDvrStreamStore stopRecording()
```

(continued from last page)

Request that stream recording stop.

Used internally. Clients should call [ILiveStreamDvrRecorder.stopRecording\(\)](#). Note that this call places the DVR stream in the *not* recording state.

A successful stop will result in registered [IDvrRecordingListeners](#) to have their [IDvrRecordingListener.recordingStopped\(IDvrStreamStore\)](#) method called.

Success only occurs if the stream is already in the recording state [IDvrStreamStore.isRecording\(\)](#).

Returns:

store if successful. null otherwise.

pauseRecording

```
public IDvrStreamStore pauseRecording()
```

Request that stream recording pause.

Used internally. Clients should call [ILiveStreamDvrRecorder.pauseRecording\(\)](#). The stream does not have to be actively recording to be paused. For example, it could be paused before packets start flowing.

A successful pause will result in registered [IDvrRecordingListeners](#) to have their [IDvrRecordingListener.recordingPaused\(IDvrStreamStore\)](#) method called.

Returns:

store if successful. null otherwise.

resumeRecording

```
public IDvrStreamStore resumeRecording()
```

Request that stream recording resume.

Used internally. Clients should call [ILiveStreamDvrRecorder.resumeRecording\(\)](#). The stream does not have to be actively recording to be resumed. For example, it could have been paused before the stream started, and this call would move it out of the paused state.

A successful resume will result in registered [IDvrRecordingListeners](#) to have their [IDvrRecordingListener.recordingResumed\(IDvrStreamStore\)](#) method called.

Returns:

store if successful. null otherwise.

loadArchivedStore

```
public IDvrStreamStore loadArchivedStore(String vStreamName,  
DvrManifestHolder manifestHolder)
```

Load archived stream store.

Parameters:

vStreamName - versioned stream Name (e.g. myStream.0)

manifestHolder - manifest holder containing list of manifest properties

Returns:

stream store

deleteArchivedStore

```
public void deleteArchivedStore(String vStreamName)
```

(continued from last page)

Delete archived stream store.

Parameters:

vStreamName - versioned stream Name (e.g. myStream.0)

addManifestEntries

```
public void addManifestEntries(String vStreamName,  
    java.util.List entries,  
    String serverGuid)
```

Add to stream stores manifest.

Parameters:

vStreamName - versioned stream Name (e.g. myStream.0)

entries - list of manifest entries.

serverGuid

purgeManifestEntries

```
public void purgeManifestEntries(String vStreamName,  
    DvrManifestEntryRangeGroup rangeGroup)
```

Purge entries from store

Parameters:

vStreamName - versioned stream Name (e.g. myStream.0)

rangeGroup - set of ranges for purging

storeChunks

```
public DvrChunkStorageInfo storeChunks(int vDuration,  
    DvrPacketHolder vPackets,  
    int aDuration,  
    DvrPacketHolder aPackets,  
    int dDuration,  
    DvrPacketHolder dPackets)
```

storeOnMetadata

```
public boolean storeOnMetadata(long pt,  
    long utc,  
    AMFPacket metaPacket)
```

addDvrStoreListener

```
public void addDvrStoreListener(IDvrStoreListener listener)
```

Add listener to DVR store lifecycle events.

Parameters:

listener - listener

removeDvrStoreListener

```
public void removeDvrStoreListener(IDvrStoreListener listener)
```

(continued from last page)

Remove listener of DVR store lifecycle events.

Parameters:

listener - listener

addDvrRecordingListener

```
public void addDvrRecordingListener(IDvrRecordingListener listener)
```

Add listener to DVR recording events.

Parameters:

listener - listener

removeDvrRecordingListener

```
public void removeDvrRecordingListener(IDvrRecordingListener listener)
```

Remove listener to DVR recording events.

Parameters:

listener - listener

notifyDvrStreamStoreCreate

```
public void notifyDvrStreamStoreCreate(IDvrStreamStore store)
```

notifyDvrStreamStoreInit

```
public void notifyDvrStreamStoreInit(IDvrStreamStore store)
```

notifyDvrStreamStoreDestroy

```
public void notifyDvrStreamStoreDestroy(IDvrStreamStore store)
```

notifyDvrStreamStoreLoaded

```
public void notifyDvrStreamStoreLoaded(IDvrStreamStore store)
```

notifyDvrStreamStorageDeleted

```
public void notifyDvrStreamStorageDeleted(IDvrStreamStore store,  
    IDvrFileSystem fileSystem,  
    boolean success)
```


(continued from last page)

notifyTimeReset

```
public void notifyTimeReset(IDvrStreamStore store,
    long oldDvrTime,
    long oldPacketTime,
    TimeMapRecord newTimeMapRecord)
```

getInitialRepeaterItems

```
public void getInitialRepeaterItems(java.util.List items)
```

Get initial repeater items to send to repeater receiver.

When an edge initially requests the items, instead of sending the entire manifest and all repeater items up until that point, the player sends a `DvrRepeaterHolder.REPEATER_INITIAL_MANIFEST` event followed by several `DvrRepeaterHolder.REPEATER_ADD_MANIFEST_ENTRIES` events.

This allows us to not bother sending any purged manifest entries.

Parameters:

items

getRepeaterItemsDvr

```
public void getRepeaterItemsDvr(java.util.List items,
    long lastSeq)
```

initialManifest

```
public void initialManifest(String vStreamName,
    IDvrManifest manifest)
```

initialManifestEnd

```
public void initialManifestEnd(String vStreamName)
```

initialAttributes

```
public void initialAttributes(String vStreamName,
    int dvrWindow)
```

stateChange

```
public void stateChange(String vStreamName,
    DvrStreamStoreState state)
```

(continued from last page)

removeStreamStore

```
public void removeStreamStore(String vStreamName)
```

addRepeaterHeartBeatItem

```
public void addRepeaterHeartBeatItem()
```

com.wowza.wms.dvr Interface IDvrStreamStore

public interface **IDvrStreamStore**
extends

Method Summary

void	<u>addDvrChunkListener</u> (<u>IDvrStoreChunkListener</u> listener) Add (very fine) listener for chunk events.
void	<u>addManifestEntries</u> (java.util.List entries)
boolean	<u>canPlay</u> () Is this store capable of playing.
boolean	<u>canRecord</u> () Is this store capable of recording.
<u>IApplicationInstance</u>	<u>getAppInstance</u> () Get associated application instance.
<u>IDvrChunker</u>	<u>getChunker</u> ()
long	<u>getClosestStartTime</u> (int type, long t)
String	<u>getContextStr</u> () Get stream context string, useful for logging.
DvrChunk	<u>getDvrChunkAtTime</u> (int fragmentType, long t)
DvrChunk	<u>getDvrChunkByIndex</u> (int fragmentType, int index)
DvrChunk	<u>getDvrChunkNearTime</u> (int fragmentType, long t, long delta)
<u>IDvrStreamManager</u>	<u>getDvrManager</u> () Get DVR Stream Manager
int	<u>getDvrStorageWindowSeconds</u> () Get DVR window size.
IDvrFileSystem	<u>getFileSystem</u> ()
<u>IDvrManifest</u>	<u>getManifest</u> () Get manifest
<u>WMSProperties</u>	<u>getProperties</u> () Get stream store properties.
<u>IDvrPurgeController</u>	<u>getPurgeController</u> () Get purge controller for stream store.

java.util.List	<u>getRecordedEntriesWithLimit</u> (int fragmentType, long t, int limit)
<u>DvrManifestEntry</u>	<u>getRecordedEntryByIndex</u> (int fragmentType, int index)
<u>DvrManifestEntry</u>	<u>getRecordedEntryByTime</u> (int fragmentType, long t)
String	<u>getStreamName</u> () Get versioned stream name.
DvrTimeMapper	<u>getTimeMapper</u> () Get time mapper, which maps between DVR, real and packet time.
boolean	<u>hasAudio</u> () Does this store have audio.
boolean	<u>hasData</u> () Does this store have data.
boolean	<u>hasEncryption</u> () Does this store contain encryption information
boolean	<u>hasOnMetadata</u> () Does this store have on metadata.
boolean	<u>hasVideo</u> () Does this store have video.
void	<u>init</u> () Initialize DVR stream store.
boolean	<u>isLive</u> () Is store currently live For an origin, <u>isRecording()</u> and <u>isLive()</u> will typically return the same result.
boolean	<u>isLoading</u> ()
boolean	<u>isRecording</u> () Is store currently recording.
boolean	<u>isRecordingPaused</u> () Is store currently paused while recording.
boolean	<u>pauseRecording</u> () Request that stream recording pause.
void	<u>purgeEntries</u> (<u>DvrManifestEntryRangeGroup</u> rangeGroup) Purge entries from store
void	<u>removeDvrChunkListener</u> (<u>IDvrStoreChunkListener</u> listener) Remove (very fine) listener for chunk events.
void	<u>reset</u> ()
boolean	<u>resumeRecording</u> () Request that stream recording resume.

void	<code>setCanPlay</code> (boolean canPlay) Set the Stream Stores ability to play.
void	<code>setCanRecord</code> (boolean canRecord) Set the Stream Stores ability to record.
void	<code>setHasEncryption</code> (boolean hasEncryption) Set whether the store has encryption.
void	<code>shutdown</code> ()
boolean	<code>startRecording</code> () Request that recording start.
boolean	<code>stopRecording</code> () Request that stream recording stop.
DvrChunkStorageInfo	<code>storeChunks</code> (int vDuration, DvrPacketHolder vPackets, int aDuration, DvrPacketHolder aPackets, int dDuration, DvrPacketHolder dPackets)
boolean	<code>storeOnMetadata</code> (long pt, long utc, <code>AMFPacket</code> metaPacket)

Methods

init

```
public void init()
```

Initialize DVR stream store.

reset

```
public void reset()
```

shutdown

```
public void shutdown()
```

getProperties

```
public WMSPProperties getProperties()
```

Get stream store properties.

Returns:

properties

getDvrStorageWindowSeconds

```
public int getDvrStorageWindowSeconds()
```

(continued from last page)

Get DVR window size. This may be set by setting the IDvrStreamStore's property [IDvrConstants.PROPERTY_WINDOW_DURATION](#) after the store is created and before it is initialized by using the [IDvrStoreListener.dvrStreamStoreCreate\(IDvrStreamStore\)](#)

Returns:

window size in seconds (0 means infinite window size)

getStreamName

```
public String getStreamName()
```

Get versioned stream name.

Returns:

(versioned) stream name

getContextStr

```
public String getContextStr()
```

Get stream context string, useful for logging.

Returns:

stream context string

getDvrManager

```
public IDvrStreamManager getDvrManager()
```

Get DVR Stream Manager

Returns:

DVR Stream Manager

getAppInstance

```
public IApplicationInstance getAppInstance()
```

Get associated application instance.

Returns:

application instance.

getManifest

```
public IDvrManifest getManifest()
```

Get manifest

Returns:

manifest

getTimeMapper

```
public DvrTimeMapper getTimeMapper()
```

Get time mapper, which maps between DVR, real and packet time.

Returns:

(continued from last page)

time mapper

getPurgeController

```
public IDvrPurgeController getPurgeController()
```

Get purge controller for stream store.

Returns:

purge controller

getFileSystem

```
public IDvrFileSystem getFileSystem()
```

getChunker

```
public IDvrChunker getChunker()
```

canRecord

```
public boolean canRecord()
```

Is this store capable of recording.

Returns:

true if store can record

isLoaded

```
public boolean isLoaded()
```

setCanRecord

```
public void setCanRecord(boolean canRecord)
```

Set the Stream Stores ability to record.

Note: Must be called when stream is not recording.

Parameters:

canRecord

canPlay

```
public boolean canPlay()
```

Is this store capable of playing.

Returns:

true if store can play

setCanPlay

```
public void setCanPlay(boolean canPlay)
```

Set the Stream Stores ability to play.

Note: Must be called when stream is not playing.

Parameters:

canPlay

setHasEncryption

```
public void setHasEncryption(boolean hasEncryption)
```

Set whether the store has encryption. Used internally.

Note: Must be called when stream is not playing.

Parameters:

hasEncryption - if store has encryption

hasEncryption

```
public boolean hasEncryption( )
```

Does this store contain encryption information

Returns:

true if store contains encryption

hasVideo

```
public boolean hasVideo( )
```

Does this store have video.

Returns:

true if store contains video, false otherwise.

hasAudio

```
public boolean hasAudio( )
```

Does this store have audio.

Returns:

true if store contains audio, false otherwise.

hasData

```
public boolean hasData( )
```

Does this store have data.

Returns:

true if store contains data, false otherwise.

(continued from last page)

hasOnMetadata

```
public boolean hasOnMetadata( )
```

Does this store have on metadata.

Returns:

true if store contains on metadata, false otherwise.

startRecording

```
public boolean startRecording( )
```

Request that recording start.

Used internally. Clients should call [ILiveStreamDvrRecorder.startRecording\(\)](#).

Success occurs if the stream canRecord [canRecord\(\)](#) and the stream is not already in the recording state [isRecording\(\)](#).

Returns:

true if successful.

pauseRecording

```
public boolean pauseRecording( )
```

Request that stream recording pause.

Used internally. Clients should call [ILiveStreamDvrRecorder.pauseRecording\(\)](#).

Success occurs if the stream is not already paused.

Returns:

true if successful.

resumeRecording

```
public boolean resumeRecording( )
```

Request that stream recording resume.

Used internally. Clients should call [ILiveStreamDvrRecorder.resumeRecording\(\)](#).

Success occurs if the stream was previously paused.

Returns:

true if successful.

stopRecording

```
public boolean stopRecording( )
```

Request that stream recording stop. This shuts down saving of chunks

Used internally. Clients should call [ILiveStreamDvrRecorder.stopRecording\(\)](#).

Success only occurs if the stream is already in the recording state [isRecording\(\)](#).

Returns:

true if successful.

isRecordingPaused

```
public boolean isRecordingPaused()
```

Is store currently paused while recording.

Returns:

true if recording is paused.

isRecording

```
public boolean isRecording()
```

Is store currently recording. This will always be false for an edge.

Returns:

true if recording

isLive

```
public boolean isLive()
```

Is store currently live For an origin, [isRecording\(\)](#) and [isLive\(\)](#) will typically return the same result. On an origin, [isRecording\(\)](#) will always be false. But [isLive\(\)](#) will reflect the status of the repeated stream.

Returns:

true if live

storeChunks

```
public DvrChunkStorageInfo storeChunks(int vDuration,  
    DvrPacketHolder vPackets,  
    int aDuration,  
    DvrPacketHolder aPackets,  
    int dDuration,  
    DvrPacketHolder dPackets)
```

storeOnMetadata

```
public boolean storeOnMetadata(long pt,  
    long utc,  
    AMFPacket metaPacket)
```

purgeEntries

```
public void purgeEntries(DvrManifestEntryRangeGroup rangeGroup)
```

Purge entries from store

Parameters:

rangeGroup - s set of ranges for purging

(continued from last page)

addManifestEntries

```
public void addManifestEntries(java.util.List entries)
```

addDvrChunkListener

```
public void addDvrChunkListener(IDvrStoreChunkListener listener)
```

Add (very fine) listener for chunk events.

Parameters:

listener - listener

removeDvrChunkListener

```
public void removeDvrChunkListener(IDvrStoreChunkListener listener)
```

Remove (very fine) listener for chunk events.

Parameters:

listener - listener

getRecordedEntriesWithLimit

```
public java.util.List getRecordedEntriesWithLimit(int fragmentType,  
    long t,  
    int limit)
```

getRecordedEntryByIndex

```
public DvrManifestEntry getRecordedEntryByIndex(int fragmentType,  
    int index)
```

getRecordedEntryByTime

```
public DvrManifestEntry getRecordedEntryByTime(int fragmentType,  
    long t)
```

getDvrChunkByIndex

```
public DvrChunk getDvrChunkByIndex(int fragmentType,  
    int index)
```

getDvrChunkAtTime

```
public DvrChunk getDvrChunkAtTime(int fragmentType,  
    long t)
```

getDvrChunkNearTime

```
public DvrChunk getDvrChunkNearTime(int fragmentType,  
    long t,  
    long delta)
```

getClosestStartTime

```
public long getClosestStartTime(int type,  
    long t)
```

com.wowza.wms.dvr Interface IDvrStreamVersionHandler

All Known Implementing Classes:
[DefaultDvrStreamVersionHandler](#)

public interface **IDvrStreamVersionHandler**
 extends

Callback to manage (archival) versions of DVR stream stores

Method Summary

IDvrStreamStore	determineExistingStoreForPlaying (IDvrStreamManager dvrManager, String baseStreamName) When a session requests a stream store w/o specifying the version, we need gto determine the version of the stream to store.
IDvrStreamStore	determineExistingStoreForRecording (IDvrStreamManager dvrManager, String baseStreamName) After a set of versioned streams are loaded from disk, one of them may be designated the stream that will be recorded to (i.e.
String	getArchiveStrategy (IDvrStreamManager dvrManager, String baseStreamName) Determine the archive strategy for a given set of streams.
boolean	handleArchivedStream (IDvrStreamManager dvrManager, String baseStreamName, String version, java.util.SortedSet versions, DvrManifestHolder manifest) When a stream group inits, the file system is checked for older versions of the streams.
boolean	shouldDeleteArchivedStream (IDvrStreamManager dvrManager, IDvrStreamStore store) Determine if a given stream store version should be deleted.
boolean	shouldLoadArchivedStream (IDvrStreamManager dvrManager, String baseStreamName, String version, java.util.SortedSet versions, DvrManifestHolder manifest) Determine if a given stream store version should be loaded into WMS.

Methods

determineExistingStoreForRecording

```
public IDvrStreamStore determineExistingStoreForRecording(IDvrStreamManager
dvrManager,
    String baseStreamName)
```

After a set of versioned streams are loaded from disk, one of them may be designated the stream that will be recorded to (i.e. append mode).

The default implementation says that if append mode, then the highest versioned stream that has canRecord set to true is used. But this API, allows for a more detailed logic.

Parameters:

dvrManager - The DVR Stream Manager

(continued from last page)

baseStreamName - The base stream name (no version info). e.g. myStream

Returns:

stream store to append to, null if no stream in the stream group is to be appended to.

See Also:

[DefaultDvrStreamVersionHandler](#)

determineExistingStoreForPlaying

```
public IDvrStreamStore determineExistingStoreForPlaying(IDvrStreamManager dvrManager,  
String baseStreamName)
```

When a session requests a stream store w/o specifying the version, we need to determine the version of the stream to store.

The default implementation first checks the recording stream. If it 'canPlay()', it is returned. Otherwise, the highest versioned stream that canRecord set to true is used.

Parameters:

dvrManager - The DVR Stream Manager

baseStreamName - The base stream name (no version info). e.g. myStream

Returns:

stream store to stream, null if no stream in the stream group can be played.

See Also:

[DefaultDvrStreamVersionHandler](#)

getArchiveStrategy

```
public String getArchiveStrategy(IDvrStreamManager dvrManager,  
String baseStreamName)
```

Determine the archive strategy for a given set of streams.

Typically, this returns the archive strategy as specified in Application.xml, but this API provides a hook for more detailed logic in determining the strategy.

Parameters:

dvrManager - The DVR App Instance Manager

baseStreamName - The base stream name (no version info). e.g. myStream

Returns:

the archive strategy

See Also:

[DefaultDvrStreamVersionHandler](#)

[IDvrConstants.ARCHIVE_STRATEGY_APPEND](#)

[IDvrConstants.ARCHIVE_STRATEGY_DELETE](#)

[IDvrConstants.ARCHIVE_STRATEGY_VERSION](#)

handleArchivedStream

```
public boolean handleArchivedStream(IDvrStreamManager dvrManager,  
String baseStreamName,  
String version,  
java.util.SortedSet versions,  
DvrManifestHolder manifest)
```

(continued from last page)

When a stream group inits, the file system is checked for older versions of the streams. This method is called to handle the streams.

Typically, 3 things can happen: The stream is ignored, loaded, or deleted.

Parameters:

dvrManager - The DVR Stream Manager
baseStreamName - The base stream name (no version info). e.g. myStream
version - The version of the stream to be loaded.
versions - A sorted set of all the versions that are attempting to be loaded
manifest - The main manifest info of the stream

Returns:

true if handled in some manner, false if ignored (unhandled)

See Also:

[DefaultDvrStreamVersionHandler](#)

shouldLoadArchivedStream

```
public boolean shouldLoadArchivedStream(IDvrStreamManager dvrManager,  
    String baseStreamName,  
    String version,  
    java.util.SortedSet versions,  
    DvrManifestHolder manifest)
```

Determine if a given stream store version should be loaded into WMS.

Parameters:

dvrManager - The DVR Stream Manager
baseStreamName - The base stream name (no version info). e.g. myStream
version - The version of the stream to be loaded.
versions - A sorted set of all the versions that are attempting to be loaded
manifest - The main manifest info of the stream

Returns:

true if should be loaded. false if not.

See Also:

[DefaultDvrStreamVersionHandler](#)

shouldDeleteArchivedStream

```
public boolean shouldDeleteArchivedStream(IDvrStreamManager dvrManager,  
    IDvrStreamStore store)
```

Determine if a given stream store version should be deleted.

Parameters:

dvrManager - The DVR Stream Manager
store - The DVR store

Returns:

true if should be deleted. false if not.

See Also:

[DefaultDvrStreamVersionHandler](#)

com.wowza.wms.dvr

Interface IDvrTextReader

public interface IDvrTextReader
extends ITextReader

Method Summary	
void	init (IApplicationInstance appInstance, IMediaStream stream, IDvrFileSystem fileSystem, String artifactName)

Methods inherited from interface com.wowza.io.ITextReader
close, exists, getBasePath, getMediaName, getPath, init, isOpen, lastModified, length, open, read, ready

Methods

init

```
public void init(IApplicationInstance appInstance,
    IMediaStream stream,
    IDvrFileSystem fileSystem,
    String artifactName)
```


com.wowza.wms.dvr Interface IDvrTextWriter

public interface **IDvrTextWriter**
extends `ITextWriter`

Method Summary

void	<code>init(IApplicationInstance appInstance, IMediaStream stream, IDvrFileSystem fileSystem, String artifactName)</code>
------	--

Methods inherited from interface `com.wowza.io.ITextWriter`

`close`, `exists`, `getBasePath`, `getMediaName`, `getPath`, `init`, `isAppend`, `isOpen`, `lastModified`, `length`, `open`, `setAppend`, `write`, `write`, `write`, `write`

Methods

init

```
public void init(IApplicationInstance appInstance,  
                IMediaStream stream,  
                IDvrFileSystem fileSystem,  
                String artifactName)
```

com.wowza.wms.dvr Interface IDvrTimeMap

All Superinterfaces:

[IDvrChannelManifest](#)

public interface **IDvrTimeMap**

extends [IDvrChannelManifest](#)

Method Summary

boolean	containsTime (long startTime, IDvrConstants.DvrTimeScale timeScale)
long	dvrToPt (long dt)
long	dvrToUtc (long dt)
java.util.List	getTimeMap ()
java.util.List	getTimeMapEntries ()
long	ptToDvr (long pt)
long	toDvr (long t, IDvrConstants.DvrTimeScale timeScale)
long	utcToDvr (long utc)

Methods inherited from interface [com.wowza.wms.dvr.IDvrChannelManifest](#)

[expandEndTime](#), [expandStartTime](#), [getClosestStartTime](#), [getFirstEntry](#), [getFirstIndex](#), [getIndexMap](#), [getLastLiveEntry](#), [getLastRecordedEntry](#), [getLastRecordedIndex](#), [getLiveDuration](#), [getLiveEntries](#), [getLiveEntries](#), [getLiveEntriesWithLimit](#), [getLiveRangeEndingBeforeTime](#), [getLiveRangeEndingBeforeTime](#), [getLiveTailEntries](#), [getNumberLiveEntries](#), [getNumberLiveEntries](#), [getNumberRecordedEntries](#), [getNumberRecordedEntries](#), [getRecordedDuration](#), [getRecordedEntries](#), [getRecordedEntries](#), [getRecordedEntries](#), [getRecordedEntriesInRange](#), [getRecordedEntriesWithLimit](#), [getRecordedEntryByIndex](#), [getRecordedEntryByTimeKey](#), [getRecordedEntryStartingBeforeTime](#), [getType](#), [isEmpty](#)

Methods

getTimeMap

public java.util.List **getTimeMap**()

(continued from last page)

getTimeMapEntries

```
public java.util.List getTimeMapEntries()
```

dvrToUtc

```
public long dvrToUtc(long dt)
```

dvrToPt

```
public long dvrToPt(long dt)
```

ptToDvr

```
public long ptToDvr(long pt)
```

utcToDvr

```
public long utcToDvr(long utc)
```

toDvr

```
public long toDvr(long t,  
    IDvrConstants.DvrTimeScale timeScale)
```

containsTime

```
public boolean containsTime(long startTime,  
    IDvrConstants.DvrTimeScale timeScale)
```

Package

com.wowza.wms.http

com.wowza.wms.http Class HTTPProvider2Base

java.lang.Object

└─com.wowza.wms.http.HTTPProvider2Base

All Implemented Interfaces:

[IHTTPProvider2](#)

public abstract class **HTTPProvider2Base**
extends Object
implements [IHTTPProvider2](#)

HTTPProvider2Base: base class for implementing HTTP Providers.

Simple HTTPProvider class

```
public class HTTPHelloWowza extends HTTPProvider2Base
{
    public void onHTTPRequest(IVHost vhost, IHTTPRequest req, IHTTPResponse resp)
    {
        if (!doHTTPAuthentication(vhost, req, resp))
            return;

        String retStr = "<head><title>Hello Wowza</title></head><body>Hello
Wowza</body>";
        try
        {
            OutputStream out = resp.getOutputStream();
            byte[] outBytes = retStr.getBytes();
            out.write(outBytes);
        }
        catch (Exception e)
        {
            WMSLoggerFactory.getLogger(HTTPHelloWowza.class).error("HTTPHelloWowza ", e);
        }
    }
}
```

Field Summary

protected	authenticateHandler
-----------	-------------------------------------

protected	authenticateHTTPProviderHandler
protected	authenticationMethod
protected	filters
protected	properties
protected	requestFilters

Constructor Summary

public	HTTPProvider2Base()
--------	-------------------------------------

Method Summary

boolean	canHandle (String path) Return true if can handle the request
boolean	doHTTPAuthentication (IVHost vhost, IHTTPRequest req, IHTTPResponse resp) Handle authentication request
String	getAuthenticationMethod () Get the authentication method: digest, basic, none...
String	getPath (IHTTPRequest req, boolean removeFilter) Get the request path
String	getRequestFilters () Get the request filter
void	init () Initialize the HTTPProvider
void	onBind (IVHost vhost, HostPort hostPort) Called when bind is called on port
void	onUnbind (IVHost vhost, HostPort hostPort) Called when unbind is called on port
void	setAuthenticationMethod (String authenticationMethod) Set authentication method: digest, basic, none...
void	setProperties (WMSProperties properties) Set properties
void	setRequestFilters (String requestFilters) Set the request filter

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface [com.wowza.wms.http.IHTTPProvider2](#)

[canHandle](#), [getAuthenticationMethod](#), [getRequestFilters](#), [init](#), [setAuthenticationMethod](#), [setRequestFilters](#)

Methods inherited from interface [com.wowza.wms.http.IHTTPProvider](#)

[onBind](#), [onHTTPRequest](#), [onUnbind](#), [setProperties](#)

Fields

properties

protected com.wowza.wms.application.WMSProperties **properties**

filters

protected java.util.List **filters**

requestFilters

protected java.lang.String **requestFilters**

authenticateHandler

protected com.wowza.wms.authentication.IAuthenticate **authenticateHandler**

authenticateHTTPProviderHandler

protected com.wowza.wms.authentication.IAuthenticateHTTPProvider
authenticateHTTPProviderHandler

authenticationMethod

protected java.lang.String **authenticationMethod**

Constructors

HTTPProvider2Base

public **HTTPProvider2Base**()

(continued from last page)

Methods

init

```
public void init()
```

Initialize the HTTPProvider

canHandle

```
public boolean canHandle(String path)
```

Return true if can handle the request

Parameters:

path - request path

Returns:

true, if can handle the request

getPath

```
public String getPath(IHttpRequest req,  
boolean removeFilter)
```

Get the request path

Parameters:

req - request

removeFilter - set to true to remove filter part of URL

Returns:

path

setRequestFilters

```
public void setRequestFilters(String requestFilters)
```

Set the request filter

Parameters:

requestFilters - request filter

getRequestFilters

```
public String getRequestFilters()
```

Get the request filter

Returns:

request filter

setProperties

```
public void setProperties(WMSProperties properties)
```

Set properties

(continued from last page)

Parameters:

properties - properties

onBind

```
public void onBind(IVHost vhost,  
                  HostPort hostPort)
```

Called when bind is called on port

Parameters:

vhost - vhost

hostPort - host port

onUnbind

```
public void onUnbind(IVHost vhost,  
                    HostPort hostPort)
```

Called when unbind is called on port

Parameters:

vhost - vhost

hostPort - host port

getAuthenticationMethod

```
public String getAuthenticationMethod()
```

Get the authentication method: digest, basic, none...

Returns:

authentication method

setAuthenticationMethod

```
public void setAuthenticationMethod(String authenticationMethod)
```

Set authentication method: digest, basic, none...

Parameters:

authenticationMethod - authentication method

doHTTPAuthentication

```
public boolean doHTTPAuthentication(IVHost vhost,  
                                     IHTTPRequest req,  
                                     IHTTPResponse resp)
```

Handle authentication request

Parameters:

vhost - vhost

req - request

resp - response

Returns:

true, if request should be handled by the HTTPProvider subclass

com.wowza.wms.http Interface IHTTPProvider

All Subinterfaces:

[IHTTPProvider2](#)

public interface **IHTTPProvider**
extends

IHTTPProvider: HTTP provider class for a given HostPort definition. Receive all HTTP requests that are not RTMPT requests. See IHTTPProvider2 for the extended version of this interface.

Method Summary

void	onBind (IVHost vhost, HostPort hostPort) Triggered after hostPort binds to socket
void	onHTTPRequest (IVHost vhost, IHTTPRequest req, IHTTPResponse resp) Triggered for each HTTP request to the given hostPort that is not an RTMPT request.
void	onUnbind (IVHost vhost, HostPort hostPort) Triggered after hostPort unbinds
void	setProperties (WMSProperties properties) Called to provide properties set in configuration files

Methods

onBind

```
public void onBind(IVHost vhost,  
                   HostPort hostPort)
```

Triggered after hostPort binds to socket

Parameters:

vhost - parent vhost
hostPort - host port definition

onHTTPRequest

```
public void onHTTPRequest(IVHost vhost,  
                           IHTTPRequest req,  
                           IHTTPResponse resp)
```

Triggered for each HTTP request to the given hostPort that is not an RTMPT request.

Parameters:

vhost - parent vhost
req - HTML request object
resp - HTML response object

onUnbind

```
public void onUnbind(IVHost vhost,  
    HostPort hostPort)
```

Triggered after hostPort unbinds

Parameters:

vhost - parent vhost

hostPort - host port definition

setProperties

```
public void setProperties(WMSProperties properties)
```

Called to provide properties set in configuration files

com.wowza.wms.http Interface IHTTPProvider2

All Superinterfaces:

[IHTTPProvider](#)

All Known Implementing Classes:

[HTTPProvider2Base](#)

public interface **IHTTPProvider2**
extends [IHTTPProvider](#)

IHTTPProvider2: Extension of IHTTPProvider that adds support for multiple HTTPProviders attached to a single HostPort along with authentication. Multiple HTTPProviders can be added to a HostPort definition. They are configured as follows:

```
<HTTPProvider>
  <BaseClass>com.wowza.wms.http.HTTPServerVersion</BaseClass>
  <RequestFilters>*serverversion</RequestFilters>
  <AuthenticationMethod>none</AuthenticationMethod>
</HTTPProvider>
```

Below is an sample implementation of the HTTPServerVersion provider:

```

import java.io.*;

import com.wowza.wms.server.*;
import com.wowza.wms.stream.*;
import com.wowza.wms.vhost.*;
import com.wowza.wms.logging.*;

public class HTTPServerVersion extends HTTPProvider2Base
{
    public void onHTTPRequest(IVHost vhost, IHTTPRequest req, IHTTPResponse resp)
    {
        if (!doHTTPAuthentication(vhost, req, resp))
            return;

        String version = MediaStreamBase.p+" "+ReleaseInfo.getVersion()+"
build"+ReleaseInfo.getBuildNumber();
        String retStr =
"<html><head><title>"+version+"</title></head><body>"+version+"</body></html>";
        try
        {
            OutputStream out = resp.getOutputStream();
            byte[] outBytes = retStr.getBytes();
            out.write(outBytes);
        }
        catch (Exception e)
        {
            WMSLoggerFactory.getLogger(HTTPServerVersion.class).error("HTMLServerVersion ", e);
        }
    }
}

```

Method Summary

boolean	canHandle (String path)
String	getAuthenticationMethod ()
String	getRequestFilters ()
void	init ()
void	setAuthenticationMethod (String authenticationMethod)
void	setRequestFilters (String requestFilters)

Methods inherited from interface [com.wowza.wms.http.IHTTPProvider](#)

[onBind](#), [onHTTPRequest](#), [onUnbind](#), [setProperties](#)

Methods

canHandle

```
public boolean canHandle(String path)
```

setRequestFilters

```
public void setRequestFilters(String requestFilters)
```

getRequestFilters

```
public String getRequestFilters()
```

init

```
public void init()
```

getAuthenticationMethod

```
public String getAuthenticationMethod()
```

setAuthenticationMethod

```
public void setAuthenticationMethod(String authenticationMethod)
```

com.wowza.wms.http Interface IHTTPRequest

public interface **IHTTPRequest**
extends

Method Summary

int	<u>getContentLength()</u> Get the content length of the body of the message
String	<u>getContentType()</u> Get the request content type
String	<u>getHeader(String name)</u> Get a HTTP header value such as 'Content-Length'
byte[]	<u>getHeaderBytes()</u> Returns the header as bytes
java.util.Map	<u>getHeaderMap()</u> Get a copy of the HTTP request header map
java.util.Set	<u>getHeaderNames()</u> Get a Set of the header names
java.io.InputStream	<u>getInputStream()</u> Get the body of the message as an input stream
int	<u>getIntHeader(String name)</u> Get a HTTP header value such as 'Content-Length' and return as int
java.util.Locale	<u>getLocale()</u> Get locale of request (Example: en-us)
String	<u>getMethod()</u> Get the method invocation method: GET, POST, HEAD
byte[]	<u>getMsgBytes()</u> Return the message bytes
String	<u>getParameter(String name)</u> Get a parameter value
java.util.Map	<u>getParameterMap()</u> Get the entire parameter Map
java.util.Set	<u>getParameterNames()</u> Get a Set of parameter names
String[]	<u>getParameterValues(String name)</u> Get a multi-value parameter as an array of String
String	<u>getPath()</u> Returns the HTTP path element of the request

String	<code>getProtocol()</code> Get the request protocol (example: HTTP/1.1)
String	<code>getQueryString()</code> Get the query string part of the url (everything after the ?)
String	<code>getRemoteAddr()</code> Get the remote ip address of the request
String	<code>getRemoteHost()</code> Get the remote host name (if known) if not return ip address
String	<code>getRequestURI()</code> Get the full request URI
String	<code>getRequestURL()</code> Get the request url (same as URI minus the query string)
String	<code>getScheme()</code> Get the request scheme (Example "http")
String	<code>getServerName()</code> Get the name of the server (Example: "Wowza Streaming Engine")
int	<code>getServerPort()</code> Get the port this request was received on
boolean	<code>isHeadRequest()</code> Returns true if the request is a HEAD request
boolean	<code>isSecure()</code> Returns true is the request is protected by SSL
void	<code>parseBodyForParams()</code> If the body of the message contains parameter data (data in name value pairs separated by & character) call this routine to decode those parameters and add them to the parameter map.
void	<code>parseBodyForParams(boolean doDecode)</code> If the body of the message contains parameter data (data in name value pairs separated by & character) call this routine to decode those parameters and add them to the parameter map.

Methods

getHeaderMap

```
public java.util.Map getHeaderMap()
```

Get a copy of the HTTP request header map

Returns:

copy of the HTTP request header map

getHeader

```
public String getHeader(String name)
```

Get a HTTP header value such as 'Content-Length'

(continued from last page)

Parameters:

name - header name

Returns:

header value

getIntHeader

```
public int getIntHeader(String name)
```

Get a HTTP header value such as 'Content-Length' and return as int

Parameters:

name - header name

Returns:

header value

getHeaderNames

```
public java.util.Set getHeaderNames()
```

Get a Set of the header names

Returns:

Set of header names

getMethod

```
public String getMethod()
```

Get the method invocation method: GET, POST, HEAD

Returns:

method

getQueryString

```
public String getQueryString()
```

Get the query string part of the url (everything after the ?)

Returns:

query string

getContentLength

```
public int getContentLength()
```

Get the content length of the body of the message

Returns:

content length of the body of the message

getRequestURI

```
public String getRequestURI()
```

(continued from last page)

Get the full request URI

Returns:

full request URI

getRequestURL

```
public String getRequestURL()
```

Get the request url (same as URI minus the query string)

Returns:

request url

getContentType

```
public String getContentType()
```

Get the request content type

Returns:

request content type

parseBodyForParams

```
public void parseBodyForParams(boolean doDecode)
```

If the body of the message contains parameter data (data in name value pairs separated by & character) call this routine to decode those parameters and add them to the parameter map.

Parameters:

doDecode - true to decode the params as though they are url params

parseBodyForParams

```
public void parseBodyForParams()
```

If the body of the message contains parameter data (data in name value pairs separated by & character) call this routine to decode those parameters and add them to the parameter map.

getParameter

```
public String getParameter(String name)
```

Get a parameter value

Parameters:

name - parameter name

Returns:

parameter value

getParameterNames

```
public java.util.Set getParameterNames()
```

Get a Set of parameter names

(continued from last page)

Returns:

Set of parameter names

getParameterValues

```
public String[] getParameterValues(String name)
```

Get a multi-value parameter as an array of String

Parameters:

name - parameter name

Returns:

multi-value parameter as an array of String

getParameterMap

```
public java.util.Map getParameterMap()
```

Get the entire parameter Map

Returns:

parameter Map

getInputStream

```
public java.io.InputStream getInputStream()
```

Get the body of the message as an input stream

Returns:

body of the message as an input stream

getProtocol

```
public String getProtocol()
```

Get the request protocol (example: HTTP/1.1)

Returns:

request protocol

getScheme

```
public String getScheme()
```

Get the request scheme (Example "http")

Returns:

request scheme

getServerName

```
public String getServerName()
```

Get the name of the server (Example: "Wowza Streaming Engine")

Returns:

(continued from last page)

name of the server

getServerPort

```
public int getServerPort()
```

Get the port this request was received on

Returns:

the port this request was received on

getRemoteAddr

```
public String getRemoteAddr()
```

Get the remote ip address of the request

Returns:

remote ip address of the request

getRemoteHost

```
public String getRemoteHost()
```

Get the remote host name (if known) if not return ip address

Returns:

remote host name

getLocale

```
public java.util.Locale getLocale()
```

Get locale of request (Example: en-us)

Returns:

locale of request

isSecure

```
public boolean isSecure()
```

Returns true is the request is protected by SSL

Returns:

true is the request is protected by SSL

getPath

```
public String getPath()
```

Returns the HTTP path element of the request

Returns:

HTTP path element of the request

(continued from last page)

getHeaderBytes

```
public byte[] getHeaderBytes()
```

Returns the header as bytes

Returns:

header as bytes

getMsgBytes

```
public byte[] getMsgBytes()
```

Return the message bytes

Returns:

message bytes

isHeadRequest

```
public boolean isHeadRequest()
```

Returns true if the request is a HEAD request

Returns:

true if the request is a HEAD request

com.wowza.wms.http Interface IHTTPResponse

public interface **IHTTPResponse**
extends

Method Summary

boolean	<code>containsHeader</code> (String name) Returns true if response header contains parameter name
String	<code>getHeader</code> (String name) Get header value
java.util.Map	<code>getHeaders</code> () Get the current response headers as a map
int	<code>getHeaderSize</code> () Get the size in bytes of the HTTP header
int	<code>getIntHeader</code> (String name) Get header value as int
java.io.OutputStream	<code>getOutputStream</code> () Get the output stream for the response.
boolean	<code>isHeadResponse</code> () returns true if the response is due to a HEAD request
void	<code>removeHeader</code> (String name) Remove header value
void	<code>setHeader</code> (String name, String value) Set header value
void	<code>setHeadResponse</code> (boolean isHeadResponse) indicates if the response is a response to a HEAD request
void	<code>setIntHeader</code> (String name, int value) Set header value as int
void	<code>setResponseCode</code> (int responseCode) Set the HTTP response code

Methods

`getOutputStream`

```
public java.io.OutputStream getOutputStream()
```

Get the output stream for the response. You can then write directly into the output stream.

(continued from last page)

Returns:output stream for the response

containsHeader

```
public boolean containsHeader(String name)
```

Returns true if response header contains parameter name

Parameters:

name - header parameter name

Returns:true if header contains value

setHeader

```
public void setHeader(String name,  
                      String value)
```

Set header value

Parameters:

name - header parameter name

value - parameter value

removeHeader

```
public void removeHeader(String name)
```

Remove header value

Parameters:name - name

setIntHeader

```
public void setIntHeader(String name,  
                        int value)
```

Set header value as int

Parameters:

name - header parameter name

value - parameter value

getHeaders

```
public java.util.Map getHeaders()
```

Get the current response headers as a map

Returns:current response headers as a map

getHeader

```
public String getHeader(String name)
```

(continued from last page)

Get header value

Parameters:

name - header parameter name

Returns:

parameter value

getIntHeader

```
public int getIntHeader(String name)
```

Get header value as int

Parameters:

name - header parameter name

Returns:

parameter value

setResponseCode

```
public void setResponseCode(int responseCode)
```

Set the HTTP response code

Parameters:

responseCode - HTTP response code

getHeaderSize

```
public int getHeaderSize()
```

Get the size in bytes of the HTTP header

Returns:

header size in bytes

setHeadResponse

```
public void setHeadResponse(boolean isHeadResponse)
```

indicates if the response is a response to a HEAD request

Parameters:

isHeadResponse - indicating if this is a response to a HEAD request

isHeadResponse

```
public boolean isHeadResponse()
```

returns true if the response is due to a HEAD request

Returns:

true if the response is due to a HEAD request

Package

**com.wowza.wms.httpstreamer.cupertinostrea
ming.httpstreamer**

com.wowza.wms.httpstreamer.cupertinostreaming.httpstreamer Class HTTPStreamerSessionCupertino

java.lang.Object

└─ com.wowza.wms.httpstreamer.model.HTTPStreamerSessionBase

└─

com.wowza.wms.httpstreamer.cupertinostreaming.httpstreamer.HTTPStreamerSessionCupertino

All Implemented Interfaces:

[IHTTPSessionNotify](#), [IHTTPStreamerSession](#)

public class **HTTPStreamerSessionCupertino**
 extends HTTPStreamerSessionBase
 implements [IHTTPStreamerSession](#), [IHTTPSessionNotify](#)

Fields inherited from class com.wowza.wms.httpstreamer.model.HTTPStreamerSessionBase

acceptSession, appInstance, connectionHolder, cookieStr, DATEFORMAT, elapsedTime, fastDateFormat, fileInfoMap, firstCheck, httpHeaders, httpStreamerAdapter, ioPerformanceCounter, ipAddress, isActive, isHTTPOrigin, isPlayLogged, isSecure, lastRequest, liveStreamingPacketizer, lock, mediaCasterStreamLock, notifyCreate, playDuration, playSeek, playStart, properties, queryStr, redirectSession, redirectSessionBody, redirectSessionCode, redirectSessionContentType, redirectSessionURL, referrer, serverIp, serverPort, sessionId, sessionProtocol, sessionTimeout, sessionType, stream, streamDomainStrSet, streamExt, streamName, streamNamePartMap, streamPosition, timeoutSession, totalIOPerformance2Last, totalIOPerformanceLast, transcoderVODIndex, transcoderVODIndexDestinationsMap, transcoderVODIndexDestinationsOrder, transcoderVODSession, uri, userAgent, userHTTPHeaders, userQueryStr, vhost, vodTranscodeNGRP

Fields inherited from interface [com.wowza.wms.httpstreamer.model.IHTTPStreamerSession](#)

[SESSIONPROTOCOL_COUNT](#), [SESSIONPROTOCOL_CUPERTINOSTREAMING](#), [SESSIONPROTOCOL_DIRECTSTREAMING](#), [SESSIONPROTOCOL_DVRCHUNKSTREAMING](#), [SESSIONPROTOCOL_MPEGDASHSTREAMING](#), [SESSIONPROTOCOL_SANJOSESTREAMING](#), [SESSIONPROTOCOL_SMOOTHSTREAMING](#), [SESSIONPROTOCOL_UNKNOWN](#), [SESSIONPROTOCOL_WEBMSTREAMING](#), [SESSIONTYPE_LIVE](#), [SESSIONTYPE_LIVEDVR](#), [SESSIONTYPE_UNKNOWN](#), [SESSIONTYPE_VOD](#)

Constructor Summary

public	HTTPStreamerSessionCupertino()
--------	--

Method Summary

void	clearLoggingValues()
boolean	containsIndex (String streamName)
IHTTPStreamerCupertinoIndex	getIndex (IHTTPStreamerApplicationContext appContext, IHTTPStreamerSession httpStreamerSession, String rawStreamName, String streamExt, String streamName, long playStart, long playDuration, long playSeek, TimedTextRequest captionRequest, String vodTranscodeNGRP)

void	indexTranscoderVOD (IHTTPStreamerApplicationContext appContext, String vodTranscodeNGRP)
void	logLiveChunk (LiveStreamPacketizerCupertinoChunk chunk)
void	logVODChunk (LiveStreamPacketizerCupertinoChunk chunk)
void	notifyHTTPSessionCreate (IApplicationInstance appInstance, IHTTPStreamerSession httpStreamerSession)
void	notifyHTTPSessionDestroy (IApplicationInstance appInstance, IHTTPStreamerSession httpStreamerSession)
void	shutdown ()
void	updateLoggingValues ()

Methods inherited from class com.wowza.wms.httpstreamer.model.HTTPStreamerSessionBase

acceptSession, addHTTPSessionToTCPSession, addIOPerformance, addIOPerformance2, addStreamDomainStr, addStreamDomainStrs, addUserHTTPHeaders, checkAndSetPlayLogged, clearLoggingValues, containsStreamDomainStr, containsStreamNameParts, doSessionRedirect, doSessionRedirect, extractHTTPRequestInfo, getAndClearNotifyCreate, getAppInstance, getConnectionHolder, getCookieStr, getDvrSessionInfo, getElapsedTime, getFileInfo, getHTTPDate, getHTTPHeader, getHTTPHeaderMap, getHTTPHeaderNames, getHTTPIntHeader, getHTTPStreamerAdapter, getIOPerformanceCounter, getIpAddress, getLastRequest, getLiveStreamingPacketizer, getLock, getPlayDuration, getPlaySeek, getPlayStart, getProperties, getQueryStr, getRedirectSessionBody, getRedirectSessionCode, getRedirectSessionContentType, getRedirectSessionURL, getReferrer, getServerIp, getServerPort, getSessionId, getSessionProtocol, getSessionTimeout, getSessionType, getStream, getStreamDomainStr, getStreamDomainStrList, getStreamExt, getStreamName, getStreamNameParts, getStreamPosition, getTimeRunning, getTimeRunningSeconds, getTranscoderVODIndex, getTranscoderVODIndex, getTranscoderVODIndexDestinationsMap, getTranscoderVODIndexDestinationsOrder, getTranscoderVODSession, getUri, getUserAgent, getUserHTTPHeaders, getUserQueryStr, getVHost, getVODTranscodeNGRP, indexTranscoderVOD, isAcceptSession, isActive, isFileInfo, isHTTPOrigin, isPlayLogged, isRedirectSession, isSecure, isTimeout, isTimeoutSession, isValidated, isValidStreamDomainStr, lockRepeaterStreams, putFileInfo, putStreamNameParts, redirectSession, redirectSession, rejectSession, removeStreamDomainStr, setAcceptSession, setActive, setAppInstance, setCookieStr, setDvrSessionInfo, setHTTPOrigin, setHTTPStreamerAdapter, setIpAddress, setLiveStreamingPacketizer, setPlayDuration, setPlayLogged, setPlaySeek, setPlayStart, setQueryStr, setRedirectSession, setRedirectSessionBody, setRedirectSessionCode, setRedirectSessionContentType, setRedirectSessionURL, setReferrer, setSecure, setServerIp, setServerPort, setSessionId, setSessionProtocol, setSessionTimeout, setSessionType, setStream, setStreamExt, setStreamName, setStreamPosition, setThreadContext, setTimeoutSession, setTranscoderVODIndex, setTranscoderVODSession, setUri, setUserAgent, setUserHTTPHeader, setUserQueryStr, setValidated, setVHost, setVODTranscodeNGRP, shutdown, shutdownLocked, touch, updateLoggingValues, validStreamDomainToString

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface [com.wowza.wms.httpstreamer.model.IHTTPStreamerSession](#)

[acceptSession](#), [addHTTPSessionToTCPSession](#), [addIOPerformance](#), [addIOPerformance2](#),
[addStreamDomainStr](#), [addStreamDomainStrs](#), [addUserHTTPHeaders](#), [checkAndSetPlayLogged](#),
[clearLoggingValues](#), [containsStreamDomainStr](#), [containsStreamNameParts](#),
[doSessionRedirect](#), [doSessionRedirect](#), [extractHTTPRequestInfo](#),
[getAndClearNotifyCreate](#), [getAppInstance](#), [getConnectionHolder](#), [getCookieStr](#),
[getDvrSessionInfo](#), [getElapsedTime](#), [getFileInfo](#), [getHTTPHeader](#), [getHTTPHeaderMap](#),
[getHTTPHeaderNames](#), [getHTTPIntHeader](#), [getHTTPStreamerAdapter](#),
[getIOPerformanceCounter](#), [getIpAddress](#), [getLastRequest](#), [getLiveStreamingPacketizer](#),
[getLock](#), [getPlayDuration](#), [getPlaySeek](#), [getPlayStart](#), [getProperties](#), [getQueryStr](#),
[getRedirectSessionBody](#), [getRedirectSessionCode](#), [getRedirectSessionContentType](#),
[getRedirectSessionURL](#), [getReferrer](#), [getServerIp](#), [getServerPort](#), [getSessionId](#),
[getSessionProtocol](#), [getSessionTimeout](#), [getSessionType](#), [getStream](#), [getStreamExt](#),
[getStreamName](#), [getStreamNameParts](#), [getStreamPosition](#), [getTimeRunning](#),
[getTimeRunningSeconds](#), [getTranscoderVODIndex](#), [getTranscoderVODIndex](#),
[getTranscoderVODIndexDestinationsMap](#), [getTranscoderVODIndexDestinationsOrder](#),
[getTranscoderVODSession](#), [getUri](#), [getUserAgent](#), [getUserHTTPHeaders](#), [getUserQueryStr](#),
[getVHost](#), [getVODTranscodeNGRP](#), [indexTranscoderVOD](#), [isAcceptSession](#), [isActive](#),
[isFileInfo](#), [isHTTPOrigin](#), [isPlayLogged](#), [isRedirectSession](#), [isSecure](#), [isTimeout](#),
[isTimeoutSession](#), [isValidated](#), [isValidStreamDomainStr](#), [lockRepeaterStreams](#),
[putFileInfo](#), [putStreamNameParts](#), [redirectSession](#), [redirectSession](#), [rejectSession](#),
[removeStreamDomainStr](#), [setAcceptSession](#), [setActive](#), [setAppInstance](#), [setCookieStr](#),
[setDvrSessionInfo](#), [setHTTPOrigin](#), [setHTTPStreamerAdapter](#), [setIpAddress](#),
[setLiveStreamingPacketizer](#), [setPlayDuration](#), [setPlayLogged](#), [setPlaySeek](#),
[setPlayStart](#), [setQueryStr](#), [setRedirectSession](#), [setRedirectSessionBody](#),
[setRedirectSessionCode](#), [setRedirectSessionContentType](#), [setRedirectSessionURL](#),
[setReferrer](#), [setSecure](#), [setServerIp](#), [setServerPort](#), [setSessionId](#), [setSessionProtocol](#),
[setSessionTimeout](#), [setSessionType](#), [setStream](#), [setStreamExt](#), [setStreamName](#),
[setStreamPosition](#), [setTimeoutSession](#), [setTranscoderVODIndex](#), [setTranscoderVODSession](#),
[setUri](#), [setUserAgent](#), [setUserHTTPHeader](#), [setUserQueryStr](#), [setVHost](#),
[setVODTranscodeNGRP](#), [shutdown](#), [touch](#), [updateLoggingValues](#), [validStreamDomainToString](#)

Methods inherited from interface [com.wowza.wms.httpstreamer.model.IHTTPSessionNotify](#)

[notifyHTTPSessionCreate](#), [notifyHTTPSessionDestroy](#)

Constructors

HTTPStreamerSessionCupertino

```
public HTTPStreamerSessionCupertino()
```

Methods

shutdown

```
public void shutdown()
```

(continued from last page)

containsIndex

```
public boolean containsIndex(String streamName)
```

getIndex

```
public IHTTPStreamerCupertinoIndex getIndex(IHTTPStreamerApplicationContext appContext,
IHTTPStreamerSession httpStreamerSession,
String rawStreamName,
String streamExt,
String streamName,
long playStart,
long playDuration,
long playSeek,
TimedTextRequest captionRequest,
String vodTranscodeNGRP)
```

updateLoggingValues

```
public void updateLoggingValues()
```

clearLoggingValues

```
public void clearLoggingValues()
```

logLiveChunk

```
public void logLiveChunk(LiveStreamPacketizerCupertinoChunk chunk)
```

logVODChunk

```
public void logVODChunk(LiveStreamPacketizerCupertinoChunk chunk)
```

notifyHTTPSessionCreate

```
public void notifyHTTPSessionCreate(IApplicationInstance appInstance,
IHTTPStreamerSession httpStreamerSession)
```

notifyHTTPSessionDestroy

```
public void notifyHTTPSessionDestroy(IApplicationInstance appInstance,
IHTTPStreamerSession httpStreamerSession)
```

indexTranscoderVOD

```
public void indexTranscoderVOD(IHTTPStreamerApplicationContext appContext,  
    String vodTranscodeNGRP)
```

com.wowza.wms.httpstreamer.cupertinostreaming.httpstreamer Interface IHTTPStreamerCupertinoVODActionNotify

All Subinterfaces:

[IHTTPStreamerCupertinoVODActionNotify2](#), [IHTTPStreamerCupertinoVODActionNotify3](#)

public interface **IHTTPStreamerCupertinoVODActionNotify**

extends

IHTTPStreamerCupertinoVODActionNotify: listener interface for video on demand iOS streaming. See

HTTPStreamerApplicationContextCupertinoStreamer.addVODActionListener(IHTTPStreamerCupertinoVODActionNotify listener)

Method Summary

void	onCreate (IHTTPStreamerCupertinoIndex fileIndex, IHTTPStreamerApplicationContext appContext, IHTTPStreamerSession httpStreamerSession, String rawStreamName, String streamExt, String streamName) Called when file index created
void	onDestroy (IHTTPStreamerCupertinoIndex fileIndex) Called after file index is destroyed
void	onFillChunkEnd (IHTTPStreamerCupertinoIndex fileIndex, IHTTPStreamerCupertinoIndexItem item, LiveStreamPacketizerCupertinoChunk chunk, boolean audioOnly) Called after each chunk is filled.
void	onFillChunkStart (IHTTPStreamerCupertinoIndex fileIndex, IHTTPStreamerCupertinoIndexItem item, LiveStreamPacketizerCupertinoChunk chunk, boolean audioOnly) Called each time a chunk is filled.
void	onIndex (IHTTPStreamerCupertinoIndex fileIndex, IHTTPStreamerApplicationContext appContext, IHTTPStreamerSession httpStreamerSession, String rawStreamName, String streamExt, String streamName) Called after file is indexed
void	onInit (IHTTPStreamerCupertinoIndex fileIndex, IHTTPStreamerApplicationContext appContext, IHTTPStreamerSession httpStreamerSession, String rawStreamName, String streamExt, String streamName) Called after initialized
void	onOpen (IHTTPStreamerCupertinoIndex fileIndex, IHTTPStreamerApplicationContext appContext, IHTTPStreamerSession httpStreamerSession, String rawStreamName, String streamExt, String streamName) Called after open

Methods

(continued from last page)

onCreate

```
public void onCreate(IHTTPStreamerCupertinoIndex fileIndex,  
    IHTTPStreamerApplicationContext appContext,  
    IHTTPStreamerSession httpStreamerSession,  
    String rawStreamName,  
    String streamExt,  
    String streamName)
```

Called when file index created

Parameters:

fileIndex - file index
appContext - HTTP application context
httpStreamerSession - HTTP streaming session
rawStreamName - stream name
streamExt - stream extension
streamName - adjusted stream name

onInit

```
public void onInit(IHTTPStreamerCupertinoIndex fileIndex,  
    IHTTPStreamerApplicationContext appContext,  
    IHTTPStreamerSession httpStreamerSession,  
    String rawStreamName,  
    String streamExt,  
    String streamName)
```

Called after initialized

Parameters:

fileIndex - file index
appContext - HTTP application context
httpStreamerSession - HTTP streaming session
rawStreamName - stream name
streamExt - stream extension
streamName - adjusted stream name

onOpen

```
public void onOpen(IHTTPStreamerCupertinoIndex fileIndex,  
    IHTTPStreamerApplicationContext appContext,  
    IHTTPStreamerSession httpStreamerSession,  
    String rawStreamName,  
    String streamExt,  
    String streamName)
```

Called after open

Parameters:

fileIndex - file index
appContext - HTTP application context
httpStreamerSession - HTTP streaming session
rawStreamName - stream name
streamExt - stream extension
streamName - adjusted stream name

(continued from last page)

onIndex

```
public void onIndex(IHTTPStreamerCupertinoIndex fileIndex,  
    IHTTPStreamerApplicationContext appContext,  
    IHTTPStreamerSession httpStreamerSession,  
    String rawStreamName,  
    String streamExt,  
    String streamName)
```

Called after file is indexed

Parameters:

fileIndex - file index
appContext - HTTP application context
httpStreamerSession - HTTP streaming session
rawStreamName - stream name
streamExt - stream extension
streamName - adjusted stream name

onFillChunkStart

```
public void onFillChunkStart(IHTTPStreamerCupertinoIndex fileIndex,  
    IHTTPStreamerCupertinoIndexItem item,  
    LiveStreamPacketizerCupertinoChunk chunk,  
    boolean audioOnly)
```

Called each time a chunk is filled. Can be used to add ID3 data to the header of a chunk.

Parameters:

fileIndex - file index
item - index item
chunk - chunk being filled
audioOnly - is audio-only chunk

onFillChunkEnd

```
public void onFillChunkEnd(IHTTPStreamerCupertinoIndex fileIndex,  
    IHTTPStreamerCupertinoIndexItem item,  
    LiveStreamPacketizerCupertinoChunk chunk,  
    boolean audioOnly)
```

Called after each chunk is filled. Can be used to add ID3 data to the end of a chunk.

Parameters:

fileIndex - file index
item - index item
chunk - chunk being filled
audioOnly - is audio-only chunk

onDestroy

```
public void onDestroy(IHTTPStreamerCupertinoIndex fileIndex)
```

Called after file index is destroyed

Parameters:

fileIndex - file index

com.wowza.wms.httpstreamer.cupertinostreaming.httpstreamer Interface IHTTPStreamerCupertinoVODActionNotify2

All Superinterfaces:

[IHTTPStreamerCupertinoVODActionNotify](#)

All Subinterfaces:

[IHTTPStreamerCupertinoVODActionNotify3](#)

public interface **IHTTPStreamerCupertinoVODActionNotify2**

extends [IHTTPStreamerCupertinoVODActionNotify](#)

Method Summary

void	onFillChunkDataPacket (IHTTPStreamerCupertinoIndex fileIndex, IHTTPStreamerCupertinoIndexItem item, LiveStreamPacketizerCupertinoChunk chunk, boolean audioOnly, AMFPacket packet, ID3Frames id3Frames) Called when data packet is encountered.
------	--

Methods inherited from interface

[com.wowza.wms.httpstreamer.cupertinostreaming.httpstreamer.IHTTPStreamerCupertinoVODActionNotify](#)

[onCreate](#), [onDestroy](#), [onFillChunkEnd](#), [onFillChunkStart](#), [onIndex](#), [onInit](#), [onOpen](#)

Methods

onFillChunkDataPacket

```
public void onFillChunkDataPacket(IHTTPStreamerCupertinoIndex fileIndex,
    IHTTPStreamerCupertinoIndexItem item,
    LiveStreamPacketizerCupertinoChunk chunk,
    boolean audioOnly,
    AMFPacket packet,
    ID3Frames id3Frames)
```

Called when data packet is encountered. Used to convert AMF events into ID3 tags for iOS streaming

Parameters:

fileIndex - file index
item - item
chunk - chunk
audioOnly - is audio-only chunk
packet - amf packet
id3Frames - ID3 frames

com.wowza.wms.httpstreamer.cupertinostreaming.httpstreamer Interface IHTTPStreamerCupertinoVODActionNotify3

All Superinterfaces:

[IHTTPStreamerCupertinoVODActionNotify2](#), [IHTTPStreamerCupertinoVODActionNotify](#)

public interface **IHTTPStreamerCupertinoVODActionNotify3**

extends [IHTTPStreamerCupertinoVODActionNotify2](#)

Method Summary

void	onFillChunkDataPacket (IHTTPStreamerCupertinoIndex fileIndex, IHTTPStreamerCupertinoIndexItem item, LiveStreamPacketizerCupertinoChunk chunk, CupertinoStreamingRendition rendition, AMFPacket packet, ID3Frames id3Frames) Called when data packet is encountered.
void	onFillChunkEnd (IHTTPStreamerCupertinoIndex fileIndex, IHTTPStreamerCupertinoIndexItem item, LiveStreamPacketizerCupertinoChunk chunk, CupertinoStreamingRendition rendition) Called after each chunk is filled.
void	onFillChunkStart (IHTTPStreamerCupertinoIndex fileIndex, IHTTPStreamerCupertinoIndexItem item, LiveStreamPacketizerCupertinoChunk chunk, CupertinoStreamingRendition rendition) Called each time a chunk is filled.

Methods inherited from interface

[com.wowza.wms.httpstreamer.cupertinostreaming.httpstreamer.IHTTPStreamerCupertinoVODActionNotify2](#)

[onFillChunkDataPacket](#)

Methods inherited from interface

[com.wowza.wms.httpstreamer.cupertinostreaming.httpstreamer.IHTTPStreamerCupertinoVODActionNotify](#)

[onCreate](#), [onDestroy](#), [onFillChunkEnd](#), [onFillChunkStart](#), [onIndex](#), [onInit](#), [onOpen](#)

Methods

onFillChunkStart

```
public void onFillChunkStart(IHTTPStreamerCupertinoIndex fileIndex,
    IHTTPStreamerCupertinoIndexItem item,
    LiveStreamPacketizerCupertinoChunk chunk,
    CupertinoStreamingRendition rendition)
```

Called each time a chunk is filled. Can be used to add ID3 data to the header of a chunk.

Parameters:

fileIndex - file index
 item - index item

(continued from last page)

chunk - chunk being filled
rendition - rendition (audioOnly, videoOnly...)

onFillChunkEnd

```
public void onFillChunkEnd(IHTTPStreamerCupertinoIndex fileIndex,  
    IHTTPStreamerCupertinoIndexItem item,  
    LiveStreamPacketizerCupertinoChunk chunk,  
    CupertinoStreamingRendition rendition)
```

Called after each chunk is filled. Can be used to add ID3 data to the end of a chunk.

Parameters:

fileIndex - file index
item - index item
chunk - chunk being filled
rendition - rendition (audioOnly, videoOnly...)

onFillChunkDataPacket

```
public void onFillChunkDataPacket(IHTTPStreamerCupertinoIndex fileIndex,  
    IHTTPStreamerCupertinoIndexItem item,  
    LiveStreamPacketizerCupertinoChunk chunk,  
    CupertinoStreamingRendition rendition,  
    AMFPacket packet,  
    ID3Frames id3Frames)
```

Called when data packet is encountered. Used to convert AMF events into ID3 tags for iOS streaming

Parameters:

fileIndex - file index
item - index item
chunk - chunk being filled
rendition - rendition (audioOnly, videoOnly...)
packet - amf packet
id3Frames - ID3 frames

Package

**com.wowza.wms.httpstreamer.directstreamin
g.httpstreamer**

com.wowza.wms.httpstreamer.directstreaming.httpstreamer Class HTTPStreamerSessionDirect

java.lang.Object

└─ com.wowza.wms.httpstreamer.model.HTTPStreamerSessionBase

└─

com.wowza.wms.httpstreamer.directstreaming.httpstreamer.HTTPStreamerSessionDirect

All Implemented Interfaces:

[IHTTPStreamerSession](#)

public class **HTTPStreamerSessionDirect**
extends HTTPStreamerSessionBase

Fields inherited from class com.wowza.wms.httpstreamer.model.HTTPStreamerSessionBase

acceptSession, appInstance, connectionHolder, cookieStr, DATEFORMAT, elapsedTime, fastDateFormat, fileInfoMap, firstCheck, httpHeaders, httpStreamerAdapter, ioPerformanceCounter, ipAddress, isActive, isHTTPOrigin, isPlayLogged, isSecure, lastRequest, liveStreamingPacketizer, lock, mediaCasterStreamLock, notifyCreate, playDuration, playSeek, playStart, properties, queryStr, redirectSession, redirectSessionBody, redirectSessionCode, redirectSessionContentType, redirectSessionURL, referrer, serverIp, serverPort, sessionId, sessionProtocol, sessionTimeout, sessionType, stream, streamDomainStrSet, streamExt, streamName, streamNamePartMap, streamPosition, timeoutSession, totalIOPerformance2Last, totalIOPerformanceLast, transcoderVODIndex, transcoderVODIndexDestinationsMap, transcoderVODIndexDestinationsOrder, transcoderVODSession, uri, userAgent, userHTTPHeaders, userQueryStr, vhost, vodTranscodeNGRP

Fields inherited from interface [com.wowza.wms.httpstreamer.model.IHTTPStreamerSession](#)

[SESSIONPROTOCOL_COUNT](#), [SESSIONPROTOCOL_CUPERTINOSTREAMING](#), [SESSIONPROTOCOL_DIRECTSTREAMING](#), [SESSIONPROTOCOL_DVRCHUNKSTREAMING](#), [SESSIONPROTOCOL_MPEGDASHSTREAMING](#), [SESSIONPROTOCOL_SANJOSESTREAMING](#), [SESSIONPROTOCOL_SMOOTHSTREAMING](#), [SESSIONPROTOCOL_UNKNOWN](#), [SESSIONPROTOCOL_WEBMSTREAMING](#), [SESSIONTYPE_LIVE](#), [SESSIONTYPE_LIVEDVR](#), [SESSIONTYPE_UNKNOWN](#), [SESSIONTYPE_VOD](#)

Constructor Summary

public [HTTPStreamerSessionDirect](#)()

Methods inherited from class com.wowza.wms.httpstreamer.model.HTTPStreamerSessionBase

```

acceptSession, addHTTPSessionToTCPSession, addIOPerformance, addIOPerformance2,
addStreamDomainStr, addStreamDomainStrs, addUserHTTPHeaders, checkAndSetPlayLogged,
clearLoggingValues, containsStreamDomainStr, containsStreamNameParts,
doSessionRedirect, doSessionRedirect, extractHTTPRequestInfo,
getAndClearNotifyCreate, getAppInstance, getConnectionHolder, getCookieStr,
getDvrSessionInfo, getElapsedTime, getFileInfo, getHTTPDate, getHTTPHeader,
getHTTPHeaderMap, getHTTPHeaderNames, getHTTPIntHeader, getHTTPStreamerAdapter,
getIOPerformanceCounter, getIpAddress, getLastRequest, getLiveStreamingPacketizer,
getLock, getPlayDuration, getPlaySeek, getPlayStart, getProperties, getQueryStr,
getRedirectSessionBody, getRedirectSessionCode, getRedirectSessionContentType,
getRedirectSessionURL, getReferrer, getServerIp, getServerPort, getSessionId,
getSessionProtocol, getSessionTimeout, getSessionType, getStream, getStreamDomainStr,
getStreamDomainStrList, getStreamExt, getStreamName, getStreamNameParts,
getStreamPosition, getTimeRunning, getTimeRunningSeconds, getTranscoderVODIndex,
getTranscoderVODIndex, getTranscoderVODIndexDestinationsMap,
getTranscoderVODIndexDestinationsOrder, getTranscoderVODSession, getUri,
getUserAgent, getUserHTTPHeaders, getUserQueryStr, getVHost, getVODTranscodeNGRP,
indexTranscoderVOD, isAcceptSession, isActive, isFileInfo, isHTTPOrigin,
isPlayLogged, isRedirectSession, isSecure, isTimeout, isTimeoutSession, isValidated,
isValidStreamDomainStr, lockRepeaterStreams, putFileInfo, putStreamNameParts,
redirectSession, redirectSession, rejectSession, removeStreamDomainStr,
setAcceptSession, setActive, setAppInstance, setCookieStr, setDvrSessionInfo,
setHTTPOrigin, setHTTPStreamerAdapter, setIpAddress, setLiveStreamingPacketizer,
setPlayDuration, setPlayLogged, setPlaySeek, setPlayStart, setQueryStr,
setRedirectSession, setRedirectSessionBody, setRedirectSessionCode,
setRedirectSessionContentType, setRedirectSessionURL, setReferrer, setSecure,
setServerIp, setServerPort, setSessionId, setSessionProtocol, setSessionTimeout,
setSessionType, setStream, setStreamExt, setStreamName, setStreamPosition,
setThreadContext, setTimeoutSession, setTranscoderVODIndex, setTranscoderVODSession,
setUri, setUserAgent, setUserHTTPHeader, setUserQueryStr, setValidated, setVHost,
setVODTranscodeNGRP, shutdown, shutdownLocked, touch, updateLoggingValues,
validStreamDomainToString

```

Methods inherited from class java.lang.Object

```

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait,
wait

```

Methods inherited from interface [com.wowza.wms.httpstreamer.model.IHTTPStreamerSession](#)

[acceptSession](#), [addHTTPSessionToTCPSession](#), [addIOPerformance](#), [addIOPerformance2](#),
[addStreamDomainStr](#), [addStreamDomainStrs](#), [addUserHTTPHeaders](#), [checkAndSetPlayLogged](#),
[clearLoggingValues](#), [containsStreamDomainStr](#), [containsStreamNameParts](#),
[doSessionRedirect](#), [doSessionRedirect](#), [extractHTTPRequestInfo](#),
[getAndClearNotifyCreate](#), [getAppInstance](#), [getConnectionHolder](#), [getCookieStr](#),
[getDvrSessionInfo](#), [getElapsedTime](#), [getFileInfo](#), [getHTTPHeader](#), [getHTTPHeaderMap](#),
[getHTTPHeaderNames](#), [getHTTPIntHeader](#), [getHTTPStreamerAdapter](#),
[getIOPerformanceCounter](#), [getIpAddress](#), [getLastRequest](#), [getLiveStreamingPacketizer](#),
[getLock](#), [getPlayDuration](#), [getPlaySeek](#), [getPlayStart](#), [getProperties](#), [getQueryStr](#),
[getRedirectSessionBody](#), [getRedirectSessionCode](#), [getRedirectSessionContentType](#),
[getRedirectSessionURL](#), [getReferrer](#), [getServerIp](#), [getServerPort](#), [getSessionId](#),
[getSessionProtocol](#), [getSessionTimeout](#), [getSessionType](#), [getStream](#), [getStreamExt](#),
[getStreamName](#), [getStreamNameParts](#), [getStreamPosition](#), [getTimeRunning](#),
[getTimeRunningSeconds](#), [getTranscoderVODIndex](#), [getTranscoderVODIndex](#),
[getTranscoderVODIndexDestinationsMap](#), [getTranscoderVODIndexDestinationsOrder](#),
[getTranscoderVODSession](#), [getUri](#), [getUserAgent](#), [getUserHTTPHeaders](#), [getUserQueryStr](#),
[getVHost](#), [getVODTranscodeNGRP](#), [indexTranscoderVOD](#), [isAcceptSession](#), [isActive](#),
[isFileInfo](#), [isHTTPOrigin](#), [isPlayLogged](#), [isRedirectSession](#), [isSecure](#), [isTimeout](#),
[isTimeoutSession](#), [isValidated](#), [isValidStreamDomainStr](#), [lockRepeaterStreams](#),
[putFileInfo](#), [putStreamNameParts](#), [redirectSession](#), [redirectSession](#), [rejectSession](#),
[removeStreamDomainStr](#), [setAcceptSession](#), [setActive](#), [setAppInstance](#), [setCookieStr](#),
[setDvrSessionInfo](#), [setHTTPOrigin](#), [setHTTPStreamerAdapter](#), [setIpAddress](#),
[setLiveStreamingPacketizer](#), [setPlayDuration](#), [setPlayLogged](#), [setPlaySeek](#),
[setPlayStart](#), [setQueryStr](#), [setRedirectSession](#), [setRedirectSessionBody](#),
[setRedirectSessionCode](#), [setRedirectSessionContentType](#), [setRedirectSessionURL](#),
[setReferrer](#), [setSecure](#), [setServerIp](#), [setServerPort](#), [setSessionId](#), [setSessionProtocol](#),
[setSessionTimeout](#), [setSessionType](#), [setStream](#), [setStreamExt](#), [setStreamName](#),
[setStreamPosition](#), [setTimeoutSession](#), [setTranscoderVODIndex](#), [setTranscoderVODSession](#),
[setUri](#), [setUserAgent](#), [setUserHTTPHeader](#), [setUserQueryStr](#), [setVHost](#),
[setVODTranscodeNGRP](#), [shutdown](#), [touch](#), [updateLoggingValues](#), [validStreamDomainToString](#)

Constructors

HTTPStreamerSessionDirect

```
public HTTPStreamerSessionDirect()
```

Package

com.wowza.wms.httpstreamer.model

com.wowza.wms.httpstreamer.model Interface IHTTPCryptoHolder

public interface IHTTPCryptoHolder
extends

Method Summary

byte[]	getBuffer()
int	getOffset()
int	getPacketType()
long	getSample()
int	getSize()
void	setBuffer (byte[] buffer)
void	setOffset (int offset)
void	setPacketType (int packetType)
void	setSample (long sample)
void	setSize (int size)

Methods

getBuffer

public byte[] **getBuffer**()

setBuffer

public void **setBuffer**(byte[] buffer)

getOffset

public int **getOffset**()

setOffset

```
public void setOffset(int offset)
```

getSize

```
public int getSize()
```

setSize

```
public void setSize(int size)
```

getPacketType

```
public int getPacketType()
```

setPacketType

```
public void setPacketType(int packetType)
```

getSample

```
public long getSample()
```

setSample

```
public void setSample(long sample)
```

com.wowza.wms.httpstreamer.model Interface IHTTPSessionFactory

public interface **IHTTPSessionFactory**
extends

Method Summary

IHTTPStreamerSession	createHTTPSessionInstance()
--------------------------------------	---

Methods

createHTTPSessionInstance

public [IHTTPStreamerSession](#) **createHTTPSessionInstance()**

com.wowza.wms.httpstreamer.model Interface IHTTPSessionNotify

All Known Implementing Classes:

[HTTPStreamerSessionSmoothStreamer](#), [HTTPStreamerSessionSanJose](#), [HTTPStreamerSessionMPEGDash](#),
[HTTPStreamerSessionCupertino](#)

public interface **IHTTPSessionNotify**
extends

Method Summary

void	notifyHTTPSessionCreate(IApplicationInstance appInstance, IHTTPStreamerSession httpStreamerSession)
void	notifyHTTPSessionDestroy(IApplicationInstance appInstance, IHTTPStreamerSession httpStreamerSession)

Methods

notifyHTTPSessionCreate

public void **notifyHTTPSessionCreate**([IApplicationInstance](#) appInstance,
[IHTTPStreamerSession](#) httpStreamerSession)

notifyHTTPSessionDestroy

public void **notifyHTTPSessionDestroy**([IApplicationInstance](#) appInstance,
[IHTTPStreamerSession](#) httpStreamerSession)

com.wowza.wms.httpstreamer.model Interface IHTTPStreamerAdapter

public interface **IHTTPStreamerAdapter**
extends

IHTTPStreamerAdapter: HTTP streaming adapter interface

Method Summary

boolean	<u>canHandle</u> (String path) Return true if can handle request
String	<u>getAdapterName</u> () Get the name of this adapter
HTTPStreamerItem	<u>getHTTPStreamerItem</u> () Get the HTTP streamer item associated with this adapter
String	<u>getID</u> () Get the id of this adapter
int	<u>getIdleFrequency</u> () Get the idle frequency (milliseconds) for HTTP requests.
<u>WMSProperties</u>	<u>getProperties</u> () Get properties
int	<u>getSessionProtocol</u> () Set the adapter session protocol.
<u>IVHost</u>	<u>getVHost</u> () Get the vhost associated with this adapter
void	<u>init</u> () Initialize the HTTP streaming adapter
void	<u>service</u> (org.apache.mina.common.io.Session session, RtmpRequestMessage req, RtmpResponseMessage resp) Called to service each request
void	<u>setHTTPStreamerItem</u> (HTTPStreamerItem httpStreamerItem) Set the HTTP streamer item associated with this adapter
void	<u>setID</u> (String id) Set the id of this adapter
void	<u>setProperties</u> (<u>WMSProperties</u> properties) Set properties
void	<u>setSessionProtocol</u> (int sessionProtocol) Get the adapter session protocol.
void	<u>setVHost</u> (<u>IVHost</u> vhost) Set the vhost associated with this adapter

void

[shutdownSession](#)([IApplicationInstance](#) appInstance, [IHTTPStreamerSession](#) session)

Called when an HTTP streaming session is shutdown

Methods

canHandle

```
public boolean canHandle(String path)
```

Return true if can handle request

Parameters:

path - request path

Returns:

true if can handle request

service

```
public void service(org.apache.mina.common.Session session,  
    RtmpRequestMessage req,  
    RtmpResponseMessage resp)
```

Called to service each request

Parameters:

session - io session

req - request

resp - response

getProperties

```
public WMSPProperties getProperties()
```

Get properties

Returns:

properties

setProperties

```
public void setProperties(WMSPProperties properties)
```

Set properties

Parameters:

properties - properties

getHTTPStreamerItem

```
public HTTPStreamerItem getHTTPStreamerItem()
```

Get the HTTP streamer item associated with this adapter

Returns:

(continued from last page)

HTTP streamer item

setHTTPStreamerItem

```
public void setHTTPStreamerItem(HTTPStreamerItem httpStreamerItem)
```

Set the HTTP streamer item associated with this adapter

Parameters:

httpStreamerItem - HTTP streamer item

getVHost

```
public IVHost getVHost()
```

Get the vhost associated with this adapter

Returns:

vhost

setVHost

```
public void setVHost(IVHost vhost)
```

Set the vhost associated with this adapter

Parameters:

vhost - vhost

init

```
public void init()
```

Initialize the HTTP streaming adapter

shutdownSession

```
public void shutdownSession(IApplicationInstance appInstance,  
    IHTTPStreamerSession session)
```

Called when an HTTP streaming session is shutdown

Parameters:

session - HTTP streaming session

getIdleFrequency

```
public int getIdleFrequency()
```

Get the idle frequency (milliseconds) for HTTP requests. This is how often the session is called back while active.

Returns:

idle frequency (milliseconds)

getAdapterName

```
public String getAdapterName()
```

(continued from last page)

Get the name of this adapter

Returns:

name of this adapter

getID

```
public String getID()
```

Get the id of this adapter

Returns:

id of this adapter

setID

```
public void setID(String id)
```

Set the id of this adapter

Parameters:

id - id of this adapter

getSessionProtocol

```
public int getSessionProtocol()
```

Set the adapter session protocol. See IHTTPStreamerSession.SESSIONPROTOCOL_*

Returns:

adapter session protocol

setSessionProtocol

```
public void setSessionProtocol(int sessionProtocol)
```

Get the adapter session protocol. See IHTTPStreamerSession.SESSIONPROTOCOL_*

Parameters:

sessionProtocol - adapter session protocol

com.wowza.wms.httpstreamer.model Interface IHTTPStreamerApplicationContext

public interface **IHTTPStreamerApplicationContext**
extends

IHTTPStreamerApplicationContext: HTTP streamer application context interface. Used by HTTP streaming adapter to store per-application information.

Field Summary

public static final	<u>HTTPORIGINMODE_AUTO</u> Value: 1
public static final	<u>HTTPORIGINMODE_OFF</u> Value: 0
public static final	<u>HTTPORIGINMODE_ON</u> Value: 2

Method Summary

void	<u>addQueryStrRemoveList</u> (String name) Add a name to the list of query string parameters that will not be copied to the media URLs for HTTP streaming
java.util.Map	<u>filterQueryStr</u> (java.util.Map map) Internal
<u>IApplicationInstance</u>	<u>getAppInstance</u> () Get application instance
String	<u>getCacheControlStatus4xx</u> () Get the cache-control header for 4xx HTTP responses
int	<u>getHTTPOriginMode</u> () Get current HTTP origin mode.
int	<u>getMaxHTTPSessionsPerTCPSession</u> () Get maximum number of HTTP sessions tracked per-TCP session
<u>WMSProperties</u>	<u>getProperties</u> () Get properties
<u>MediaCasterItem</u>	<u>getRepeaterMediaCasterDef</u> () Get the live repeater media caster definition
int	<u>getResponseOnIncompleteChunk</u> () Get the response code used when an incomplete chunk (read error) is encountered.
String	<u>getStreamTypeStr</u> () Get stream type

java.util.Map	<u>getUserHTTPHeaders()</u> Get user HTTP headers.
<u>IVHost</u>	<u>getVHost()</u> Get vhost
void	<u>init(IApplicationInstance</u> appInstance, HTTPStreamerItem httpStreamerItem) Initialize context
boolean	<u>isHTTPOriginOn()</u> Returns true if HTTP origin mode is set to ON.
boolean	<u>isStreamDomainProtectionActive()</u> Return true if stream domain protection is on
void	<u>setCacheControlStatus4xx</u> (String cacheControlStatus4xx) Set the cache-control header for 4xx HTTP responses
void	<u>setHTTPOriginMode</u> (int httpOriginMode) Set current HTTP origin mode.
void	<u>setMaxHTTPSessionsPerTCPSession</u> (int maxHTTPSessionsPerTCPSession) Set maximum number of HTTP sessions tracked per-TCP session
void	<u>setRepeaterMediaCasterDef</u> (<u>MediaCasterItem</u> repeaterMediaCasterDef) Set the live repeater media caster definition
void	<u>setResponseOnIncompleteChunk</u> (int responseOnIncompleteChunk) Set the response code used when an incomplete chunk (read error) is encountered.
void	<u>setStreamDomainProtectionActive</u> (boolean streamDomainProtectionActive) Turn on/off stream domain protection
void	<u>setStreamTypeStr</u> (String streamTypeStr) Set stream type
void	<u>setUserHTTPHeader</u> (String name, String value) Add a user HTTP header to the header map.

Fields

HTTPORIGINMODE_OFF

```
public static final int HTTPORIGINMODE_OFF
```

Constant value: **0**

HTTPORIGINMODE_AUTO

```
public static final int HTTPORIGINMODE_AUTO
```

Constant value: **1**

(continued from last page)

HTTPORIGINMODE_ON

```
public static final int HTTPORIGINMODE_ON
```

Constant value: **2**

Methods

init

```
public void init(IApplicationInstance appInstance,  
                HTTPStreamerItem httpStreamerItem)
```

Initialize context

Parameters:

[appInstance](#) - application instance

[httpStreamerItem](#) - HTTP streamer

getProperties

```
public WMSProperties getProperties()
```

Get properties

Returns:

properties

getAppInstance

```
public IApplicationInstance getAppInstance()
```

Get application instance

Returns:

application instance

getVHost

```
public IVHost getVHost()
```

Get vhost

Returns:

vhost

getRepeaterMediaCasterDef

```
public MediaCasterItem getRepeaterMediaCasterDef()
```

Get the live repeater media caster definition

Returns:

live repeater media caster definition

(continued from last page)

setRepeaterMediaCasterDef

```
public void setRepeaterMediaCasterDef(MediaCasterItem repeaterMediaCasterDef)
```

Set the live repeater media caster definition

Parameters:

repeaterMediaCasterDef - live repeater media caster definition

getStreamTypeStr

```
public String getStreamTypeStr()
```

Get stream type

Returns:

stream type

setStreamTypeStr

```
public void setStreamTypeStr(String streamTypeStr)
```

Set stream type

Parameters:

streamTypeStr - stream type

isStreamDomainProtectionActive

```
public boolean isStreamDomainProtectionActive()
```

Return true if stream domain protection is on

Returns:

true if stream domain protection is on

setStreamDomainProtectionActive

```
public void setStreamDomainProtectionActive(boolean streamDomainProtectionActive)
```

Turn on/off stream domain protection

Parameters:

streamDomainProtectionActive - true if stream domain protection is on

getHTTPOriginMode

```
public int getHTTPOriginMode()
```

Get current HTTP origin mode. See HTTPORIGINMODE_ (auto is currently not supported)

Returns:

HTTP origin mode

setHTTPOriginMode

```
public void setHTTPOriginMode(int httpOriginMode)
```

(continued from last page)

Set current HTTP origin mode. See HTTPORIGINMODE_ (auto is currently not supported)

Parameters:

httpOriginMode - HTTP origin mode

isHTTPOriginOn

```
public boolean isHTTPOriginOn()
```

Returns true if HTTP origin mode is set to ON.

Returns:

true if HTTP origin mode is set to ON.

setUserHTTPHeader

```
public void setUserHTTPHeader(String name,  
    String value)
```

Add a user HTTP header to the header map. These headers will be added to every HTTP response.

Parameters:

name - name

value - value (null if no value)

getUserHTTPHeaders

```
public java.util.Map getUserHTTPHeaders()
```

Get user HTTP headers.

Returns:

header map

addQueryStringRemoveList

```
public void addQueryStringRemoveList(String name)
```

Add a name to the list of query string parameters that will not be copied to the media URLs for HTTP streaming

Parameters:

name - query name

filterQueryString

```
public java.util.Map filterQueryString(java.util.Map map)
```

Internal

Parameters:

map - query map

Returns:

query map

getCacheControlStatus4xx

```
public String getCacheControlStatus4xx()
```

(continued from last page)

Get the cache-control header for 4xx HTTP responses

Returns:

cache-control header for 4xx HTTP responses

setCacheControlStatus4xx

```
public void setCacheControlStatus4xx(String cacheControlStatus4xx)
```

Set the cache-control header for 4xx HTTP responses

Parameters:

cacheControlStatus4xx - cache-control header for 4xx HTTP responses

getResponseOnIncompleteChunk

```
public int getResponseOnIncompleteChunk()
```

Get the response code used when an incomplete chunk (read error) is encountered. If zero, chunk is sent.

Returns:

response code

setResponseOnIncompleteChunk

```
public void setResponseOnIncompleteChunk(int responseOnIncompleteChunk)
```

Set the response code used when an incomplete chunk (read error) is encountered. If zero, chunk is sent.

Parameters:

responseOnIncompleteChunk - response code

getMaxHTTPSessionsPerTCPSession

```
public int getMaxHTTPSessionsPerTCPSession()
```

Get maximum number of HTTP sessions tracked per-TCP session

Returns:

maximum number of HTTP sessions tracked per-TCP session

setMaxHTTPSessionsPerTCPSession

```
public void setMaxHTTPSessionsPerTCPSession(int maxHTTPSessionsPerTCPSession)
```

Set maximum number of HTTP sessions tracked per-TCP session

Parameters:

maxHTTPSessionsPerTCPSession - maximum number of HTTP sessions tracked per-TCP session

com.wowza.wms.httpstreamer.model

Interface IHTTPStreamerRepeater

public interface IHTTPStreamerRepeater
extends

IHTTPStreamerRepeater: stream switching constants

Field Summary	
public static final	MSG_STARTSTREAM Value: startStream
public static final	MSG_SWITCHSTREAM Value: switchStream

Fields

MSG_STARTSTREAM

public static final java.lang.String MSG_STARTSTREAM

Constant value: **startStream**

MSG_SWITCHSTREAM

public static final java.lang.String MSG_SWITCHSTREAM

Constant value: **switchStream**

com.wowza.wms.httpstreamer.model Interface IHTTPStreamerRequestContext

public interface **IHTTPStreamerRequestContext**
extends

Field Summary

public static final	<u>CUPERTINO_CHUNKLIST</u> Value: 2
public static final	<u>CUPERTINO_MANIFEST</u> Value: 1
public static final	<u>CUPERTINO_MEDIA</u> Value: 3
public static final	<u>CUPERTINO_SUBTITLE_CHUNK</u> Value: 5
public static final	<u>CUPERTINO_SUBTITLE_LIST</u> Value: 4
public static final	<u>MPEGDASH_MANIFEST</u> Value: 30
public static final	<u>MPEGDASH_MEDIA</u> Value: 31
public static final	<u>NONE</u> Value: 0
public static final	<u>SANJOSE_MANIFEST</u> Value: 10
public static final	<u>SANJOSE_MEDIA</u> Value: 11
public static final	<u>SMOOTHSTREAMING_AUDIO_FRAGMENT</u> Value: 21
public static final	<u>SMOOTHSTREAMING_DATA_FRAGMENT</u> Value: 23
public static final	<u>SMOOTHSTREAMING_MANIFEST</u> Value: 20

public static final	SMOOTHSTREAMING_VIDEO_FRAGMENT Value: 22
---------------------	--

Method Summary

IHTTPRequest	getRequest()
int	getRequestType()
void	setRequest(IHTTPRequest request)
void	setRequestType(int requestType)

Fields

NONE

public static final int **NONE**

Constant value: **0**

CUPERTINO_MANIFEST

public static final int **CUPERTINO_MANIFEST**

Constant value: **1**

CUPERTINO_CHUNKLIST

public static final int **CUPERTINO_CHUNKLIST**

Constant value: **2**

CUPERTINO_MEDIA

public static final int **CUPERTINO_MEDIA**

Constant value: **3**

CUPERTINO_SUBTITLE_LIST

public static final int **CUPERTINO_SUBTITLE_LIST**

Constant value: **4**

CUPERTINO_SUBTITLE_CHUNK

public static final int **CUPERTINO_SUBTITLE_CHUNK**

(continued from last page)

Constant value: **5**

SANJOSE_MANIFEST

```
public static final int SANJOSE_MANIFEST
```

Constant value: **10**

SANJOSE_MEDIA

```
public static final int SANJOSE_MEDIA
```

Constant value: **11**

SMOOTHSTREAMING_MANIFEST

```
public static final int SMOOTHSTREAMING_MANIFEST
```

Constant value: **20**

SMOOTHSTREAMING_AUDIO_FRAGMENT

```
public static final int SMOOTHSTREAMING_AUDIO_FRAGMENT
```

Constant value: **21**

SMOOTHSTREAMING_VIDEO_FRAGMENT

```
public static final int SMOOTHSTREAMING_VIDEO_FRAGMENT
```

Constant value: **22**

SMOOTHSTREAMING_DATA_FRAGMENT

```
public static final int SMOOTHSTREAMING_DATA_FRAGMENT
```

Constant value: **23**

MPEGDASH_MANIFEST

```
public static final int MPEGDASH_MANIFEST
```

Constant value: **30**

MPEGDASH_MEDIA

```
public static final int MPEGDASH_MEDIA
```

Constant value: **31**

(continued from last page)

Methods

getRequestType

```
public int getRequestType()
```

setRequestType

```
public void setRequestType(int requestType)
```

getRequest

```
public IHttpRequest getRequest()
```

setRequest

```
public void setRequest(IHttpRequest request)
```

com.wowza.wms.httpstreamer.model Interface IHTTPStreamerRequestNotify

public interface IHTTPStreamerRequestNotify
extends

IHTTPStreamerRequestNotify: HTTP streamer request interface

Method Summary

void	onHTTPStreamerRequest (IHTTPStreamerSession httpStreamerSession, IHTTPStreamerRequestContext requestContext) Called during an HTTP streaming request after the session has been created
------	--

Methods

onHTTPStreamerRequest

public void **onHTTPStreamerRequest**([IHTTPStreamerSession](#) httpStreamerSession,
[IHTTPStreamerRequestContext](#) requestContext)

Called during an HTTP streaming request after the session has been created

Parameters:

httpStreamerSession - HTTP streaming session

com.wowza.wms.httpstreamer.model Interface IHTTPStreamerSession

public interface **IHTTPStreamerSession**
extends

IHTTPStreamerSession: HTTP streaming session interface

Field Summary

public static final	SESSIONPROTOCOL_COUNT Value: 7
public static final	SESSIONPROTOCOL_CUPERTINOSTREAMING Value: 1
public static final	SESSIONPROTOCOL_DIRECTSTREAMING Value: 6
public static final	SESSIONPROTOCOL_DVRCHUNKSTREAMING Value: 5
public static final	SESSIONPROTOCOL_MPEGDASHSTREAMING Value: 4
public static final	SESSIONPROTOCOL_SANJOSESTREAMING Value: 2
public static final	SESSIONPROTOCOL_SMOOTHSTREAMING Value: 0
public static final	SESSIONPROTOCOL_UNKNOWN Value: -1
public static final	SESSIONPROTOCOL_WEBMSTREAMING Value: 3
public static final	SESSIONTYPE_LIVE Value: 1
public static final	SESSIONTYPE_LIVEDVR Value: 3
public static final	SESSIONTYPE_UNKNOWN Value: 0

public static final	<u>SESSIONTYPE_VOD</u> Value: 2
---------------------	---

Method Summary

void	<u>acceptSession()</u> Accept the HTTP session.
void	<u>addHTTPSessionToTCPSession</u> (RtmpSessionInfo wmsSessionInfo, <u>IHTTPStreamerApplicationContext</u> appContext) Add HTTP session to TCP session for tracking
void	<u>addIOPerformance</u> (<u>IOPerformanceCounter</u> totalIOPerformanceResult) Internal user, keep track of IO performance
void	<u>addIOPerformance2</u> (<u>IOPerformanceCounter</u> totalIOPerformanceResult) Internal user, keep track of IO performance
void	<u>addStreamDomainStr</u> (String streamDomainStr) Internal user, add stream name
void	<u>addStreamDomainStrs</u> (java.util.List streamNames) Internal user, add stream names
void	<u>addUserHTTPHeaders</u> (<u>IHTTPResponse</u> resp) Internal
boolean	<u>checkAndSetPlayLogged()</u> If play has not been logged return false, else returns true, sets play has been logged
void	<u>clearLoggingValues()</u> Internal user, clear logging values
boolean	<u>containsStreamDomainStr</u> (String streamDomainStr) Internal user, test stream name
boolean	<u>containsStreamNameParts</u> (String streamName) Return true if stream name in stream name parts
void	<u>doSessionRedirect</u> (<u>IHTTPResponse</u> resp) Internal
void	<u>doSessionRedirect</u> (<u>IHTTPResponse</u> resp, String redirectSessionURL, String redirectSessionContentType, int redirectSessionCode) Internal
void	<u>extractHTTPRequestInfo</u> (<u>IHTTPRequest</u> req) Extract information from HTTP request
boolean	<u>getAndClearNotifyCreate()</u>
<u>IApplicationInstance</u>	<u>getAppInstance()</u> Get the application instance associated with this HTTP session
ConnectionHolder	<u>getConnectionHolder()</u> Connection holder for this session

String	<u>getCookieStr()</u> Get cookie string
DvrSessionInfo	<u>getDvrSessionInfo()</u> This information is used to manage the connection to the DVR store.
<u>ElapsedTimer</u>	<u>getElapsedTime()</u> Get the elapsed timer to see how long this session has been running
HTTPStreamerFileInfo	<u>getFileInfo(String streamName)</u> Get the file information if video on demand streaming
String	<u>getHTTPHeader(String name)</u> Get a HTTP header value such as 'Content-Length'
java.util.Map	<u>getHTTPHeaderMap()</u> Get a copy of the HTTP request header map
java.util.Set	<u>getHTTPHeaderNames()</u> Get a Set of the header names
int	<u>getHTTPIntHeader(String name)</u> Get a HTTP header value such as 'Content-Length' and return as int
<u>IHTTPStreamerAdapter</u>	<u>getHTTPStreamerAdapter()</u> Get the HTTP streaming adapter associated with this HTTP session
<u>IOPerformanceCounter</u>	<u>getIOPerformanceCounter()</u> Get IO performance counter
String	<u>getIpAddress()</u> Get the IP address
long	<u>getLastRequest()</u> Get the last timestamp of the last Io request
String	<u>getLiveStreamingPacketizer()</u> Get the live stream packetizer name
Object	<u>getLock()</u> Get the synchronization lock for this HTTP session
long	<u>getPlayDuration()</u> Get the play duration (milliseconds) for video on demand playback.
long	<u>getPlaySeek()</u> Get the seek location (milliseconds) where playback should start
long	<u>getPlayStart()</u> Get the play start time offset (milliseconds) for video on demand playback.
<u>WMSProperties</u>	<u>getProperties()</u> Get the properties associated with this session
String	<u>getQueryStr()</u> Get query string
byte[]	<u>getRedirectSessionBody()</u> Get redirect session body

int	<u>getRedirectSessionCode()</u> Get session redirect HTTP response code (default 302)
String	<u>getRedirectSessionContentType()</u> Get redirect session HTTP Content-Type
String	<u>getRedirectSessionURL()</u> Get redirect session URL
String	<u>getReferrer()</u> Get referrer
String	<u>getServerIp()</u> Get server IP address
int	<u>getServerPort()</u> Get server port
String	<u>getSessionId()</u> Get session id
int	<u>getSessionProtocol()</u> Get protocol, see SESSIONPROTOCOL_*
int	<u>getSessionTimeout()</u> Get the session timeout for this session (milliseconds)
int	<u>getSessionType()</u> Get session type: see SESSIONTYPE_*
<u>IMediaStream</u>	<u>getStream()</u> Get the IMediaStream associated with this HTTP session
String	<u>getStreamExt()</u> Get stream extension
String	<u>getStreamName()</u> Get stream name
HTTPStreamerStreamNameParts	<u>getStreamNameParts(String streamName)</u> Break the stream name into parts
long	<u>getStreamPosition()</u> Get stream position
String	<u>getTimeRunning()</u> Get the time this session has been running (milliseconds)
double	<u>getTimeRunningSeconds()</u> Get the time this session has been running (seconds)
<u>IHTTPStreamerTranscoderVODIndex</u>	<u>getTranscoderVODIndex()</u> Get the VOD Transcoder index information
<u>IHTTPStreamerTranscoderVODDestinationIndex</u>	<u>getTranscoderVODIndex(String bitrateStr)</u> Get VOD transcoder index for given bitrate
java.util.Map	<u>getTranscoderVODIndexDestinationsMap()</u> Get the VOD Transcoder destination map

java.util.List	<u>getTranscoderVODIndexDestinationsOrder()</u> Get the list of VOD Transcoder destinations in playlist order
com.wowza.wms.transcoder.vod.TranscoderVODSession	<u>getTranscoderVODSession()</u> Get the vod transcoder session associated with this HTTP session
String	<u>getUri()</u> Get the URI associated with initial request
String	<u>getUserAgent()</u> Get user agent
java.util.Map	<u>getUserHTTPHeaders()</u> Get user HTTP header.
String	<u>getUserQueryStr()</u> This query string will be added to URLs used in HTTP streaming
<u>IVHost</u>	<u>getVHost()</u> Get vhost
String	<u>getVODTranscodeNGRP()</u> Get the VOD Transcoder NGRP name
void	<u>indexTranscoderVOD(IHTTPStreamerApplicationContext appContext, String vodTranscodeNGRP)</u> Index the HTTP session for VOD transcoder streaming
boolean	<u>isAcceptSession()</u> Return true if this session has not been rejected
boolean	<u>isActive()</u> Is this session active, false after shutdown
boolean	<u>isFileInfo(String streamName)</u> Return true if the is file information for a given stream name
boolean	<u>isHTTPOrigin()</u> Return true if session in an HTTP origin session
boolean	<u>isPlayLogged()</u> true, if play has been logged
boolean	<u>isRedirectSession()</u> Is session redirect
boolean	<u>isSecure()</u> Returns true if the stream is being streamed over an SSL port
boolean	<u>isTimeout(long timecode)</u> Return true if this session is timed out.
boolean	<u>isTimeoutSession()</u> Get is session timeout.
boolean	<u>isValidated()</u> Has this session been validated

boolean	<u>isValidStreamDomainStr</u> (String streamDomainStr) Internal user, is stream name valid for HTTP session
void	<u>lockRepeaterStreams</u> (java.util.List streamNames, String liveStreamPacketizer, String liveStreamRepeater, String streamTypeStr) Internal user, lock in reapter streams
void	<u>putFileInfo</u> (String streamName, HTTPStreamerFileInfo fileInfo) Set the file information
void	<u>putStreamNameParts</u> (String streamName, HTTPStreamerStreamNameParts streamNameParts) Add stream name to stream name parts
void	<u>redirectSession</u> (String redirectSessionURL) Redirect session
void	<u>redirectSession</u> (String redirectSessionURL, int redirectSessionCode) Redirect session
void	<u>rejectSession</u> () Reject this HTTP session.
void	<u>removeStreamDomainStr</u> (String streamDomainStr) Internal user, remove stream name
void	<u>setAcceptSession</u> (boolean acceptSession) Set to false to reject session
void	<u>setActive</u> (boolean isActive) Set session active
void	<u>setAppInstance</u> (<u>IApplicationInstance</u> appInstance) Set the application instance associated with this HTTP session
void	<u>setCookieStr</u> (String cookieStr) Set cookie string
void	<u>setDvrSessionInfo</u> (DvrSessionInfo dvr) This information is used to manage the connection to the DVR store.
void	<u>setHTTPOrigin</u> (boolean isHTTPOrigin) Set to true if HTTP origin session
void	<u>setHTTPStreamerAdapter</u> (<u>IHTTPStreamerAdapter</u> httpStreamerAdapter) Set the HTTP streaming adapter associated with this HTTP session
void	<u>setIpAddress</u> (String ipAddress) Set the IP address
void	<u>setLiveStreamingPacketizer</u> (String liveStreamingPacketizer) Set the live stream packetizer name
void	<u>setPlayDuration</u> (long playDuration) Set the play duration (milliseconds) for video on demand playback.
void	<u>setPlayLogged</u> (boolean isPlayLogged) true, if play has been logged

void	<u>setPlaySeek</u> (long playSeek) Set the seek location (milliseconds) where playback should start
void	<u>setPlayStart</u> (long playStart) Set the play start time offset (milliseconds) for video on demand playback.
void	<u>setQueryStr</u> (String queryStr) Set query string
void	<u>setRedirectSession</u> (boolean redirectSession) Set session redirect
void	<u>setRedirectSessionBody</u> (byte[] redirectSessionBody) Set redirect session body
void	<u>setRedirectSessionCode</u> (int redirectSessionCode) Set session redirect HTTP response code (default 302)
void	<u>setRedirectSessionContentType</u> (String redirectSessionContentType) Set redirect session HTTP Content-Type
void	<u>setRedirectSessionURL</u> (String redirectSessionURL) Set redirect session URL
void	<u>setReferrer</u> (String referrer) Set referrer
void	<u>setSecure</u> (boolean isSecure) Set to true if the stream is being streamed over an SSL port
void	<u>setServerIp</u> (String serverIp) Set server IP address
void	<u>setServerPort</u> (int serverPort) Set server port
void	<u>setSessionId</u> (String sessionId) Set session id
void	<u>setSessionProtocol</u> (int sessionProtocol) Set protocol, see SESSIONPROTOCOL_*
void	<u>setSessionTimeout</u> (int sessionTimeout) Set the session timeout for this session (milliseconds)
void	<u>setSessionType</u> (int sessionType) Set session type: see SESSIONTYPE_*
void	<u>setStream</u> (<u>IMediaStream</u> stream) Set the IMediaStream associated with this HTTP session
void	<u>setStreamExt</u> (String streamExt) Set stream extension
void	<u>setStreamName</u> (String streamName) Set stream name
void	<u>setStreamPosition</u> (long streamPosition) Set stream position (will not cause seek)

void	<u>setTimeoutSession</u> (boolean timeoutSession) Set is session timeout.
void	<u>setTranscoderVODIndex</u> (<u>IHTTPStreamerTranscoderVODIndex</u> transcoderVODIndex) Set the VOD Transcoder index information
void	<u>setTranscoderVODSession</u> (com.wowza.wms.transcoder.vod.TranscoderVODSession transcoderVODSession) Set the vod transcoder session associated with this HTTP session
void	<u>setUri</u> (String uri) Set the URI associated with initial request
void	<u>setUserAgent</u> (String userAgent) Set user agent
void	<u>setUserHTTPHeader</u> (String name, String value) Set user HTTP header.
void	<u>setUserQueryStr</u> (String userQueryStr) This query string will be added to URLs used in HTTP streaming
void	<u>setVHost</u> (<u>IVHost</u> vhost) Set vhost
void	<u>setVODTranscodeNGRP</u> (String vodTranscodeNGRP) Set the VOD Transcoder NGRP name
void	<u>shutdown</u> () Called then the HTTP session is shutting down
void	<u>touch</u> (long timecode) Touch this session to keep it active.
void	<u>updateLoggingValues</u> () Internal user, update logging values
String	<u>validStreamDomainToString</u> () Return the valid domain strings as a string

Fields

SESSIONTYPE_UNKNOWN

```
public static final int SESSIONTYPE_UNKNOWN
```

Constant value: **0**

SESSIONTYPE_LIVE

```
public static final int SESSIONTYPE_LIVE
```

Constant value: **1**

(continued from last page)

SESSIONTYPE_VOD

```
public static final int SESSIONTYPE_VOD
```

Constant value: **2**

SESSIONTYPE_LIVEDVR

```
public static final int SESSIONTYPE_LIVEDVR
```

Constant value: **3**

SESSIONPROTOCOL_UNKNOWN

```
public static final int SESSIONPROTOCOL_UNKNOWN
```

Constant value: **-1**

SESSIONPROTOCOL_SMOOTHSTREAMING

```
public static final int SESSIONPROTOCOL_SMOOTHSTREAMING
```

Constant value: **0**

SESSIONPROTOCOL_CUPERTINOSTREAMING

```
public static final int SESSIONPROTOCOL_CUPERTINOSTREAMING
```

Constant value: **1**

SESSIONPROTOCOL_SANJOSESTREAMING

```
public static final int SESSIONPROTOCOL_SANJOSESTREAMING
```

Constant value: **2**

SESSIONPROTOCOL_WEBMSTREAMING

```
public static final int SESSIONPROTOCOL_WEBMSTREAMING
```

Constant value: **3**

SESSIONPROTOCOL_MPEGDASHSTREAMING

```
public static final int SESSIONPROTOCOL_MPEGDASHSTREAMING
```

Constant value: **4**

SESSIONPROTOCOL_DVRCHUNKSTREAMING

```
public static final int SESSIONPROTOCOL_DVRCHUNKSTREAMING
```

(continued from last page)

Constant value: **5**

SESSIONPROTOCOL_DIRECTSTREAMING

```
public static final int SESSIONPROTOCOL_DIRECTSTREAMING
```

Constant value: **6**

SESSIONPROTOCOL_COUNT

```
public static final int SESSIONPROTOCOL_COUNT
```

Constant value: **7**

Methods

getSessionId

```
public String getSessionId()
```

Get session id

Returns:

session id

setSessionId

```
public void setSessionId(String sessionId)
```

Set session id

Parameters:

sessionId - session id

getVHost

```
public IVHost getVHost()
```

Get vhost

Returns:

vhost

setVHost

```
public void setVHost(IVHost vhost)
```

Set vhost

Parameters:

vhost - vhost

touch

```
public void touch(long timecode)
```

(continued from last page)

Touch this session to keep it active.

Parameters:

timecode - timecode of touch - System.currentTimeMillis();

isTimeout

```
public boolean isTimeout(long timecode)
```

Return true if this session is timed out.

Parameters:

timecode - last touch - System.currentTimeMillis();

Returns:

true, if timed out

getLock

```
public Object getLock()
```

Get the synchronization lock for this HTTP session

Returns:

synchronization lock

shutdown

```
public void shutdown()
```

Called then the HTTP session is shutting down

isActive

```
public boolean isActive()
```

Is this session active, false after shutdown

Returns:

true if session is active

setActive

```
public void setActive(boolean isActive)
```

Set session active

Parameters:

isActive - session active

getStream

```
public IMediaStream getStream()
```

Get the IMediaStream associated with this HTTP session

Returns:

IMediaStream associated with this HTTP session

setStream

```
public void setStream(IMediaStream stream)
```

Set the IMediaStream associated with this HTTP session

Parameters:

stream - IMediaStream associated with this HTTP session

isTimeoutSession

```
public boolean isTimeoutSession()
```

Get is session timeout.

Returns:

true if this session is timed out

setTimeoutSession

```
public void setTimeoutSession(boolean timeoutSession)
```

Set is session timeout.

Parameters:

timeoutSession - true if this session is timed out

getSessionTimeout

```
public int getSessionTimeout()
```

Get the session timeout for this session (milliseconds)

Returns:

session timeout for this session (milliseconds)

setSessionTimeout

```
public void setSessionTimeout(int sessionTimeout)
```

Set the session timeout for this session (milliseconds)

Parameters:

sessionTimeout - session timeout for this session (milliseconds)

isValidated

```
public boolean isValidated()
```

Has this session been validated

Returns:

true, if validated

checkAndSetPlayLogged

```
public boolean checkAndSetPlayLogged()
```

(continued from last page)

If play has not been logged return false, else returns true, sets play has been logged

Returns:

true, if play has been logged

isPlayLogged

```
public boolean isPlayLogged()
```

true, if play has been logged

Returns:

true, if play has been logged

setPlayLogged

```
public void setPlayLogged(boolean isPlayLogged)
```

true, if play has been logged

Parameters:

isPlayLogged - true, if play has been logged

addIOPerformance

```
public void addIOPerformance(IOPerformanceCounter totalIOPerformanceResult)
```

Internal user, keep track of IO performance

Parameters:

totalIOPerformanceResult - IO performance

addIOPerformance2

```
public void addIOPerformance2(IOPerformanceCounter totalIOPerformanceResult)
```

Internal user, keep track of IO performance

Parameters:

totalIOPerformanceResult - IO performance

getConnectionHolder

```
public ConnectionHolder getConnectionHolder()
```

Connection holder for this session

Returns:

connection holder

getHTTPStreamerAdapter

```
public IHTTPStreamerAdapter getHTTPStreamerAdapter()
```

Get the HTTP streaming adapter associated with this HTTP session

Returns:

HTTP streaming adapter

setHTTPStreamerAdapter

```
public void setHTTPStreamerAdapter(IHTTPStreamerAdapter httpStreamerAdapter)
```

Set the HTTP streaming adapter associated with this HTTP session

Parameters:

httpStreamerAdapter - HTTP streaming adapter

getAppInstance

```
public IApplicationInstance getAppInstance()
```

Get the application instance associated with this HTTP session

Returns:

application instance associated with this HTTP session

setAppInstance

```
public void setAppInstance(IApplicationInstance appInstance)
```

Set the application instance associated with this HTTP session

Parameters:

appInstance - application instance associated with this HTTP session

getSessionType

```
public int getSessionType()
```

Get session type: see SESSIONTYPE_*

Returns:

session type: see SESSIONTYPE_*

setSessionType

```
public void setSessionType(int sessionType)
```

Set session type: see SESSIONTYPE_*

Parameters:

sessionType - session type: see SESSIONTYPE_*

getLiveStreamingPacketizer

```
public String getLiveStreamingPacketizer()
```

Get the live stream packetizer name

Returns:

live stream packetizer name

setLiveStreamingPacketizer

```
public void setLiveStreamingPacketizer(String liveStreamingPacketizer)
```

(continued from last page)

Set the live stream packetizer name

Parameters:

liveStreamingPacketizer - live stream packetizer name

getIpAddress

```
public String getIpAddress()
```

Get the IP address

Returns:

IP address

setIpAddress

```
public void setIpAddress(String ipAddress)
```

Set the IP address

Parameters:

ipAddress - IP address

updateLoggingValues

```
public void updateLoggingValues()
```

Internal user, update logging values

clearLoggingValues

```
public void clearLoggingValues()
```

Internal user, clear logging values

getSessionProtocol

```
public int getSessionProtocol()
```

Get protocol, see SESSIONPROTOCOL_*

Returns:

protocol, see SESSIONPROTOCOL_*

setSessionProtocol

```
public void setSessionProtocol(int sessionProtocol)
```

Set protocol, see SESSIONPROTOCOL_*

Parameters:

sessionProtocol - protocol, see SESSIONPROTOCOL_*

getServerIp

```
public String getServerIp()
```

Get server IP address

(continued from last page)

Returns:

server IP address

setServerIp

```
public void setServerIp(String serverIp)
```

Set server IP address

Parameters:

serverIp - server IP address

getServerPort

```
public int getServerPort()
```

Get server port

Returns:

server port

setServerPort

```
public void setServerPort(int serverPort)
```

Set server port

Parameters:

serverPort - server port

getUserAgent

```
public String getUserAgent()
```

Get user agent

Returns:

user agent

setUserAgent

```
public void setUserAgent(String userAgent)
```

Set user agent

Parameters:

userAgent - user agent

getUri

```
public String getUri()
```

Get the URI associated with initial request

Returns:

URI associated with initial request

setUri

```
public void setUri(String uri)
```

Set the URI associated with initial request

Parameters:

uri - URI associated with initial request

getReferrer

```
public String getReferrer()
```

Get referrer

Returns:

referrer

setReferrer

```
public void setReferrer(String referrer)
```

Set referrer

Parameters:

referrer - referrer

getQueryStr

```
public String getQueryStr()
```

Get query string

Returns:

query string

setQueryStr

```
public void setQueryStr(String queryStr)
```

Set query string

Parameters:

queryStr - query string

lockRepeaterStreams

```
public void lockRepeaterStreams(java.util.List streamNames,  
    String liveStreamPacketizer,  
    String liveStreamRepeater,  
    String streamTypeStr)
```

Internal user, lock in reapter streams

Parameters:

streamNames - list of stream names

liveStreamPacketizer - live stream packetizer name

liveStreamRepeater - live repeater name

(continued from last page)

streamTypeStr - stream type

rejectSession

```
public void rejectSession()
```

Reject this HTTP session. No further processing should occur

acceptSession

```
public void acceptSession()
```

Accept the HTTP session.

isAcceptSession

```
public boolean isAcceptSession()
```

Return true if this session has not been rejected

Returns:

true if this session has not been rejected

setAcceptSession

```
public void setAcceptSession(boolean acceptSession)
```

Set to false to reject session

Parameters:

acceptSession - false to reject session

getCookieStr

```
public String getCookieStr()
```

Get cookie string

Returns:

cookie string

setCookieStr

```
public void setCookieStr(String cookieStr)
```

Set cookie string

Parameters:

cookieStr - cookie string

getStreamName

```
public String getStreamName()
```

Get stream name

Returns:

stream name

setStreamName

```
public void setStreamName(String streamName)
```

Set stream name

Parameters:

streamName - stream name

getStreamExt

```
public String getStreamExt()
```

Get stream extension

Returns:

stream extension

setStreamExt

```
public void setStreamExt(String streamExt)
```

Set stream extension

Parameters:

streamExt - stream extension

getStreamNameParts

```
public HTTPStreamerStreamNameParts getStreamNameParts(String streamName)
```

Break the stream name into parts

Parameters:

streamName - stream name

Returns:

stream name parts

containsStreamNameParts

```
public boolean containsStreamNameParts(String streamName)
```

Return true if stream name in stream name parts

Parameters:

streamName - stream name

Returns:

true if stream name in stream name parts

putStreamNameParts

```
public void putStreamNameParts(String streamName,  
    HTTPStreamerStreamNameParts streamNameParts)
```

Add stream name to stream name parts

(continued from last page)

Parameters:

streamName - stream name

streamNameParts - stream name parts

getStreamPosition

```
public long getStreamPosition()
```

Get stream position

Returns:

stream position

setStreamPosition

```
public void setStreamPosition(long streamPosition)
```

Set stream position (will not cause seek)

Parameters:

streamPosition - stream position

getIOPerformanceCounter

```
public IOPerformanceCounter getIOPerformanceCounter()
```

Get IO performance counter

Returns:

IO performance counter

getFileInfo

```
public HTTPStreamerFileInfo getFileInfo(String streamName)
```

Get the file information if video on demand streaming

Parameters:

streamName - stream name

Returns:

file information

putFileInfo

```
public void putFileInfo(String streamName,  
    HTTPStreamerFileInfo fileInfo)
```

Set the file information

Parameters:

streamName - stream name

fileInfo - file information

isFileInfo

```
public boolean isFileInfo(String streamName)
```

Return true if the is file information for a given stream name

(continued from last page)

Parameters:

streamName - stream name

Returns:

true if the is file information for a given stream name

isValidStreamDomainStr

```
public boolean isValidStreamDomainStr(String streamDomainStr)
```

Internal user, is stream name valid for HTTP session

Parameters:

streamDomainStr - stream name

Returns:

true if valid

validStreamDomainToString

```
public String validStreamDomainToString()
```

Return the valid domain strings as a string

Returns:

valid domain strings as a string

containsStreamDomainStr

```
public boolean containsStreamDomainStr(String streamDomainStr)
```

Internal user, test stream name

Parameters:

streamDomainStr - stream name

Returns:

true if valid

removeStreamDomainStr

```
public void removeStreamDomainStr(String streamDomainStr)
```

Internal user, remove stream name

Parameters:

streamDomainStr - stream name

addStreamDomainStr

```
public void addStreamDomainStr(String streamDomainStr)
```

Internal user, add stream name

Parameters:

streamDomainStr - stream name

addStreamDomainStrs

```
public void addStreamDomainStrs(java.util.List streamNames)
```

Internal user, add stream names

Parameters:

streamNames - stream names

getElapsedTime

```
public ElapsedTimer getElapsedTime()
```

Get the elapsed timer to see how long this session has been running

Returns:

elapsed timer (milliseconds)

getTimeRunning

```
public String getTimeRunning()
```

Get the time this session has been running (milliseconds)

Returns:

time this session has been running (milliseconds)

getTimeRunningSeconds

```
public double getTimeRunningSeconds()
```

Get the time this session has been running (seconds)

Returns:

time this session has been running (seconds)

getProperties

```
public WMSProperties getProperties()
```

Get the properties associated with this session

Returns:

properties

getUserQueryStr

```
public String getUserQueryStr()
```

This query string will be added to URLs used in HTTP streaming

Returns:

user query string

setUserQueryStr

```
public void setUserQueryStr(String userQueryStr)
```

(continued from last page)

This query string will be added to URLs used in HTTP streaming

Parameters:

userQueryStr - user query string

setDvrSessionInfo

```
public void setDvrSessionInfo(DvrSessionInfo dvr)
```

This information is used to manage the connection to the DVR store.

Parameters:

dvr - The DVR session info.

getDvrSessionInfo

```
public DvrSessionInfo getDvrSessionInfo()
```

This information is used to manage the connection to the DVR store.

Returns:

DVR session info.

extractHTTPRequestInfo

```
public void extractHTTPRequestInfo(IHttpRequest req)
```

Extract information from HTTP request

Parameters:

req - HTTP request

getHTTPHeaderMap

```
public java.util.Map getHTTPHeaderMap()
```

Get a copy of the HTTP request header map

Returns:

copy of the HTTP request header map

getHTTPHeader

```
public String getHTTPHeader(String name)
```

Get a HTTP header value such as 'Content-Length'

Parameters:

name - header name

Returns:

header value

getHTTPIntHeader

```
public int getHTTPIntHeader(String name)
```

Get a HTTP header value such as 'Content-Length' and return as int

(continued from last page)

Parameters:

name - header name

Returns:

header value

getHTTPHeaderNames

```
public java.util.Set getHTTPHeaderNames()
```

Get a Set of the header names

Returns:

Set of header names

getPlayStart

```
public long getPlayStart()
```

Get the play start time offset (milliseconds) for video on demand playback.

Returns:

play start time offset (milliseconds)

setPlayStart

```
public void setPlayStart(long playStart)
```

Set the play start time offset (milliseconds) for video on demand playback.

Parameters:

playStart - play start time offset (milliseconds)

getPlayDuration

```
public long getPlayDuration()
```

Get the play duration (milliseconds) for video on demand playback. A values of -1 means play to end.

Returns:

play duration (milliseconds)

setPlayDuration

```
public void setPlayDuration(long playDuration)
```

Set the play duration (milliseconds) for video on demand playback. A values of -1 means play to end.

Parameters:

playDuration - play duration (milliseconds)

getPlaySeek

```
public long getPlaySeek()
```

Get the seek location (milliseconds) where playback should start

(continued from last page)

Returns:

seek location (milliseconds)

setPlaySeek

```
public void setPlaySeek(long playSeek)
```

Set the seek location (milliseconds) where playback should start

Parameters:

playSeek - seek location (milliseconds)

setUserHTTPHeader

```
public void setUserHTTPHeader(String name,  
    String value)
```

Set user HTTP header. This header value will be added to all HTTP responses

Parameters:name - name
value - value

getUserHTTPHeaders

```
public java.util.Map getUserHTTPHeaders()
```

Get user HTTP header. This header value will be added to all HTTP responses

Returns:

header map

addUserHTTPHeaders

```
public void addUserHTTPHeaders(IHTTPResponse resp)
```

Internal

Parameters:

resp - response

doSessionRedirect

```
public void doSessionRedirect(IHTTPResponse resp)
```

Internal

Parameters:

resp - response

doSessionRedirect

```
public void doSessionRedirect(IHTTPResponse resp,  
    String redirectSessionURL,  
    String redirectSessionContentType,  
    int redirectSessionCode)
```

Internal

(continued from last page)

Parameters:

resp - response
redirectSessionURL - redirect session URL
redirectSessionContentType
redirectSessionCode - redirect session response code (default 302)

isRedirectSession

```
public boolean isRedirectSession()
```

Is session redirect

Returns:

session redirect

setRedirectSession

```
public void setRedirectSession(boolean redirectSession)
```

Set session redirect

Parameters:

redirectSession - session redirect

getRedirectSessionCode

```
public int getRedirectSessionCode()
```

Get session redirect HTTP response code (default 302)

Returns:

session redirect HTTP response code

setRedirectSessionCode

```
public void setRedirectSessionCode(int redirectSessionCode)
```

Set session redirect HTTP response code (default 302)

Parameters:

redirectSessionCode - session redirect HTTP response code

getRedirectSessionURL

```
public String getRedirectSessionURL()
```

Get redirect session URL

Returns:

redirect session URL

setRedirectSessionURL

```
public void setRedirectSessionURL(String redirectSessionURL)
```

Set redirect session URL

Parameters:

(continued from last page)

`redirectSessionURL` - redirect session URL

redirectSession

```
public void redirectSession(String redirectSessionURL)
```

Redirect session

Parameters:

`redirectSessionURL` - redirect session URL

redirectSession

```
public void redirectSession(String redirectSessionURL,  
    int redirectSessionCode)
```

Redirect session

Parameters:

`redirectSessionURL` - redirect session URL

`redirectSessionCode` - redirect session response code (default 302)

getRedirectSessionBody

```
public byte[] getRedirectSessionBody()
```

Get redirect session body

Returns:

redirect session body

setRedirectSessionBody

```
public void setRedirectSessionBody(byte[] redirectSessionBody)
```

Set redirect session body

Parameters:

`redirectSessionBody` - redirect session body

getRedirectSessionContentType

```
public String getRedirectSessionContentType()
```

Get redirect session HTTP Content-Type

Returns:

redirect session HTTP Content-Type

setRedirectSessionContentType

```
public void setRedirectSessionContentType(String redirectSessionContentType)
```

Set redirect session HTTP Content-Type

Parameters:

`redirectSessionContentType`

getLastRequest

```
public long getLastRequest()
```

Get the last timestamp of the last Io request

Returns:

last timestamp of the last Io request (milliseconds)

isHTTPOrigin

```
public boolean isHTTPOrigin()
```

Return true if session in an HTTP origin session

Returns:

true if session in an HTTP origin session

setHTTPOrigin

```
public void setHTTPOrigin(boolean isHTTPOrigin)
```

Set to true if HTTP origin session

Parameters:

isHTTPOrigin - true if HTTP origin session

getAndClearNotifyCreate

```
public boolean getAndClearNotifyCreate()
```

getTranscoderVODSession

```
public com.wowza.wms.transcoder.vod.TranscoderVODSession getTranscoderVODSession()
```

Get the vod transcoder session associated with this HTTP session

Returns:

vod transcoder session

setTranscoderVODSession

```
public void setTranscoderVODSession(com.wowza.wms.transcoder.vod.TranscoderVODSession  
transcoderVODSession)
```

Set the vod transcoder session associated with this HTTP session

Parameters:

transcoderVODSession - vod transcoder session

getVODTranscodeNGRP

```
public String getVODTranscodeNGRP()
```

Get the VOD Transcoder NGRP name

(continued from last page)

Returns:

VOD Transcoder NGRP name

setVODTranscodeNGRP

```
public void setVODTranscodeNGRP(String vodTranscodeNGRP)
```

Set the VOD Transcoder NGRP name

Parameters:

vodTranscodeNGRP - VOD Transcoder NGRP name

getTranscoderVODIndex

```
public IHTTPStreamerTranscoderVODIndex getTranscoderVODIndex()
```

Get the VOD Transcoder index information

Returns:

VOD Transcoder index

setTranscoderVODIndex

```
public void setTranscoderVODIndex(IHTTPStreamerTranscoderVODIndex transcoderVODIndex)
```

Set the VOD Transcoder index information

Parameters:

transcoderVODIndex - VOD Transcoder index

getTranscoderVODIndexDestinationsOrder

```
public java.util.List getTranscoderVODIndexDestinationsOrder()
```

Get the list of VOD Transcoder destinations in playlist order

Returns:

list of VOD Transcoder destinations

getTranscoderVODIndexDestinationsMap

```
public java.util.Map getTranscoderVODIndexDestinationsMap()
```

Get the VOD Transcoder destination map

Returns:

VOD Transcoder destination map

indexTranscoderVOD

```
public void indexTranscoderVOD(IHTTPStreamerApplicationContext appContext,  
String vodTranscodeNGRP)
```

Index the HTTP session for VOD transcoder streaming

Parameters:

appContext - application context

vodTranscodeNGRP - transcoder name group

getTranscoderVODIndex

```
public IHTTPStreamerTranscoderVODDestinationIndex getTranscoderVODIndex(String  
bitrateStr)
```

Get VOD transcoder index for given bitrate

Parameters:

bitrateStr - bitrate string

Returns:

VOD transcoder index

addHTTPSessionToTCPSession

```
public void addHTTPSessionToTCPSession(RtmpSessionInfo wmsSessionInfo,  
IHTTPStreamerApplicationContext appContext)
```

Add HTTP session to TCP session for tracking

Parameters:

wmsSessionInfo - TCP session info

appContext - app context

isSecure

```
public boolean isSecure()
```

Returns true if the stream is being streamed over an SSL port

Returns:

true if the stream is being streamed over an SSL port

setSecure

```
public void setSecure(boolean isSecure)
```

Set to true if the stream is being streamed over an SSL port

Parameters:

isSecure - true if the stream is being streamed over an SSL port

com.wowza.wms.httpstreamer.model Interface IHTTPStreamerSessionNotify

public interface IHTTPStreamerSessionNotify
extends

IHTTPStreamerSessionNotify: HTTP session create/destroy interface

Method Summary

void	onHTTPStreamerSessionCreate (IHTTPStreamerSession httpStreamerSession) Called when an HTTP streaming session is created
void	onHTTPStreamerSessionDestroy (IHTTPStreamerSession httpStreamerSession) Called when an HTTP streaming session is destroyed

Methods

onHTTPStreamerSessionCreate

public void **onHTTPStreamerSessionCreate**([IHTTPStreamerSession](#) httpStreamerSession)

Called when an HTTP streaming session is created

Parameters:

httpStreamerSession - HTTP streaming session

onHTTPStreamerSessionDestroy

public void **onHTTPStreamerSessionDestroy**([IHTTPStreamerSession](#) httpStreamerSession)

Called when an HTTP streaming session is destroyed

Parameters:

httpStreamerSession - HTTP streaming session

com.wowza.wms.httpstreamer.model Interface IHTTPStreamerTranscoderVODDestinationIndex

public interface IHTTPStreamerTranscoderVODDestinationIndex
extends

Method Summary

int	<code>getBitrate()</code>
com.wowza.wms.transcoder.vod.TranscoderVODDestination	<code>getTranscoderVODDestination()</code>
void	<code>setBitrate(int bitrate)</code>
void	<code>setTranscoderVODDestination</code> (com.wowza.wms.transcoder.vod.TranscoderVODDestination transcoderVODDestination)

Methods

`getTranscoderVODDestination`

```
public com.wowza.wms.transcoder.vod.TranscoderVODDestination  
getTranscoderVODDestination()
```

`setTranscoderVODDestination`

```
public void  
setTranscoderVODDestination(com.wowza.wms.transcoder.vod.TranscoderVODDestination  
transcoderVODDestination)
```

`getBitrate`

```
public int getBitrate()
```

`setBitrate`

```
public void setBitrate(int bitrate)
```

com.wowza.wms.httpstreamer.model

Interface IHTTPStreamerTranscoderVODIndex

public interface IHTTPStreamerTranscoderVODIndex
extends

com.wowza.wms.httpstreamer.model Interface **ILiveStreamPacketizerChunkWriter**

public interface **ILiveStreamPacketizerChunkWriter**
extends

Method Summary

void	addToChunk (LiveStreamPacketizerPacketHolder holder)
void	endChunk (long timecode)
boolean	isPacketizeAudio ()
boolean	isPacketizeData ()
boolean	isPacketizeVideo ()
boolean	isValidAudioCodec (int codec)
boolean	isValidVideoCodec (int codec)
void	resetStream ()
void	setCodecInfoAudio (com.wowza.wms.media.model.MediaCodecInfoAudio codecInfoAudio)
void	setCodecInfoVideo (com.wowza.wms.media.model.MediaCodecInfoVideo codecInfoVideo)
void	startChunk (int streamMode, int videoCodecId, int audioCodecId, long timecode, String originRepeaterContext, CencInfo cencInfo)

Methods

startChunk

```
public void startChunk(int streamMode,  
    int videoCodecId,  
    int audioCodecId,  
    long timecode,  
    String originRepeaterContext,  
    CencInfo cencInfo)
```

(continued from last page)

endChunk

```
public void endChunk(long timecode)
```

addToChunk

```
public void addToChunk(LiveStreamPacketizerPacketHolder holder)
```

setCodecInfoAudio

```
public void setCodecInfoAudio(com.wowza.wms.media.model.MediaCodecInfoAudio  
codecInfoAudio)
```

setCodecInfoVideo

```
public void setCodecInfoVideo(com.wowza.wms.media.model.MediaCodecInfoVideo  
codecInfoVideo)
```

resetStream

```
public void resetStream()
```

isPacketizeAudio

```
public boolean isPacketizeAudio()
```

isPacketizeVideo

```
public boolean isPacketizeVideo()
```

isPacketizeData

```
public boolean isPacketizeData()
```

isValidAudioCodec

```
public boolean isValidAudioCodec(int codec)
```

(continued from last page)

isValidVideoCodec

```
public boolean isValidVideoCodec(int codec)
```

Package

com.wowza.wms.httpstreamer.mpegdashstreaming.httpstreamer

com.wowza.wms.httpstreamer.mpegdashstreaming.httpstreamer

Class HTTPStreamerSessionMPEGDash

java.lang.Object

└-com.wowza.wms.httpstreamer.model.HTTPStreamerSessionBase

└-

com.wowza.wms.httpstreamer.mpegdashstreaming.httpstreamer.HTTPStreamerSessionMPEGDash

All Implemented Interfaces:

[IHTTPSessionNotify](#), [IHTTPStreamerSession](#)

public class **HTTPStreamerSessionMPEGDash**
extends HTTPStreamerSessionBase
implements [IHTTPStreamerSession](#), [IHTTPSessionNotify](#)

Fields inherited from class com.wowza.wms.httpstreamer.model.HTTPStreamerSessionBase
acceptSession, appInstance, connectionHolder, cookieStr, DATEFORMAT, elapsedTime, fastDateFormat, fileInfoMap, firstCheck, httpHeaders, httpStreamerAdapter, ioPerformanceCounter, ipAddress, isActive, isHTTPOrigin, isPlayLogged, isSecure, lastRequest, liveStreamingPacketizer, lock, mediaCasterStreamLock, notifyCreate, playDuration, playSeek, playStart, properties, queryStr, redirectSession, redirectSessionBody, redirectSessionCode, redirectSessionContentType, redirectSessionURL, referrer, serverIp, serverPort, sessionId, sessionProtocol, sessionTimeout, sessionType, stream, streamDomainStrSet, streamExt, streamName, streamNamePartMap, streamPosition, timeoutSession, totalIOPerformance2Last, totalIOPerformanceLast, transcoderVODIndex, transcoderVODIndexDestinationsMap, transcoderVODIndexDestinationsOrder, transcoderVODSession, uri, userAgent, userHTTPHeaders, userQueryStr, vhost, vodTranscodeNGRP

Fields inherited from interface com.wowza.wms.httpstreamer.model.IHTTPStreamerSession
SESSIONPROTOCOL_COUNT , SESSIONPROTOCOL_CUPERTINOSTREAMING , SESSIONPROTOCOL_DIRECTSTREAMING , SESSIONPROTOCOL_DVRCHUNKSTREAMING , SESSIONPROTOCOL_MPEGDASHSTREAMING , SESSIONPROTOCOL_SANJOSESTREAMING , SESSIONPROTOCOL_SMOOTHSTREAMING , SESSIONPROTOCOL_UNKNOWN , SESSIONPROTOCOL_WEBMSTREAMING , SESSIONTYPE_LIVE , SESSIONTYPE_LIVEDVR , SESSIONTYPE_UNKNOWN , SESSIONTYPE_VOD

Constructor Summary	
public	HTTPStreamerSessionMPEGDash()

Method Summary	
void	clearLoggingValues()
boolean	containsIndex (String streamName)
static IHTTPStreamerMPEGDash Index	createIndexLive (IHTTPStreamerApplicationContext appContext, IHTTPStreamerSession httpStreamerSession, String rawStreamName, String streamExt, String streamName)

IHTTPStreamerMPEGDash Index	getIndex(IHTTPStreamerApplicationContext appContext, IHTTPStreamerSession httpStreamerSession, String rawStreamName, String streamExt, String streamName, long playStart, long playDuration, long playSeek, TimedTextRequest captionRequest, String vodTranscodeNGRP)
IHTTPStreamerMPEGDash Index	getIndexLive(IHTTPStreamerApplicationContext appContext, IHTTPStreamerSession httpStreamerSession, String rawStreamName, String streamExt, String streamName)
void	indexTranscoderVOD(IHTTPStreamerApplicationContext appContext, String vodTranscodeNGRP)
void	logLiveChunk (MPEGDashDashChunk chunk)
void	logVODChunk (MPEGDashDashChunk chunk)
void	notifyHTTPSessionCreate(IApplicationInstance appInstance, IHTTPStreamerSession httpStreamerSession)
void	notifyHTTPSessionDestroy(IApplicationInstance appInstance, IHTTPStreamerSession httpStreamerSession)
void	shutdown ()
void	updateLoggingValues ()

Methods inherited from class com.wowza.wms.httpstreamer.model.HTTPStreamerSessionBase

```

acceptSession, addHTTPSessionToTCPSession, addIOPerformance, addIOPerformance2,
addStreamDomainStr, addStreamDomainStrs, addUserHTTPHeaders, checkAndSetPlayLogged,
clearLoggingValues, containsStreamDomainStr, containsStreamNameParts,
doSessionRedirect, doSessionRedirect, extractHTTPRequestInfo,
getAndClearNotifyCreate, getAppInstance, getConnectionHolder, getCookieStr,
getDvrSessionInfo, getElapsedTime, getFileInfo, getHTTPDate, getHTTPHeader,
getHTTPHeaderMap, getHTTPHeaderNames, getHTTPIntHeader, getHTTPStreamerAdapter,
getIOPerformanceCounter, getIpAddress, getLastRequest, getLiveStreamingPacketizer,
getLock, getPlayDuration, getPlaySeek, getPlayStart, getProperties, getQueryStr,
getRedirectSessionBody, getRedirectSessionCode, getRedirectSessionContentType,
getRedirectSessionURL, getReferrer, getServerIp, getServerPort, getSessionId,
getSessionProtocol, getSessionTimeout, getSessionType, getStream, getStreamDomainStr,
getStreamDomainStrList, getStreamExt, getStreamName, getStreamNameParts,
getStreamPosition, getTimeRunning, getTimeRunningSeconds, getTranscoderVODIndex,
getTranscoderVODIndex, getTranscoderVODIndexDestinationsMap,
getTranscoderVODIndexDestinationsOrder, getTranscoderVODSession, getUri,
getUserAgent, getUserHTTPHeaders, getUserQueryStr, getVHost, getVODTranscodeNGRP,
indexTranscoderVOD, isAcceptSession, isActive, isFileInfo, isHTTPOrigin,
isPlayLogged, isRedirectSession, isSecure, isTimeout, isTimeoutSession, isValidated,
isValidStreamDomainStr, lockRepeaterStreams, putFileInfo, putStreamNameParts,
redirectSession, redirectSession, rejectSession, removeStreamDomainStr,
setAcceptSession, setActive, setAppInstance, setCookieStr, setDvrSessionInfo,
setHTTPOrigin, setHTTPStreamerAdapter, setIpAddress, setLiveStreamingPacketizer,
setPlayDuration, setPlayLogged, setPlaySeek, setPlayStart, setQueryStr,
setRedirectSession, setRedirectSessionBody, setRedirectSessionCode,
setRedirectSessionContentType, setRedirectSessionURL, setReferrer, setSecure,
setServerIp, setServerPort, setSessionId, setSessionProtocol, setSessionTimeout,
setSessionType, setStream, setStreamExt, setStreamName, setStreamPosition,
setThreadContext, setTimeoutSession, setTranscoderVODIndex, setTranscoderVODSession,
setUri, setUserAgent, setUserHTTPHeader, setUserQueryStr, setValidated, setVHost,
setVODTranscodeNGRP, shutdown, shutdownLocked, touch, updateLoggingValues,
validStreamDomainToString

```

Methods inherited from class java.lang.Object

```

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait,
wait

```

Methods inherited from interface [com.wowza.wms.httpstreamer.model.IHTTPStreamerSession](#)

[acceptSession](#), [addHTTPSessionToTCPSession](#), [addIOPerformance](#), [addIOPerformance2](#),
[addStreamDomainStr](#), [addStreamDomainStrs](#), [addUserHTTPHeaders](#), [checkAndSetPlayLogged](#),
[clearLoggingValues](#), [containsStreamDomainStr](#), [containsStreamNameParts](#),
[doSessionRedirect](#), [doSessionRedirect](#), [extractHTTPRequestInfo](#),
[getAndClearNotifyCreate](#), [getAppInstance](#), [getConnectionHolder](#), [getCookieStr](#),
[getDvrSessionInfo](#), [getElapsedTime](#), [getFileInfo](#), [getHTTPHeader](#), [getHTTPHeaderMap](#),
[getHTTPHeaderNames](#), [getHTTPIntHeader](#), [getHTTPStreamerAdapter](#),
[getIOPerformanceCounter](#), [getIpAddress](#), [getLastRequest](#), [getLiveStreamingPacketizer](#),
[getLock](#), [getPlayDuration](#), [getPlaySeek](#), [getPlayStart](#), [getProperties](#), [getQueryStr](#),
[getRedirectSessionBody](#), [getRedirectSessionCode](#), [getRedirectSessionContentType](#),
[getRedirectSessionURL](#), [getReferrer](#), [getServerIp](#), [getServerPort](#), [getSessionId](#),
[getSessionProtocol](#), [getSessionTimeout](#), [getSessionType](#), [getStream](#), [getStreamExt](#),
[getStreamName](#), [getStreamNameParts](#), [getStreamPosition](#), [getTimeRunning](#),
[getTimeRunningSeconds](#), [getTranscoderVODIndex](#), [getTranscoderVODIndex](#),
[getTranscoderVODIndexDestinationsMap](#), [getTranscoderVODIndexDestinationsOrder](#),
[getTranscoderVODSession](#), [getUri](#), [getUserAgent](#), [getUserHTTPHeaders](#), [getUserQueryStr](#),
[getVHost](#), [getVODTranscodeNGRP](#), [indexTranscoderVOD](#), [isAcceptSession](#), [isActive](#),
[isFileInfo](#), [isHTTPOrigin](#), [isPlayLogged](#), [isRedirectSession](#), [isSecure](#), [isTimeout](#),
[isTimeoutSession](#), [isValidated](#), [isValidStreamDomainStr](#), [lockRepeaterStreams](#),
[putFileInfo](#), [putStreamNameParts](#), [redirectSession](#), [redirectSession](#), [rejectSession](#),
[removeStreamDomainStr](#), [setAcceptSession](#), [setActive](#), [setAppInstance](#), [setCookieStr](#),
[setDvrSessionInfo](#), [setHTTPOrigin](#), [setHTTPStreamerAdapter](#), [setIpAddress](#),
[setLiveStreamingPacketizer](#), [setPlayDuration](#), [setPlayLogged](#), [setPlaySeek](#),
[setPlayStart](#), [setQueryStr](#), [setRedirectSession](#), [setRedirectSessionBody](#),
[setRedirectSessionCode](#), [setRedirectSessionContentType](#), [setRedirectSessionURL](#),
[setReferrer](#), [setSecure](#), [setServerIp](#), [setServerPort](#), [setSessionId](#), [setSessionProtocol](#),
[setSessionTimeout](#), [setSessionType](#), [setStream](#), [setStreamExt](#), [setStreamName](#),
[setStreamPosition](#), [setTimeoutSession](#), [setTranscoderVODIndex](#), [setTranscoderVODSession](#),
[setUri](#), [setUserAgent](#), [setUserHTTPHeader](#), [setUserQueryStr](#), [setVHost](#),
[setVODTranscodeNGRP](#), [shutdown](#), [touch](#), [updateLoggingValues](#), [validStreamDomainToString](#)

Methods inherited from interface [com.wowza.wms.httpstreamer.model.IHTTPSessionNotify](#)

[notifyHTTPSessionCreate](#), [notifyHTTPSessionDestroy](#)

Constructors

HTTPStreamerSessionMPEGDash

```
public HTTPStreamerSessionMPEGDash()
```

Methods

shutdown

```
public void shutdown()
```

containsIndex

```
public boolean containsIndex(String streamName)
```

(continued from last page)

createIndexLive

```
public static IHTTPStreamerMPEGDashIndex
createIndexLive(IHTTPStreamerApplicationContext appContext,
IHTTPStreamerSession httpStreamerSession,
String rawStreamName,
String streamExt,
String streamName)
```

getIndexLive

```
public IHTTPStreamerMPEGDashIndex getIndexLive(IHTTPStreamerApplicationContext
appContext,
IHTTPStreamerSession httpStreamerSession,
String rawStreamName,
String streamExt,
String streamName)
```

getIndex

```
public IHTTPStreamerMPEGDashIndex getIndex(IHTTPStreamerApplicationContext appContext,
IHTTPStreamerSession httpStreamerSession,
String rawStreamName,
String streamExt,
String streamName,
long playStart,
long playDuration,
long playSeek,
TimedTextRequest captionRequest,
String vodTranscodeNGRP)
```

updateLoggingValues

```
public void updateLoggingValues()
```

clearLoggingValues

```
public void clearLoggingValues()
```

logLiveChunk

```
public void logLiveChunk(MPEGDashDashChunk chunk)
```

logVODChunk

```
public void logVODChunk(MPEGDashDashChunk chunk)
```

notifyHTTPSessionCreate

```
public void notifyHTTPSessionCreate(IApplicationInstance appInstance,  
    IHTTPStreamerSession httpStreamerSession)
```

notifyHTTPSessionDestroy

```
public void notifyHTTPSessionDestroy(IApplicationInstance appInstance,  
    IHTTPStreamerSession httpStreamerSession)
```

indexTranscoderVOD

```
public void indexTranscoderVOD(IHTTPStreamerApplicationContext appContext,  
    String vodTranscodeNGRP)
```

Package

**com.wowza.wms.httpstreamer.sanjoestream
ing.httpstreamer**

com.wowza.wms.httpstreamer.sanjosestreaming.httpstreamer
Class HTTPStreamerSessionSanJose

java.lang.Object
└-com.wowza.wms.httpstreamer.model.HTTPStreamerSessionBase
└-
com.wowza.wms.httpstreamer.sanjosestreaming.httpstreamer.HTTPStreamerSessionSanJose

All Implemented Interfaces:
[IHTTPSessionNotify](#), [IHTTPStreamerSession](#)

public class **HTTPStreamerSessionSanJose**
extends HTTPStreamerSessionBase
implements [IHTTPStreamerSession](#), [IHTTPSessionNotify](#)

Fields inherited from class com.wowza.wms.httpstreamer.model.HTTPStreamerSessionBase
acceptSession, appInstance, connectionHolder, cookieStr, DATEFORMAT, elapsedTime, fastDateFormat, fileInfoMap, firstCheck, httpHeaders, httpStreamerAdapter, ioPerformanceCounter, ipAddress, isActive, isHTTPOrigin, isPlayLogged, isSecure, lastRequest, liveStreamingPacketizer, lock, mediaCasterStreamLock, notifyCreate, playDuration, playSeek, playStart, properties, queryStr, redirectSession, redirectSessionBody, redirectSessionCode, redirectSessionContentType, redirectSessionURL, referrer, serverIp, serverPort, sessionId, sessionProtocol, sessionTimeout, sessionType, stream, streamDomainStrSet, streamExt, streamName, streamNamePartMap, streamPosition, timeoutSession, totalIOPerformance2Last, totalIOPerformanceLast, transcoderVODIndex, transcoderVODIndexDestinationsMap, transcoderVODIndexDestinationsOrder, transcoderVODSession, uri, userAgent, userHTTPHeaders, userQueryStr, vhost, vodTranscodeNGRP

Fields inherited from interface com.wowza.wms.httpstreamer.model.IHTTPStreamerSession
SESSIONPROTOCOL_COUNT , SESSIONPROTOCOL_CUPERTINOSTREAMING , SESSIONPROTOCOL_DIRECTSTREAMING , SESSIONPROTOCOL_DVRCHUNKSTREAMING , SESSIONPROTOCOL_MPEGDASHSTREAMING , SESSIONPROTOCOL_SANJOSESTREAMING , SESSIONPROTOCOL_SMOOTHSTREAMING , SESSIONPROTOCOL_UNKNOWN , SESSIONPROTOCOL_WEBMSTREAMING , SESSIONTYPE_LIVE , SESSIONTYPE_LIVEDVR , SESSIONTYPE_UNKNOWN , SESSIONTYPE_VOD

Constructor Summary	
public	HTTPStreamerSessionSanJose()

Method Summary	
void	addFirstABSTRequest (String streamName)
void	clearLoggingValues ()
boolean	containsIndex (String streamName)

static IHTTPStreamerSanJoseIndex	createIndexLive(IHTTPStreamerApplicationContext appContext, IHTTPStreamerSession httpStreamerSession, String rawStreamName, String streamExt, String streamName)
IHTTPStreamerSanJoseIndex	getIndex(IHTTPStreamerApplicationContext appContext, IHTTPStreamerSession httpStreamerSession, String rawStreamName, String streamExt, String streamName, long playStart, long playDuration, long playSeek, TimedTextRequest captionRequest, String vodTranscodeNGRP)
IHTTPStreamerSanJoseIndex	getIndexLive(IHTTPStreamerApplicationContext appContext, IHTTPStreamerSession httpStreamerSession, String rawStreamName, String streamExt, String streamName)
void	indexTranscoderVOD(IHTTPStreamerApplicationContext appContext, String vodTranscodeNGRP)
boolean	isFirstABSTRequest(String streamName)
void	logLiveChunk(LiveStreamPacketizerSanJoseChunk chunk)
void	logVODChunk(LiveStreamPacketizerSanJoseChunk chunk)
void	notifyHTTPSessionCreate(IApplicationInstance appInstance, IHTTPStreamerSession httpStreamerSession)
void	notifyHTTPSessionDestroy(IApplicationInstance appInstance, IHTTPStreamerSession httpStreamerSession)
void	shutdown()
void	updateLoggingValues()

Methods inherited from class `com.wowza.wms.httpstreamer.model.HTTPStreamerSessionBase`

```

acceptSession, addHTTPSessionToTCPSession, addIOPerformance, addIOPerformance2,
addStreamDomainStr, addStreamDomainStrs, addUserHTTPHeaders, checkAndSetPlayLogged,
clearLoggingValues, containsStreamDomainStr, containsStreamNameParts,
doSessionRedirect, doSessionRedirect, extractHTTPRequestInfo,
getAndClearNotifyCreate, getAppInstance, getConnectionHolder, getCookieStr,
getDvrSessionInfo, getElapsedTime, getFileInfo, getHTTPDate, getHTTPHeader,
getHTTPHeaderMap, getHTTPHeaderNames, getHTTPIntHeader, getHTTPStreamerAdapter,
getIOPerformanceCounter, getIpAddress, getLastRequest, getLiveStreamingPacketizer,
getLock, getPlayDuration, getPlaySeek, getPlayStart, getProperties, getQueryStr,
getRedirectSessionBody, getRedirectSessionCode, getRedirectSessionContentType,
getRedirectSessionURL, getReferrer, getServerIp, getServerPort, getSessionId,
getSessionProtocol, getSessionTimeout, getSessionType, getStream, getStreamDomainStr,
getStreamDomainStrList, getStreamExt, getStreamName, getStreamNameParts,
getStreamPosition, getTimeRunning, getTimeRunningSeconds, getTranscoderVODIndex,
getTranscoderVODIndex, getTranscoderVODIndexDestinationsMap,
getTranscoderVODIndexDestinationsOrder, getTranscoderVODSession, getUri,
getUserAgent, getUserHTTPHeaders, getUserQueryStr, getVHost, getVODTranscodeNGRP,
indexTranscoderVOD, isAcceptSession, isActive, isFileInfo, isHTTPOrigin,
isPlayLogged, isRedirectSession, isSecure, isTimeout, isTimeoutSession, isValidated,
isValidStreamDomainStr, lockRepeaterStreams, putFileInfo, putStreamNameParts,
redirectSession, redirectSession, rejectSession, removeStreamDomainStr,
setAcceptSession, setActive, setAppInstance, setCookieStr, setDvrSessionInfo,
setHTTPOrigin, setHTTPStreamerAdapter, setIpAddress, setLiveStreamingPacketizer,
setPlayDuration, setPlayLogged, setPlaySeek, setPlayStart, setQueryStr,
setRedirectSession, setRedirectSessionBody, setRedirectSessionCode,
setRedirectSessionContentType, setRedirectSessionURL, setReferrer, setSecure,
setServerIp, setServerPort, setSessionId, setSessionProtocol, setSessionTimeout,
setSessionType, setStream, setStreamExt, setStreamName, setStreamPosition,
setThreadContext, setTimeoutSession, setTranscoderVODIndex, setTranscoderVODSession,
setUri, setUserAgent, setUserHTTPHeader, setUserQueryStr, setValidated, setVHost,
setVODTranscodeNGRP, shutdown, shutdownLocked, touch, updateLoggingValues,
validStreamDomainToString

```

Methods inherited from class java.lang.Object

```

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait,
wait

```

Methods inherited from interface [com.wowza.wms.httpstreamer.model.IHTTPStreamerSession](#)

[acceptSession](#), [addHTTPSessionToTCPSession](#), [addIOPerformance](#), [addIOPerformance2](#),
[addStreamDomainStr](#), [addStreamDomainStrs](#), [addUserHTTPHeaders](#), [checkAndSetPlayLogged](#),
[clearLoggingValues](#), [containsStreamDomainStr](#), [containsStreamNameParts](#),
[doSessionRedirect](#), [doSessionRedirect](#), [extractHTTPRequestInfo](#),
[getAndClearNotifyCreate](#), [getAppInstance](#), [getConnectionHolder](#), [getCookieStr](#),
[getDvrSessionInfo](#), [getElapsedTime](#), [getFileInfo](#), [getHTTPHeader](#), [getHTTPHeaderMap](#),
[getHTTPHeaderNames](#), [getHTTPIntHeader](#), [getHTTPStreamerAdapter](#),
[getIOPerformanceCounter](#), [getIpAddress](#), [getLastRequest](#), [getLiveStreamingPacketizer](#),
[getLock](#), [getPlayDuration](#), [getPlaySeek](#), [getPlayStart](#), [getProperties](#), [getQueryStr](#),
[getRedirectSessionBody](#), [getRedirectSessionCode](#), [getRedirectSessionContentType](#),
[getRedirectSessionURL](#), [getReferrer](#), [getServerIp](#), [getServerPort](#), [getSessionId](#),
[getSessionProtocol](#), [getSessionTimeout](#), [getSessionType](#), [getStream](#), [getStreamExt](#),
[getStreamName](#), [getStreamNameParts](#), [getStreamPosition](#), [getTimeRunning](#),
[getTimeRunningSeconds](#), [getTranscoderVODIndex](#), [getTranscoderVODIndex](#),
[getTranscoderVODIndexDestinationsMap](#), [getTranscoderVODIndexDestinationsOrder](#),
[getTranscoderVODSession](#), [getUri](#), [getUserAgent](#), [getUserHTTPHeaders](#), [getUserQueryStr](#),
[getVHost](#), [getVODTranscodeNGRP](#), [indexTranscoderVOD](#), [isAcceptSession](#), [isActive](#),
[isFileInfo](#), [isHTTPOrigin](#), [isPlayLogged](#), [isRedirectSession](#), [isSecure](#), [isTimeout](#),
[isTimeoutSession](#), [isValidated](#), [isValidStreamDomainStr](#), [lockRepeaterStreams](#),
[putFileInfo](#), [putStreamNameParts](#), [redirectSession](#), [redirectSession](#), [rejectSession](#),
[removeStreamDomainStr](#), [setAcceptSession](#), [setActive](#), [setAppInstance](#), [setCookieStr](#),
[setDvrSessionInfo](#), [setHTTPOrigin](#), [setHTTPStreamerAdapter](#), [setIpAddress](#),
[setLiveStreamingPacketizer](#), [setPlayDuration](#), [setPlayLogged](#), [setPlaySeek](#),
[setPlayStart](#), [setQueryStr](#), [setRedirectSession](#), [setRedirectSessionBody](#),
[setRedirectSessionCode](#), [setRedirectSessionContentType](#), [setRedirectSessionURL](#),
[setReferrer](#), [setSecure](#), [setServerIp](#), [setServerPort](#), [setSessionId](#), [setSessionProtocol](#),
[setSessionTimeout](#), [setSessionType](#), [setStream](#), [setStreamExt](#), [setStreamName](#),
[setStreamPosition](#), [setTimeoutSession](#), [setTranscoderVODIndex](#), [setTranscoderVODSession](#),
[setUri](#), [setUserAgent](#), [setUserHTTPHeader](#), [setUserQueryStr](#), [setVHost](#),
[setVODTranscodeNGRP](#), [shutdown](#), [touch](#), [updateLoggingValues](#), [validStreamDomainToString](#)

Methods inherited from interface [com.wowza.wms.httpstreamer.model.IHTTPSessionNotify](#)

[notifyHTTPSessionCreate](#), [notifyHTTPSessionDestroy](#)

Constructors

HTTPStreamerSessionSanJose

```
public HTTPStreamerSessionSanJose()
```

Methods

shutdown

```
public void shutdown()
```

containsIndex

```
public boolean containsIndex(String streamName)
```

(continued from last page)

createIndexLive

```
public static IHTTPStreamerSanJoseIndex  
createIndexLive(IHTTPStreamerApplicationContext appContext,  
                IHTTPStreamerSession httpStreamerSession,  
                String rawStreamName,  
                String streamExt,  
                String streamName)
```

getIndexLive

```
public IHTTPStreamerSanJoseIndex getIndexLive(IHTTPStreamerApplicationContext  
appContext,  
        IHTTPStreamerSession httpStreamerSession,  
        String rawStreamName,  
        String streamExt,  
        String streamName)
```

getIndex

```
public IHTTPStreamerSanJoseIndex getIndex(IHTTPStreamerApplicationContext appContext,  
        IHTTPStreamerSession httpStreamerSession,  
        String rawStreamName,  
        String streamExt,  
        String streamName,  
        long playStart,  
        long playDuration,  
        long playSeek,  
        TimedTextRequest captionRequest,  
        String vodTranscodeNGRP)
```

updateLoggingValues

```
public void updateLoggingValues()
```

clearLoggingValues

```
public void clearLoggingValues()
```

logLiveChunk

```
public void logLiveChunk(LiveStreamPacketizerSanJoseChunk chunk)
```

logVODChunk

```
public void logVODChunk(LiveStreamPacketizerSanJoseChunk chunk)
```

isFirstABSTRequest

```
public boolean isFirstABSTRequest(String streamName)
```

addFirstABSTRequest

```
public void addFirstABSTRequest(String streamName)
```

notifyHTTPSessionCreate

```
public void notifyHTTPSessionCreate(IApplicationInstance appInstance,  
    IHTTPStreamerSession httpStreamerSession)
```

notifyHTTPSessionDestroy

```
public void notifyHTTPSessionDestroy(IApplicationInstance appInstance,  
    IHTTPStreamerSession httpStreamerSession)
```

indexTranscoderVOD

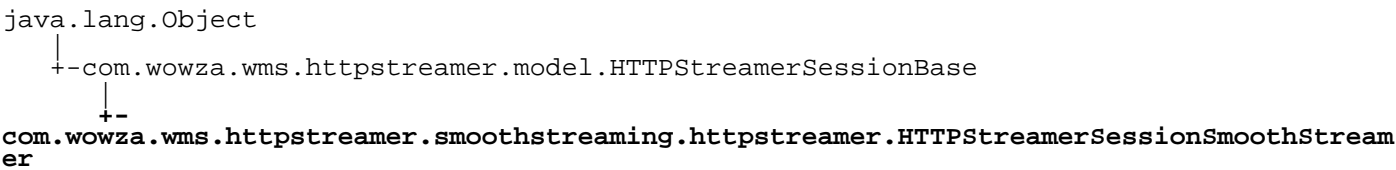
```
public void indexTranscoderVOD(IHTTPStreamerApplicationContext appContext,  
    String vodTranscodeNGRP)
```

Package

**com.wowza.wms.httpstreamer.smoothstream
ing.httpstreamer**

com.wowza.wms.httpstreamer.smoothstreaming.httpstreamer

Class HTTPStreamerSessionSmoothStreamer



All Implemented Interfaces:
[IHTTPSessionNotify](#), [IHTTPStreamerSession](#)

```
public class HTTPStreamerSessionSmoothStreamer
extends HTTPStreamerSessionBase
implements IHTTPStreamerSession, IHTTPSessionNotify
```

Fields inherited from class com.wowza.wms.httpstreamer.model.HTTPStreamerSessionBase
acceptSession, appInstance, connectionHolder, cookieStr, DATEFORMAT, elapsedTime, fastDateFormat, fileInfoMap, firstCheck, httpHeaders, httpStreamerAdapter, ioPerformanceCounter, ipAddress, isActive, isHTTPOrigin, isPlayLogged, isSecure, lastRequest, liveStreamingPacketizer, lock, mediaCasterStreamLock, notifyCreate, playDuration, playSeek, playStart, properties, queryStr, redirectSession, redirectSessionBody, redirectSessionCode, redirectSessionContentType, redirectSessionURL, referrer, serverIp, serverPort, sessionId, sessionProtocol, sessionTimeout, sessionType, stream, streamDomainStrSet, streamExt, streamName, streamNamePartMap, streamPosition, timeoutSession, totalIOPerformance2Last, totalIOPerformanceLast, transcoderVODIndex, transcoderVODIndexDestinationsMap, transcoderVODIndexDestinationsOrder, transcoderVODSession, uri, userAgent, userHTTPHeaders, userQueryStr, vhost, vodTranscodeNGRP

Fields inherited from interface com.wowza.wms.httpstreamer.model.IHTTPStreamerSession
SESSIONPROTOCOL_COUNT , SESSIONPROTOCOL_CUPERTINOSTREAMING , SESSIONPROTOCOL_DIRECTSTREAMING , SESSIONPROTOCOL_DVRCHUNKSTREAMING , SESSIONPROTOCOL_MPEGDASHSTREAMING , SESSIONPROTOCOL_SANJOSESTREAMING , SESSIONPROTOCOL_SMOOTHSTREAMING , SESSIONPROTOCOL_UNKNOWN , SESSIONPROTOCOL_WEBMSTREAMING , SESSIONTYPE_LIVE , SESSIONTYPE_LIVEDVR , SESSIONTYPE_UNKNOWN , SESSIONTYPE_VOD

Constructor Summary	
public	HTTPStreamerSessionSmoothStreamer()

Method Summary	
void	clearLoggingValues()
boolean	containsIndex (String streamName)

IHTTPStreamerSmoothStreamerIndex	getIndex(IHTTPStreamerSession httpStreamerSession, IHTTPStreamerApplicationContext appContext, String streamExt, String streamName, long playStart, long playDuration, long playSeek, String vodTranscodeNGRP)
SmoothStreamingLivePlaylist	getLivePlaylist()
Object	getPublishData()
void	indexTranscoderVOD(IHTTPStreamerApplicationContext appContext, String vodTranscodeNGRP)
void	logLiveFragment(SmoothStreamerFragmentId fragmentId, PacketFragmentList fragmentData)
void	logVODFragment(SmoothStreamerFragmentId fragmentId, PacketFragmentList fragmentData)
void	notifyHTTPSessionCreate(IApplicationInstance appInstance, IHTTPStreamerSession httpStreamerSession)
void	notifyHTTPSessionDestroy(IApplicationInstance appInstance, IHTTPStreamerSession httpStreamerSession)
void	setLivePlaylist(SmoothStreamingLivePlaylist livePlaylist)
void	setPublishData(Object publishData)
void	shutdown()
void	updateLoggingValues()

Methods inherited from class com.wowza.wms.httpstreamer.model.HTTPStreamerSessionBase

```

acceptSession, addHTTPSessionToTCPSession, addIOPerformance, addIOPerformance2,
addStreamDomainStr, addStreamDomainStrs, addUserHTTPHeaders, checkAndSetPlayLogged,
clearLoggingValues, containsStreamDomainStr, containsStreamNameParts,
doSessionRedirect, doSessionRedirect, extractHTTPRequestInfo,
getAndClearNotifyCreate, getAppInstance, getConnectionHolder, getCookieStr,
getDvrSessionInfo, getElapsedTime, getFileInfo, getHTTPDate, getHTTPHeader,
getHTTPHeaderMap, getHTTPHeaderNames, getHTTPIntHeader, getHTTPStreamerAdapter,
getIOPerformanceCounter, getIpAddress, getLastRequest, getLiveStreamingPacketizer,
getLock, getPlayDuration, getPlaySeek, getPlayStart, getProperties, getQueryStr,
getRedirectSessionBody, getRedirectSessionCode, getRedirectSessionContentType,
getRedirectSessionURL, getReferrer, getServerIp, getServerPort, getSessionId,
getSessionProtocol, getSessionTimeout, getSessionType, getStream, getStreamDomainStr,
getStreamDomainStrList, getStreamExt, getStreamName, getStreamNameParts,
getStreamPosition, getTimeRunning, getTimeRunningSeconds, getTranscoderVODIndex,
getTranscoderVODIndex, getTranscoderVODIndexDestinationsMap,
getTranscoderVODIndexDestinationsOrder, getTranscoderVODSession, getUri,
getUserAgent, getUserHTTPHeaders, getUserQueryStr, getVHost, getVODTranscodeNGRP,
indexTranscoderVOD, isAcceptSession, isActive, isFileInfo, isHTTPOrigin,
isPlayLogged, isRedirectSession, isSecure, isTimeout, isTimeoutSession, isValidated,
isValidStreamDomainStr, lockRepeaterStreams, putFileInfo, putStreamNameParts,
redirectSession, redirectSession, rejectSession, removeStreamDomainStr,
setAcceptSession, setActive, setAppInstance, setCookieStr, setDvrSessionInfo,
setHTTPOrigin, setHTTPStreamerAdapter, setIpAddress, setLiveStreamingPacketizer,
setPlayDuration, setPlayLogged, setPlaySeek, setPlayStart, setQueryStr,
setRedirectSession, setRedirectSessionBody, setRedirectSessionCode,
setRedirectSessionContentType, setRedirectSessionURL, setReferrer, setSecure,
setServerIp, setServerPort, setSessionId, setSessionProtocol, setSessionTimeout,
setSessionType, setStream, setStreamExt, setStreamName, setStreamPosition,
setThreadContext, setTimeoutSession, setTranscoderVODIndex, setTranscoderVODSession,
setUri, setUserAgent, setUserHTTPHeader, setUserQueryStr, setValidated, setVHost,
setVODTranscodeNGRP, shutdown, shutdownLocked, touch, updateLoggingValues,
validStreamDomainToString

```

Methods inherited from class java.lang.Object

```

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait,
wait

```

Methods inherited from interface [com.wowza.wms.httpstreamer.model.IHTTPStreamerSession](#)

[acceptSession](#), [addHTTPSessionToTCPSession](#), [addIOPerformance](#), [addIOPerformance2](#),
[addStreamDomainStr](#), [addStreamDomainStrs](#), [addUserHTTPHeaders](#), [checkAndSetPlayLogged](#),
[clearLoggingValues](#), [containsStreamDomainStr](#), [containsStreamNameParts](#),
[doSessionRedirect](#), [doSessionRedirect](#), [extractHTTPRequestInfo](#),
[getAndClearNotifyCreate](#), [getAppInstance](#), [getConnectionHolder](#), [getCookieStr](#),
[getDvrSessionInfo](#), [getElapsedTime](#), [getFileInfo](#), [getHTTPHeader](#), [getHTTPHeaderMap](#),
[getHTTPHeaderNames](#), [getHTTPIntHeader](#), [getHTTPStreamerAdapter](#),
[getIOPerformanceCounter](#), [getIpAddress](#), [getLastRequest](#), [getLiveStreamingPacketizer](#),
[getLock](#), [getPlayDuration](#), [getPlaySeek](#), [getPlayStart](#), [getProperties](#), [getQueryStr](#),
[getRedirectSessionBody](#), [getRedirectSessionCode](#), [getRedirectSessionContentType](#),
[getRedirectSessionURL](#), [getReferrer](#), [getServerIp](#), [getServerPort](#), [getSessionId](#),
[getSessionProtocol](#), [getSessionTimeout](#), [getSessionType](#), [getStream](#), [getStreamExt](#),
[getStreamName](#), [getStreamNameParts](#), [getStreamPosition](#), [getTimeRunning](#),
[getTimeRunningSeconds](#), [getTranscoderVODIndex](#), [getTranscoderVODIndex](#),
[getTranscoderVODIndexDestinationsMap](#), [getTranscoderVODIndexDestinationsOrder](#),
[getTranscoderVODSession](#), [getUri](#), [getUserAgent](#), [getUserHTTPHeaders](#), [getUserQueryStr](#),
[getVHost](#), [getVODTranscodeNGRP](#), [indexTranscoderVOD](#), [isAcceptSession](#), [isActive](#),
[isFileInfo](#), [isHTTPOrigin](#), [isPlayLogged](#), [isRedirectSession](#), [isSecure](#), [isTimeout](#),
[isTimeoutSession](#), [isValidated](#), [isValidStreamDomainStr](#), [lockRepeaterStreams](#),
[putFileInfo](#), [putStreamNameParts](#), [redirectSession](#), [redirectSession](#), [rejectSession](#),
[removeStreamDomainStr](#), [setAcceptSession](#), [setActive](#), [setAppInstance](#), [setCookieStr](#),
[setDvrSessionInfo](#), [setHTTPOrigin](#), [setHTTPStreamerAdapter](#), [setIpAddress](#),
[setLiveStreamingPacketizer](#), [setPlayDuration](#), [setPlayLogged](#), [setPlaySeek](#),
[setPlayStart](#), [setQueryStr](#), [setRedirectSession](#), [setRedirectSessionBody](#),
[setRedirectSessionCode](#), [setRedirectSessionContentType](#), [setRedirectSessionURL](#),
[setReferrer](#), [setSecure](#), [setServerIp](#), [setServerPort](#), [setSessionId](#), [setSessionProtocol](#),
[setSessionTimeout](#), [setSessionType](#), [setStream](#), [setStreamExt](#), [setStreamName](#),
[setStreamPosition](#), [setTimeoutSession](#), [setTranscoderVODIndex](#), [setTranscoderVODSession](#),
[setUri](#), [setUserAgent](#), [setUserHTTPHeader](#), [setUserQueryStr](#), [setVHost](#),
[setVODTranscodeNGRP](#), [shutdown](#), [touch](#), [updateLoggingValues](#), [validStreamDomainToString](#)

Methods inherited from interface [com.wowza.wms.httpstreamer.model.IHTTPSessionNotify](#)

[notifyHTTPSessionCreate](#), [notifyHTTPSessionDestroy](#)

Constructors

HTTPStreamerSessionSmoothStreamer

```
public HTTPStreamerSessionSmoothStreamer()
```

Methods

shutdown

```
public void shutdown()
```

containsIndex

```
public boolean containsIndex(String streamName)
```

(continued from last page)

getIndex

```
public IHTTPStreamerSmoothStreamerIndex getIndex(IHTTPStreamerSession
httpStreamerSession,
IHTTPStreamerApplicationContext appContext,
String streamExt,
String streamName,
long playStart,
long playDuration,
long playSeek,
String vodTranscodeNGRP)
```

updateLoggingValues

```
public void updateLoggingValues()
```

clearLoggingValues

```
public void clearLoggingValues()
```

getLivePlaylist

```
public SmoothStreamingLivePlaylist getLivePlaylist()
```

setLivePlaylist

```
public void setLivePlaylist(SmoothStreamingLivePlaylist livePlaylist)
```

logLiveFragment

```
public void logLiveFragment(SmoothStreamerFragmentId fragmentId,
PacketFragmentList fragmentData)
```

logVODFragment

```
public void logVODFragment(SmoothStreamerFragmentId fragmentId,
PacketFragmentList fragmentData)
```

notifyHTTPSessionCreate

```
public void notifyHTTPSessionCreate(IApplicationInstance appInstance,
IHTTPStreamerSession httpStreamerSession)
```

notifyHTTPSessionDestroy

```
public void notifyHTTPSessionDestroy(IApplicationInstance appInstance,  
    IHTTPStreamerSession httpStreamerSession)
```

indexTranscoderVOD

```
public void indexTranscoderVOD(IHTTPStreamerApplicationContext appContext,  
    String vodTranscodeNGRP)
```

getPublishData

```
public Object getPublishData()
```

setPublishData

```
public void setPublishData(Object publishData)
```

Package

**com.wowza.wms.httpstreamer.webmstreami
ng.httpstreamer**

com.wowza.wms.httpstreamer.webmstreaming.httpstreamer Class HTTPStreamerSessionWebM

java.lang.Object

└-com.wowza.wms.httpstreamer.model.HTTPStreamerSessionBase

└-com.wowza.wms.httpstreamer.webmstreaming.httpstreamer.HTTPStreamerSessionWebM

All Implemented Interfaces:

[IHTTPStreamerSession](#)

```
public class HTTPStreamerSessionWebM
extends HTTPStreamerSessionBase
```

Fields inherited from class com.wowza.wms.httpstreamer.model.HTTPStreamerSessionBase

acceptSession, appInstance, connectionHolder, cookieStr, DATEFORMAT, elapsedTime, fastDateFormat, fileInfoMap, firstCheck, httpHeaders, httpStreamerAdapter, ioPerformanceCounter, ipAddress, isActive, isHTTPOrigin, isPlayLogged, isSecure, lastRequest, liveStreamingPacketizer, lock, mediaCasterStreamLock, notifyCreate, playDuration, playSeek, playStart, properties, queryStr, redirectSession, redirectSessionBody, redirectSessionCode, redirectSessionContentType, redirectSessionURL, referrer, serverIp, serverPort, sessionId, sessionProtocol, sessionTimeout, sessionType, stream, streamDomainStrSet, streamExt, streamName, streamNamePartMap, streamPosition, timeoutSession, totalIOPerformance2Last, totalIOPerformanceLast, transcoderVODIndex, transcoderVODIndexDestinationsMap, transcoderVODIndexDestinationsOrder, transcoderVODSession, uri, userAgent, userHTTPHeaders, userQueryStr, vhost, vodTranscodeNGRP

Fields inherited from interface [com.wowza.wms.httpstreamer.model.IHTTPStreamerSession](#)

[SESSIONPROTOCOL_COUNT](#), [SESSIONPROTOCOL_CUPERTINOSTREAMING](#), [SESSIONPROTOCOL_DIRECTSTREAMING](#), [SESSIONPROTOCOL_DVRCHUNKSTREAMING](#), [SESSIONPROTOCOL_MPEGDASHSTREAMING](#), [SESSIONPROTOCOL_SANJOSESTREAMING](#), [SESSIONPROTOCOL_SMOOTHSTREAMING](#), [SESSIONPROTOCOL_UNKNOWN](#), [SESSIONPROTOCOL_WEBMSTREAMING](#), [SESSIONTYPE_LIVE](#), [SESSIONTYPE_LIVEDVR](#), [SESSIONTYPE_UNKNOWN](#), [SESSIONTYPE_VOD](#)

Constructor Summary

public	HTTPStreamerSessionWebM()
--------	---

Method Summary

WebMPacketizerSessionTracker	getSessionTracker()
void	setSessionTracker (WebMPacketizerSessionTracker sessionTracker)
void	shutdown()

Methods inherited from class com.wowza.wms.httpstreamer.model.HTTPStreamerSessionBase


```

acceptSession, addHTTPSessionToTCPSession, addIOPerformance, addIOPerformance2,
addStreamDomainStr, addStreamDomainStrs, addUserHTTPHeaders, checkAndSetPlayLogged,
clearLoggingValues, containsStreamDomainStr, containsStreamNameParts,
doSessionRedirect, doSessionRedirect, extractHTTPRequestInfo,
getAndClearNotifyCreate, getAppInstance, getConnectionHolder, getCookieStr,
getDvrSessionInfo, getElapsedTime, getFileInfo, getHTTPDate, getHTTPHeader,
getHTTPHeaderMap, getHTTPHeaderNames, getHTTPIntHeader, getHTTPStreamerAdapter,
getIOPerformanceCounter, getIpAddress, getLastRequest, getLiveStreamingPacketizer,
getLock, getPlayDuration, getPlaySeek, getPlayStart, getProperties, getQueryStr,
getRedirectSessionBody, getRedirectSessionCode, getRedirectSessionContentType,
getRedirectSessionURL, getReferrer, getServerIp, getServerPort, getSessionId,
getSessionProtocol, getSessionTimeout, getSessionType, getStream, getStreamDomainStr,
getStreamDomainStrList, getStreamExt, getStreamName, getStreamNameParts,
getStreamPosition, getTimeRunning, getTimeRunningSeconds, getTranscoderVODIndex,
getTranscoderVODIndex, getTranscoderVODIndexDestinationsMap,
getTranscoderVODIndexDestinationsOrder, getTranscoderVODSession, getUri,
getUserAgent, getUserHTTPHeaders, getUserQueryStr, getVHost, getVODTranscodeNGRP,
indexTranscoderVOD, isAcceptSession, isActive, isFileInfo, isHTTPOrigin,
isPlayLogged, isRedirectSession, isSecure, isTimeout, isTimeoutSession, isValidated,
isValidStreamDomainStr, lockRepeaterStreams, putFileInfo, putStreamNameParts,
redirectSession, redirectSession, rejectSession, removeStreamDomainStr,
setAcceptSession, setActive, setAppInstance, setCookieStr, setDvrSessionInfo,
setHTTPOrigin, setHTTPStreamerAdapter, setIpAddress, setLiveStreamingPacketizer,
setPlayDuration, setPlayLogged, setPlaySeek, setPlayStart, setQueryStr,
setRedirectSession, setRedirectSessionBody, setRedirectSessionCode,
setRedirectSessionContentType, setRedirectSessionURL, setReferrer, setSecure,
setServerIp, setServerPort, setSessionId, setSessionProtocol, setSessionTimeout,
setSessionType, setStream, setStreamExt, setStreamName, setStreamPosition,
setThreadContext, setTimeoutSession, setTranscoderVODIndex, setTranscoderVODSession,
setUri, setUserAgent, setUserHTTPHeader, setUserQueryStr, setValidated, setVHost,
setVODTranscodeNGRP, shutdown, shutdownLocked, touch, updateLoggingValues,
validStreamDomainToString

```

Methods inherited from class java.lang.Object

```

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait,
wait

```

Methods inherited from interface [com.wowza.wms.httpstreamer.model.IHTTPStreamerSession](#)

[acceptSession](#), [addHTTPSessionToTCPSession](#), [addIOPerformance](#), [addIOPerformance2](#),
[addStreamDomainStr](#), [addStreamDomainStrs](#), [addUserHTTPHeaders](#), [checkAndSetPlayLogged](#),
[clearLoggingValues](#), [containsStreamDomainStr](#), [containsStreamNameParts](#),
[doSessionRedirect](#), [doSessionRedirect](#), [extractHTTPRequestInfo](#),
[getAndClearNotifyCreate](#), [getAppInstance](#), [getConnectionHolder](#), [getCookieStr](#),
[getDvrSessionInfo](#), [getElapsedTime](#), [getFileInfo](#), [getHTTPHeader](#), [getHTTPHeaderMap](#),
[getHTTPHeaderNames](#), [getHTTPIntHeader](#), [getHTTPStreamerAdapter](#),
[getIOPerformanceCounter](#), [getIpAddress](#), [getLastRequest](#), [getLiveStreamingPacketizer](#),
[getLock](#), [getPlayDuration](#), [getPlaySeek](#), [getPlayStart](#), [getProperties](#), [getQueryStr](#),
[getRedirectSessionBody](#), [getRedirectSessionCode](#), [getRedirectSessionContentType](#),
[getRedirectSessionURL](#), [getReferrer](#), [getServerIp](#), [getServerPort](#), [getSessionId](#),
[getSessionProtocol](#), [getSessionTimeout](#), [getSessionType](#), [getStream](#), [getStreamExt](#),
[getStreamName](#), [getStreamNameParts](#), [getStreamPosition](#), [getTimeRunning](#),
[getTimeRunningSeconds](#), [getTranscoderVODIndex](#), [getTranscoderVODIndex](#),
[getTranscoderVODIndexDestinationsMap](#), [getTranscoderVODIndexDestinationsOrder](#),
[getTranscoderVODSession](#), [getUri](#), [getUserAgent](#), [getUserHTTPHeaders](#), [getUserQueryStr](#),
[getVHost](#), [getVODTranscodeNGRP](#), [indexTranscoderVOD](#), [isAcceptSession](#), [isActive](#),
[isFileInfo](#), [isHTTPOrigin](#), [isPlayLogged](#), [isRedirectSession](#), [isSecure](#), [isTimeout](#),
[isTimeoutSession](#), [isValidated](#), [isValidStreamDomainStr](#), [lockRepeaterStreams](#),
[putFileInfo](#), [putStreamNameParts](#), [redirectSession](#), [redirectSession](#), [rejectSession](#),
[removeStreamDomainStr](#), [setAcceptSession](#), [setActive](#), [setAppInstance](#), [setCookieStr](#),
[setDvrSessionInfo](#), [setHTTPOrigin](#), [setHTTPStreamerAdapter](#), [setIpAddress](#),
[setLiveStreamingPacketizer](#), [setPlayDuration](#), [setPlayLogged](#), [setPlaySeek](#),
[setPlayStart](#), [setQueryStr](#), [setRedirectSession](#), [setRedirectSessionBody](#),
[setRedirectSessionCode](#), [setRedirectSessionContentType](#), [setRedirectSessionURL](#),
[setReferrer](#), [setSecure](#), [setServerIp](#), [setServerPort](#), [setSessionId](#), [setSessionProtocol](#),
[setSessionTimeout](#), [setSessionType](#), [setStream](#), [setStreamExt](#), [setStreamName](#),
[setStreamPosition](#), [setTimeoutSession](#), [setTranscoderVODIndex](#), [setTranscoderVODSession](#),
[setUri](#), [setUserAgent](#), [setUserHTTPHeader](#), [setUserQueryStr](#), [setVHost](#),
[setVODTranscodeNGRP](#), [shutdown](#), [touch](#), [updateLoggingValues](#), [validStreamDomainToString](#)

Constructors

HTTPStreamerSessionWebM

```
public HTTPStreamerSessionWebM()
```

Methods

getSessionTracker

```
public WebMPacketizerSessionTracker getSessionTracker()
```

setSessionTracker

```
public void setSessionTracker(WebMPacketizerSessionTracker sessionTracker)
```

(continued from last page)

shutdown

```
public void shutdown()
```

Package

com.wowza.wms.livestreamrecord.manager

com.wowza.wms.livestreamrecord.manager

Interface `ILiveStreamRecordManager`

public interface `ILiveStreamRecordManager`
 extends

This interface provides access to the functionality of the `LiveStreamRecordManager`. The `LiveStreamRecordManager` is responsible for maintaining the list of `IStreamRecorders` as well as managing the `IStreamRecorder` life cycle.

Method Summary

<code>ILiveStreamRecordManagerActionNotify</code>	<code>addListener(ILiveStreamRecordManagerActionNotify listener)</code> Used to add a listener to receive <code>ILiveStreamRecordManagerActionNotify</code> events from the <code>LiveStreamRecordManager</code> .
<code>IStreamRecorder</code>	<code>getRecorder(IApplicationInstance appInstance, String streamName)</code> Used to get a specific <code>IStreamRecorder</code> from the <code>LiveStreamRecordManager</code> . Modifying the returned <code>IStreamRecorder</code> while a recording is active may result in undefined behavior.
<code>java.util.List</code>	<code>getRecordersList(IApplicationInstance appInstance)</code> Used to get the list of <code>IStreamRecorders</code> for a specific application instance.
<code>java.util.Map</code>	<code>getRecordersMap(IApplicationInstance appInstance)</code> Used to get the Map of stream Name to <code>IStreamRecorder</code> for a specific application instance.
<code>void</code>	<code>init(IVHost vhost)</code> Called by vhost init to initialize the <code>LiveStreamRecordManager</code> .
<code>boolean</code>	<code>removeListener(ILiveStreamRecordManagerActionNotify listener)</code> Used to remove an <code>ILiveStreamRecordManagerActionNotify</code> listener from the <code>LiveStreamRecordManager</code> .
<code>void</code>	<code>splitRecording(IApplicationInstance appInstance, String streamName)</code> Used to tell the <code>LiveStreamRecordManager</code> to split (a.k.a segment) a recording immediately for a specific stream in a specific application instance.
<code>void</code>	<code>startRecording(IApplicationInstance appInstance, StreamRecorderParameters params)</code> Used to tell the <code>LiveStreamRecordManager</code> to create <code>StreamRecorders</code> for all streams which are published to the specified application instance.
<code>void</code>	<code>startRecording(IApplicationInstance appInstance, String streamName, StreamRecorderParameters params)</code> Used to tell the <code>LiveStreamRecordManager</code> to create a <code>StreamRecorder</code> for a specific stream in a specific application instance.
<code>void</code>	<code>stopRecording(IApplicationInstance appInstance)</code> Used to tell the <code>LiveStreamRecordManager</code> to stop all <code>StreamRecorders</code> on a specific application instance.
<code>void</code>	<code>stopRecording(IApplicationInstance appInstance, String streamName)</code> Used to tell the <code>LiveStreamRecordManager</code> to stop a <code>StreamRecorder</code> for a specific stream in a specific application instance.

Methods

(continued from last page)

init

```
public void init(IVHost vhost)
```

Called by vhost init to initialize the LiveStreamRecordManager

Parameters:

vhost - The vhost to associated with the instance of LiveStreamRecordManager

getRecorder

```
public IStreamRecorder getRecorder(IApplicationInstance appInstance,  
    String streamName)
```

Used to get a specific IStreamRecorder from the LiveStreamRecordManager. Modifying the returned IStreamRecorder while a recording is active may result in undefined behavior.

Parameters:

appInstance - The application instance associated with the desired IStreamRecorder

streamName - The stream name associated with the desired IStreamRecorder

Returns:

The requested IStreamRecorder or null if it does not exist

getRecordersList

```
public java.util.List getRecordersList(IApplicationInstance appInstance)
```

Used to get the list of IStreamRecorders for a specific application instance. Making changes to the returned list (e.g. removing an IStreamRecorder from the list) does not affect the LiveStreamRecordManager's list, however, changes made to the IStreamRecorders themselves will be reflected in the LiveStreamRecordManager's IStreamRecorder. Therefore, changing IStreamRecorders while a recording is active may result in undefined behavior.

Parameters:

appInstance - The application instance associated with the desired IStreamRecorder List

Returns:

The requested IStreamRecorder List. The list can be empty if no IStreamRecorders are set.

getRecordersMap

```
public java.util.Map getRecordersMap(IApplicationInstance appInstance)
```

Used to get the Map of stream Name to IStreamRecorder for a specific application instance. Making changes to the returned map (e.g. removing a mapping entry from the map) does not affect the LiveStreamRecordManager's map, however, changes made to the IStreamRecorders themselves will be reflected in the LiveStreamRecordManager's IStreamRecorder. Therefore, changing IStreamRecorders while a recording is active may result in undefined behavior.

Parameters:

appInstance - The application instance associated with the desired IStreamRecorder List

Returns:

The requested stream name to IStreamRecorder mappings. The map can be empty if no IStreamRecorders are set.

startRecording

```
public void startRecording(IApplicationInstance appInstance,  
    StreamRecorderParameters params)
```

(continued from last page)

Used to tell the LiveStreamRecordManager to create StreamRecorders for all streams which are published to the specified application instance. The StreamRecorderParameters specified will be used for each StreamRecorder created. If the StreamRecordParameters are null the default StreamRecordParameters will be used.

Parameters:

appInstance - The application instance on which to record all incoming streams
params - The StreamRecorderParameters to use for all recorders

See Also:

[StreamRecorderParameters](#)
[IStreamRecorderConstants](#)

startRecording

```
public void startRecording(IApplicationInstance appInstance,  
    String streamName,  
    StreamRecorderParameters params)
```

Used to tell the LiveStreamRecordManager to create a StreamRecorder for a specific stream in a specific application instance. The StreamRecorderParameters specified will be used for the StreamRecorder. If the StreamRecordParameters are null the default StreamRecordParameters will be used.

Parameters:

appInstance - The application instance on which to record the requested stream
streamName - The stream name associated with the incoming stream to record
params - The StreamRecorderParameters to use for the StreamRecorder

See Also:

[StreamRecorderParameters](#)
[IStreamRecorderConstants](#)

stopRecording

```
public void stopRecording(IApplicationInstance appInstance,  
    String streamName)
```

Used to tell the LiveStreamRecordManager to stop a StreamRecorder for a specific stream in a specific application instance. If a StreamRecorder already exists for the specified application instance and stream name, nothing is done.

Parameters:

appInstance - The application instance of the StreamRecorder to stop
streamName - The stream name associated with the StreamRecorder to stop

stopRecording

```
public void stopRecording(IApplicationInstance appInstance)
```

Used to tell the LiveStreamRecordManager to stop all StreamRecorders on a specific application instance. If no StreamRecorders exist for the specified application instance, nothing is done.

Parameters:

appInstance - The application instance on which to stop all StreamRecorders

splitRecording

```
public void splitRecording(IApplicationInstance appInstance,  
    String streamName)
```

Used to tell the LiveStreamRecordManager to split (a.k.a segment) a recording immediately for a specific stream in a specific application instance. If no StreamRecorder exists for the specified application instance and stream name, nothing is done.

(continued from last page)

Parameters:

appInstance - The application instance of the StreamRecorder to split
streamName - The stream name associated with the StreamRecorder to split

addListener

```
public ILiveStreamRecordManagerActionNotify  
addListener(ILiveStreamRecordManagerActionNotify listener)
```

Used to add a listener to receive [ILiveStreamRecordManagerActionNotify](#) events from the [LiveStreamRecordManager](#). Listeners are notified in the order in which they are added.

Parameters:

listener - An instance of a class which implements the [ILiveStreamRecordManagerActionNotify](#) interface

Returns:

The listener instance passed in

See Also:

[ILiveStreamRecordManagerActionNotify](#)

removeListener

```
public boolean removeListener(ILiveStreamRecordManagerActionNotify listener)
```

Used to remove an [ILiveStreamRecordManagerActionNotify](#) listener from the [LiveStreamRecordManager](#).

Parameters:

listener - The listener passed into the [addListener](#) method.

Returns:

true if the listener was found in the list otherwise false.

com.wowza.wms.livestreamrecord.manager Interface **ILiveStreamRecordManagerActionNotify**

public interface **ILiveStreamRecordManagerActionNotify**
extends

This interface provides notifications to registered listeners for LiveStreamRecordManager events as it operates on IStreamRecorders. These notification are sent prior to invoking the associated method for a recorder.

Multiple Listener's can be registered for the LiveStreamRecordManager. The listeners are notified in the order in which they were added.

See Also:
[for API's to add and remove listeners](#)

Method Summary

void	onCreateRecord (IStreamRecorder recorder) This notification occurs when a new recorder is created due to a call to one of the ILiveStreamRecordManager startRecording API's.
void	onSplitRecord (IStreamRecorder recorder) This notification occurs due to a call to the ILiveStreamRecordManager splitRecording API.
void	onStartRecord (IStreamRecorder recorder) This notification occurs due to a call to one of the ILiveStreamRecordManager startRecording API's.
void	onStopRecord (IStreamRecorder recorder) This notification occurs due to a call to the ILiveStreamRecordManager stopRecording API.
void	onSwitchRecord (IStreamRecorder recorder, IMediaStream newStream) This notification occurs when an existing stream is replaced with a stream of the same name.
IStreamRecorder	recordFactory (String streamName, StreamRecorderParameters recordParams) This notification is the first one to occur when the ILiveStreamRecordManager is requested to start a recording via any of the ILiveStreamRecordManager startRecording API's It gives the listener the opportunity to return a custom class which implements the IStreamRecorder interface to handle the recording. If the listener returns null, the LiveStreamRecordManager will internally create a new StreamRecorder to handle the recording.

Methods

recordFactory

```
public IStreamRecorder recordFactory(String streamName,  
    StreamRecorderParameters recordParams)
```

(continued from last page)

This notification is the first one to occur when the `ILiveStreamRecordManager` is requested to start a recording via any of the `ILiveStreamRecordManager` `startRecording` API's

It gives the listener the opportunity to return a custom class which implements the `IStreamRecorder` interface to handle the recording.

If the listener returns null, the `LiveStreamRecordManager` will internally create a new `StreamRecorder` to handle the recording. Since multiple listeners can be registered, `LiveStreamRecordManager` notifies listeners in the order in which they were added, and will use the first `IStreamRecorder` returned which is not null, however, remaining listeners will receive the notification.

Called before `LiveStreamRecordManager` internal action is taken

Parameters:

`recordParams` - the parameters passed into `IStreamRecordManager` `startRecording` API

`streamName` - the stream name passed into `IStreamRecordManager` `startRecording` API

Returns:

the `IStreamRecorder` which will handled recording or null

onCreateRecord

```
public void onCreateRecord(IStreamRecorder recorder)
```

This notification occurs when a new recorder is created due to a call to one of the `ILiveStreamRecordManager` `startRecording` API's.

Called after the `recordFactory` notification is called, but before `LiveStreamRecordManager` calls the recorders own `createRecorder` method.

Parameters:

`recorder` - The new recorder

onStartRecord

```
public void onStartRecord(IStreamRecorder recorder)
```

This notification occurs due to a call to one of the `ILiveStreamRecordManager` `startRecording` API's.

Called after the `onCreateRecord` notification is called, but before `LiveStreamRecordManager` calls the recorders own `startRecorder` method.

Parameters:

`recorder` - The recorder which is starting

onSplitRecord

```
public void onSplitRecord(IStreamRecorder recorder)
```

This notification occurs due to a call to the `ILiveStreamRecordManager` `splitRecording` API.

Called before `LiveStreamRecordManager` calls the recorders own `splitRecorder` method.

Parameters:

`recorder` - The recorder for which the split has been requested.

onStopRecord

```
public void onStopRecord(IStreamRecorder recorder)
```

This notification occurs due to a call to the `ILiveStreamRecordManager` `stopRecording` API.

Called before `LiveStreamRecordManager` calls the recorders own `stopRecorder` method.

Parameters:

`recorder` - The recorder for which a stop has been requested.

onSwitchRecord

```
public void onSwitchRecord(IStreamRecorder recorder,  
    IMediaStream newStream)
```

This notification occurs when an existing stream is replaced with a stream of the same name. This can occur when a stream is published from an encoder, and then a different encoder publishes a stream using the same stream name.

Called before `LiveStreamRecordManager` calls the recorders own `switchRecorder` method.

Parameters:

`recorder` - The recorder for the existing stream

`newStream` - The new `IMediaStream` associated with the existing stream name

com.wowza.wms.livestreamrecord.manager

Interface IStreamRecorder

public interface **IStreamRecorder**
extends

This interface provides access to into a StreamRecorder.

The StreamRecorder is responsible for performing the actual recording functionality of a live stream, as directed by the LiveStreamRecordManager.

Method Summary

IStreamRecorderActionNotify	addListener(IStreamRecorderActionNotify listener) Called to add a listener for IStreamRecorderActionNotify events generated by the recorder. Multiple listeners can be added.
void	createRecorder() Called to tell the recorder to initialize itself after it has been created.
IApplicationInstance	getAppInstance() Called to get the application Instance the recorder is associated with
String	getBaseFilePath() Called to get the full path and filename used by the recorder as the PRIOR to any changes made due to the selected segmentationType and versioningOption.
long	getCurrentDuration() Called to get the current segment size regardless of segmentationType. This value is updated continuously while the recorder state is RECORDER_STATE_RECORDING.
String	getCurrentFile() Called to get the current segment size regardless of segmentationType. This value is updated continuously while the recorder state is RECORDER_STATE_RECORDING.
long	getCurrentSize() Called to get the current segment size regardless of segmentationType This value is updated continuously while the recorder state is RECORDER_STATE_RECORDING.
org.joda.time.DateTime	getEndTime() Called to get the end time of the segment regardless of segmentationType. When the recording is still active, the returned value is for the previous segment. When the recording has been stopped, the returned value is for the last segment.
String	getErrorString() Called to get an error string describing the error, when the recorder state is RECORDER_STATE_ERROR.
String	getFileFormat() Called to get output file format being used by the recorder.
String	getFilePath() Called to get the full path and filename used by the recorder AFTER changes made due to the selected segmentationType and versioningOption. NOTE: When the recorder is not in RECORDER_STATE_RECORDING state, the returned value will be the basefilePath value.
IStreamRecorderFileVersionDelegate	getFileVersionDelegate() Called to get the IStreamRecorderFileVersionDelegate being used by the recorder.

<u>StreamRecorderParameters</u>	<u>getRecorderParams()</u> Called to get the StreamRecorderParameters for the recorder This is typically used to check or set the StreamRecorderParameters used to configure a recorder.
int	<u>getRecorderState()</u> Called to get the current state of the recorder
String	<u>getRecorderStateString()</u> Called to get the current state of the recorder as a text string.
String	<u>getSegmentationType()</u> Called to get the SegmentationType being used by the recorder.
long	<u>getSegmentDuration()</u> Called to get the segment duration used by the recorder when segmentationType is set to SEGMENT_BY_DURATION.
int	<u>getSegmentNumber()</u> Called to get the current segment number regardless of segmentationType. This value is updated continuously while the recorder state is RECORDER_STATE_RECORDING.
<u>CrontabEvent</u>	<u>getSegmentSchedule()</u> Called to get the CrontabEvent object associated with the recorder when the segmentationType is set to SEGMENT_BY_SCHEDULE When the recorder state is RECORDER_STATE_ERROR, RECORDER_STATE_PENDING or RECORDER_STATE_WAITING, the return value is null, otherwise the CrontabEvent use by the recorder is returned.
String	<u>getSegmentScheduleString()</u> Called to get the segment schedule crontab string used by the recorder when segmentationType is set to SEGMENT_BY_SCHEDULE.
long	<u>getSegmentSize()</u> Called to get the segment size used by the recorder when segmentationType is set to SEGMENT_BY_SIZE.
org.joda.time.DateTime	<u>getStartTime()</u> Called to get the start time of the current segment regardless of segmentationType.
<u>IMediaStream</u>	<u>getStream()</u> Called to get the IMediaStream object being used by the recorder
String	<u>getStreamName()</u> Called to get the stream name being used by the recorder
String	<u>getVersioningOption()</u> Called to get the method used by the recorder to version a file when a file exists of the same name.
boolean	<u>isAppendFile()</u> Called to check the versioningOption used by the recorder. Users should use getVersioningOption instead.
boolean	<u>isMoveFirstVideoFrameToZero()</u> Called to check the value of MoveFirstVideoFrameToZero being used by the recorder.
boolean	<u>isRecordData()</u> Called to check the value of RecordData being used by the recorder.
boolean	<u>isSplitOnTcDiscontinuity()</u> Called to check the value of splitOnTcDiscontinuity being used by the recorder.

boolean	<u>isStartOnKeyFrame()</u> Called to check the value of StartOnKeyFrame being used by the recorder.
boolean	<u>isVersionFile()</u> Called to check the versioningOption used by the recorder. Users should use getVersioningOption instead.
boolean	<u>removeListener(IStreamRecorderActionNotify listener)</u> Called to remove a listener from the the recorder.
void	<u>setInternalRecorder(com.wowza.wms.livestreamrecord.model.ILiveStreamRecord internalRecorder)</u> Called to set the internal recorder used by the StreamRecorder This method should only be used if existing recorder workflows cannot be implemented using the IStreamRecorderActionNotify API's.
void	<u>setStream(IMediaStream stream)</u> Called to set the IMediaStream object the recorder will use NOTE: This method is used by LiveStreamRecordManager to manage recorders and associated IMediaStream objects and should NOT be used directly when a StreamRecorder is under the control of LiveStreamRecordManager
void	<u>splitRecorder()</u> Called to tell the recorder to stop writing to the current file, create a new file and write begin writing data to the new file. The recorder should start writing data to the new file beginning with a video key frame.
void	<u>startRecorder()</u> Called to tell the recorder to start recording.
void	<u>stopRecorder()</u> Called to tell the recorder to stop recording.
void	<u>switchRecorder(IMediaStream newStream)</u> Called to tell the recorder to switch over to the data from a new IMediaStream. This occurs when an existing stream is replaced with a stream of the same name.

Methods

createRecorder

```
public void createRecorder()
```

Called to tell the recorder to initialize itself after it has been created. This API should be called prior to calling the startRecorder() method.

startRecorder

```
public void startRecorder()
```

Called to tell the recorder to start recording. This API should be called after createRecorder() method has been called to ensure the recorder has been properly initialized

switchRecorder

```
public void switchRecorder(IMediaStream newStream)
```

(continued from last page)

Called to tell the recorder to switch over to the data from a new IMediaStream.
This occurs when an existing stream is replaced with a stream of the same name.

For example: When a stream is published from an encoder, and then a different encoder publishes a stream using the same stream name. Called before LiveStreamRecordManager calls the recorders own switchRecorder method.

Parameters:

newStream - The new IMediaStream the recorder should switch to.

stopRecorder

```
public void stopRecorder()
```

Called to tell the recorder to stop recording.

splitRecorder

```
public void splitRecorder()
```

Called to tell the recorder to stop writing to the current file, create a new file and write begin writing data to the new file.
The recorder should start writing data to the new file beginning with a video key frame.

getRecorderParams

```
public StreamRecorderParameters getRecorderParams()
```

Called to get the StreamRecorderParameters for the recorder
This is typically used to check or set the StreamRecorderParameters used to configure a recorder. To get the actual value used by the recorder, use the appropriate get method found below.

See Also:

[StreamRecorderParameters](#)

getAppInstance

```
public IApplicationInstance getAppInstance()
```

Called to get the application Instance the recorder is associated with

Returns:

IApplicationInstance

getStreamName

```
public String getStreamName()
```

Called to get the stream name being used by the recorder

Returns:

String Stream name

getStream

```
public IMediaStream getStream( )
```

Called to get the IMediaStream object being used by the recorder

Returns:

IMediaStream Stream object

setStream

```
public void setStream(IMediaStream stream)
```

Called to set the IMediaStream object the recorder will use

NOTE: This method is used by LiveStreamRecordManager to manage recorders and associated IMediaStream objects and should NOT be used directly when a StreamRecorder is under the control of LiveStreamRecordManager

Parameters:

stream - IMediaStream The new Stream object

getRecorderState

```
public int getRecorderState( )
```

Called to get the current state of the recorder

Returns:

int Current state

See Also:

[IStreamRecorderConstants.RECORDER_STATE_ERROR](#)
[IStreamRecorderConstants.RECORDER_STATE_PENDING](#)
[IStreamRecorderConstants.RECORDER_STATE_WAITING](#)
[IStreamRecorderConstants.RECORDER_STATE_RECORDING](#)

getRecorderStateString

```
public String getRecorderStateString( )
```

Called to get the current state of the recorder as a text string.

Returns:

String Current state as a string

(continued from last page)

See Also:

[IStreamRecorderConstants.RECORDER_STATE_ERROR_STRING](#)
[IStreamRecorderConstants.RECORDER_STATE_PENDING_STRING](#)
[IStreamRecorderConstants.RECORDER_STATE_WAITING_STRING](#)
[IStreamRecorderConstants.RECORDER_STATE_RECORDING_STRING](#)

getFileFormat

```
public String getFileFormat()
```

Called to get output file format being used by the recorder.

Returns:

String Output file format

See Also:

[IStreamRecorderConstants.FORMAT_MP4](#)
[IStreamRecorderConstants.FORMAT_FLV](#)

isStartOnKeyFrame

```
public boolean isStartOnKeyFrame()
```

Called to check the value of StartOnKeyFrame being used by the recorder.

Returns:

boolean true, false

See Also:

[StreamRecorderParameters.startOnKeyFrame](#)

isRecordData

```
public boolean isRecordData()
```

Called to check the value of RecordData being used by the recorder.

Returns:

boolean true, false

See Also:

[StreamRecorderParameters.recordData](#)

(continued from last page)

isSplitOnTcDiscontinuity

```
public boolean isSplitOnTcDiscontinuity()
```

Called to check the value of splitOnTcDiscontinuity being used by the recorder.

Returns:

boolean true, false

See Also:

[StreamRecorderParameters.splitOnTcDiscontinuity](#)

isMoveFirstVideoFrameToZero

```
public boolean isMoveFirstVideoFrameToZero()
```

Called to check the value of MoveFirstVideoFrameToZero being used by the recorder.

Returns:

boolean true, false

See Also:

[StreamRecorderParameters.moveFirstVideoFrameToZero](#)

getSegmentationType

```
public String getSegmentationType()
```

Called to get the SegmentationType being used by the recorder.

Returns:

String Segmentation type

See Also:

[StreamRecorderParameters.segmentationType](#)
[IStreamRecorderConstants.SEGMENT_NONE](#)
[IStreamRecorderConstants.SEGMENT_BY_DURATION](#)
[IStreamRecorderConstants.SEGMENT_BY_SIZE](#)
[IStreamRecorderConstants.SEGMENT_BY_SCHEDULE](#)

getVersioningOption

```
public String getVersioningOption()
```

(continued from last page)

Called to get the method used by the recorder to version a file when a file exists of the same name.

Returns:

String Versioning option

See Also:

[StreamRecorderParameters.versioningOption](#)
[IStreamRecorderConstants.APPEND_FILE](#)
[IStreamRecorderConstants.OVERWRITE_FILE](#)
[IStreamRecorderConstants.VERSION_FILE](#)

getSegmentSize

```
public long getSegmentSize()
```

Called to get the segment size used by the recorder when segmentationType is set to SEGMENT_BY_SIZE.

Returns:

long Segment size in bytes

See Also:

[StreamRecorderParameters.segmentSize](#)

getSegmentDuration

```
public long getSegmentDuration()
```

Called to get the segment duration used by the recorder when segmentationType is set to SEGMENT_BY_DURATION.

Returns:

long Segment duration in milliseconds

See Also:

[StreamRecorderParameters.segmentDuration](#)

getSegmentScheduleString

```
public String getSegmentScheduleString()
```

Called to get the segment schedule crontab string used by the recorder when segmentationType is set to SEGMENT_BY_SCHEDULE.

Returns:

(continued from last page)

String Segment schedule

See Also:

[StreamRecorderParameters.segmentSchedule](#)

getFilePath

```
public String getFilePath()
```

Called to get the full path and filename used by the recorder AFTER changes made due to the selected segmentationType and versioningOption.

NOTE: When the recorder is not in RECORDER_STATE_RECORDING state, the returned value will be the basefilePath value.

Returns:

String Full path and filename

See Also:

[getBaseFilePath\(\)](#)

getBaseFilePath

```
public String getBaseFilePath()
```

Called to get the full path and filename used by the recorder as the PRIOR to any changes made due to the selected segmentationType and versioningOption.

Returns:

String Full path and filename

See Also:

[getFilePath\(\)](#)

getCurrentSize

```
public long getCurrentSize()
```

Called to get the current segment size regardless of segmentationType

This value is updated continuously while the recorder state is RECORDER_STATE_RECORDING.

Returns:

long Segment size in bytes

getCurrentDuration

```
public long getCurrentDuration()
```

(continued from last page)

Called to get the current segment size regardless of segmentationType.

This value is updated continuously while the recorder state is RECORDER_STATE_RECORDING.

Returns:

long Segment duration in milliseconds

getCurrentFile

```
public String getCurrentFile()
```

Called to get the current segment size regardless of segmentationType.

This value is updated continuously while the recorder state is RECORDER_STATE_RECORDING.

Returns:

String The current fully qualified file name, packets are being written to.

getSegmentNumber

```
public int getSegmentNumber()
```

Called to get the current segment number regardless of segmentationType.

This value is updated continuously while the recorder state is RECORDER_STATE_RECORDING.

Returns:

int Current segment number.

getStartTime

```
public org.joda.time.DateTime getStartTime()
```

Called to get the start time of the current segment regardless of segmentationType.

Returns:

DateTime Current segment's start time.

getEndTime

```
public org.joda.time.DateTime getEndTime()
```

Called to get the end time of the segment regardless of segmentationType.

When the recording is still active, the returned value is for the previous segment.

When the recording has been stopped, the returned value is for the last segment.

Returns:

DateTime Segment end time.

getErrorString

```
public String getErrorString()
```

(continued from last page)

Called to get an error string describing the error, when the recorder state is is RECORDER_STATE_ERROR.

Returns:

String Error description string

getFileVersionDelegate

```
public IStreamRecorderFileVersionDelegate getFileVersionDelegate()
```

Called to get the IStreamRecorderFileVersionDelegate being used by the recorder.

Returns:

IStreamRecorderFileVersionDelegate

See Also:

[IStreamRecorderFileVersionDelegate](#)

[StreamRecorderFileVersionDelegate](#)

[StreamRecorderSimpleFileVersionDelegate](#)

addListener

```
public IStreamRecorderActionNotify addListener(IStreamRecorderActionNotify listener)
```

Called to add a listener for IStreamRecorderActionNotify events generated by the recorder.
Multiple listeners can be added. All listeners will be called for each event, in the order in which they were added.

Parameters:

listener - An object which implements the IStreamRecorderActionNotify interface

Returns:

IStreamRecorderActionNotify The listener object passed in.

See Also:

[IStreamRecorderActionNotify](#)

removeListener

```
public boolean removeListener(IStreamRecorderActionNotify listener)
```

Called to remove a listener from the the recorder.

Parameters:

listener - The listener which is to be removed

Returns:

(continued from last page)

boolean true, if listener was found in list, else false

isVersionFile

```
public boolean isVersionFile()
```

Called to check the versioningOption used by the recorder.
Users should use getVersioningOption instead.

Returns:

boolean true when versioningOption is set to VERISON_FILE, otherwise false

isAppendFile

```
public boolean isAppendFile()
```

Called to check the versioningOption used by the recorder.
Users should use getVersioningOption instead.

Returns:

boolean true when versioningOption is set to APPEND_FILE, otherwise false

getSegmentSchedule

```
public CrontabEvent getSegmentSchedule()
```

Called to get the CrontabEvent object associated with the recorder when the segmentationType is set to SEGMENT_BY_SCHEDULE
When the recorder state is RECORDER_STATE_ERROR, RECORDER_STATE_PENDING or RECORDER_STATE_WAITING, the return value is null,
otherwise the CrontabEvent use by the recorder is returned.

Returns:

CrontabEvent

setInternalRecorder

```
public void setInternalRecorder(com.wowza.wms.livestreamrecord.model.ILiveStreamRecord  
internalRecorder)
```

Called to set the internal recorder used by the StreamRecorder
This method should only be used if existing recorder workflows cannot be implemented using the IStreamRecorderActionNotify API's.

Parameters:

internalRecorder

com.wowza.wms.livestreamrecord.manager Interface IStreamRecorderActionNotify

public interface **IStreamRecorderActionNotify**
extends

This interface provides notifications to registered listeners throughout the life cycle of a StreamRecorder. These events are similar to the ILiveStreamRecordManageActionNotify events, however, they are called by the StreamRecorder deeper in the action processing.

Multiple Listener's can be registered for each StreamRecorder. The listeners are notified in the order in which they were added.

See Also:
[for API's to add and remove listeners](#)

Method Summary

void	onCreateRecorder (IStreamRecorder recorder) This notification occurs when a new recorder has been created, but before recording has started.
void	onSegmentEnd (IStreamRecorder recorder) This notification occurs when a recorder stops writing to the current file
void	onSegmentStart (IStreamRecorder recorder) This notification occurs when a recorder starts writing to the current file
void	onSplitRecorder (IStreamRecorder recorder) This notification occurs when a recorder is directed to split the active recording.
void	onStartRecorder (IStreamRecorder recorder) This notification occurs when a recorder starts recording.
void	onStopRecorder (IStreamRecorder recorder) This notification occurs when a recorder stops recording.
void	onSwitchRecorder (IStreamRecorder recorder, IMediaStream newStream) This notification occurs when an existing stream has been replaced by a new stream of the same name.

Methods

onCreateRecorder

public void **onCreateRecorder**([IStreamRecorder](#) recorder)

This notification occurs when a new recorder has been created, but before recording has started. Within this method, changes can be made to the StreamRecorder's StreamRecorderParameters, which will then take affect when the recorder is started.

Called before action is taken

Parameters:

recorder - the new recorder

(continued from last page)

onStartRecorder

```
public void onStartRecorder(IStreamRecorder recorder)
```

This notification occurs when a recorder starts recording.

Called after all configuration of the recorder is complete, but before the recording actually starts.

Parameters:

recorder - the recorder which is starting

onSplitRecorder

```
public void onSplitRecorder(IStreamRecorder recorder)
```

This notification occurs when a recorder is directed to split the active recording.

Called after all configuration of the recorder is complete, but before the recording actually starts.

Parameters:

recorder - the recorder which is starting

onStopRecorder

```
public void onStopRecorder(IStreamRecorder recorder)
```

This notification occurs when a recorder stops recording.

NOTE: Recorders are stopped whenever a stream is unpublished, or when a stream is "reset".

Within this method, calling; `getCurrentFile()`, `getCurrentDuration()`, `getCurrentSize()`, `segmentNumber()`, `getStartTime()`, `getEndTime()` will return the values accumulated from the time the `onStartRecorder` notification was called, to the time this notification was called.

Called after the recorder is stopped.

Parameters:

recorder - the recorder which is starting

onSwitchRecorder

```
public void onSwitchRecorder(IStreamRecorder recorder,  
                             IMediaStream newStream)
```

This notification occurs when an existing stream has been replaced by a new stream of the same name. The `StreamRecorder` will stop the recording associated with the existing stream and start a new recording for the new stream, using the same parameters as those used to originally start the recording.

Called before any action is taken

Parameters:

recorder - the current recorder for existing stream

newStream - the new stream which will be recorded

onSegmentStart

```
public void onSegmentStart(IStreamRecorder recorder)
```

This notification occurs when a recorder starts writing to the current file

Called after a the file has been opened for writing, but before any data is written to the file.

Parameters:

(continued from last page)

recorder - the recorder which is writing the segment

onSegmentEnd

```
public void onSegmentEnd(IStreamRecorder recorder)
```

This notification occurs when a recorder stops writing to the current file

Called after all packets have been written tot he file and the file has been closed.

Parameters:

recorder - the recorder which is wrote the segment

com.wowza.wms.livestreamrecord.manager Interface IStreamRecorderConstants

public interface **IStreamRecorderConstants**
extends

Defines the static constants used by the StreamRecorder system

The Properties for StreamRecorders should be placed in application.xml within the StreamRecorder properties container

for example:

```
</StreamRecorder>
  </Properties>
    </Property>
      </Name>streamRecorderDebugEnabled</Name>
      </Value>true</Value>
      </Type>Boolean</Type>
    </Property>
  </Properties>
</StreamRecorder>
```

Field Summary

public static final	<u>APPEND_FILE</u> Appends new stream data to the existing file. Value: append
public static final	<u>FORMAT_FLV</u> Value: flv
public static final	<u>FORMAT_MP4</u> Value: mp4
public static final	<u>OVERWRITE_FILE</u> Overwrites the existing file with the new file Value: overwrite
public static final	<u>PROPERTY_DEBUG_ENABLE</u> Application Property "streamRecorderDebugEnabled": for enabling additional log statements throughout StreamRecorder Value: streamRecorderDebugEnabled
public static final	<u>PROPERTY_FILE_FORMAT</u> Application Property "streamRecorderFileFormat": for specifying the default file format used by StreamRecorder. Value: streamRecorderFileFormat
public static final	<u>PROPERTY_FILE_VERSION_DELEGATE</u> Application Property "streamRecorderFileVersionDelegate": for specifying a default custom file version delegate. Value: streamRecorderFileVersionDelegate

public static final	<u>PROPERTY_FILE_VERSION_TEMPLATE</u> Application Property "streamRecorderFileVersionTemplate": for specifying a default file version template string. Value: streamRecorderFileVersionTemplate
public static final	<u>PROPERTY_MOVE_FIRST_VIDEO_FRAME_TO_ZERO</u> Application Property "streamRecorderMoveFirstVideoFrameToZero": for specifying the default moveFirstVideoFrameToZero value used by StreamRecorder. Value: streamRecorderMoveFirstVideoFrameToZero
public static final	<u>PROPERTY_NOTIFY_LISTENER</u> Application Property "streamRecorderNotifyListener": for specifying the default listener which will receive StreamRecorder notifications. Value: streamRecorderNotifyListener
public static final	<u>PROPERTY_OUTPUT_FILE</u> Application Property "streamRecorderOutputFile": for specifying the default filename used by StreamRecorder for recordings. Value: streamRecorderOutputFile
public static final	<u>PROPERTY_OUTPUT_PATH</u> Application Property "streamRecorderOutputPath": for specifying the default path used by StreamRecorder to write recordings. Value: streamRecorderOutputPath
public static final	<u>PROPERTY_RECORD_DATA</u> Application Property "streamRecorderRecordData": for specifying the default recordData value used by StreamRecorders. Value: streamRecorderRecordData
public static final	<u>PROPERTY_SEGMENT_DURATION</u> Application Property "streamRecorderSegmentDuration": for specifying the default SegmentDuration used by StreamRecorder when the segmentationType is set to SEGMENT_BY_DURATION. Value: streamRecorderSegmentDuration
public static final	<u>PROPERTY_SEGMENT_SCHEDULE</u> Application Property "streamRecorderSegmentSchedule": for specifying the default SegmentSchedule string used by StreamRecorder when the segmentationType is set to SEGMENT_BY_SCHEDULE. Value: streamRecorderSegmentSchedule
public static final	<u>PROPERTY_SEGMENT_SIZE</u> Application Property "streamRecorderSegmentSize": for specifying the default SegmentSize used by StreamRecorder when the segmentationType is set to SEGMENT_BY_SIZE. Value: streamRecorderSegmentSize
public static final	<u>PROPERTY_SEGMENTATION_TYPE</u> Application Property "streamRecorderSegmentationType": for specifying the default segmentation type used by StreamRecorder. Value: streamRecorderSegmentationType
public static final	<u>PROPERTY_SPLIT_ON_TC_DISCONTINUITY</u> Application Property "streamRecorderSplitOnTcDiscontinuity": for specifying the default splitOnTcDiscontinuity value used by StreamRecorder. Value: streamRecorderSplitOnTcDiscontinuity

public static final	<u>PROPERTY_START_ON_KEYFRAME</u> Application Property "streamRecorderStartOnKeyFrame": for specifying the default startOnKeyFrame value used by StreamRecorder. Value: streamRecorderStartOnKeyFrame
public static final	<u>PROPERTY_USE_SIMPLE_FILE_VERSION_NAMING</u> Application Property "streamRecorderUseSimpleFileVersionNaming": for using StreamRecorderSimpleFileVersionDelegate instead of StreamRecorderFileVersionDelegate as the default file version naming delegate Value: streamRecorderUseSimpleFileVersionNaming
public static final	<u>PROPERTY_VERSIONING_OPTION</u> Application Property "streamRecorderVersioningOption": for specifying the default file versioning method used by StreamRecorders Value: streamRecorderVersioningOption
public static final	<u>RECORDER_STATE_ERROR</u> Value returned by IStreamRecorder getRecorderState() to indicate an error has occurred in the recorder. Value: -1
public static final	<u>RECORDER_STATE_ERROR_STRING</u> String value returned by IStreamRecorder getRecorderStateString() when the recorder state is RECORDER_STATE_ERROR. Value: Recorder Error
public static final	<u>RECORDER_STATE_PENDING</u> Value returned by IStreamRecorder getRecorderState() to indicate the recorder has been created but it's IStreamRecorder startRecording() method has not yet been called. Value: 0
public static final	<u>RECORDER_STATE_PENDING_STRING</u> String value returned by IStreamRecorder getRecorderStateString() when the recorder state is RECORDER_STATE_PENDING. Value: Pending
public static final	<u>RECORDER_STATE_RECORDING</u> Value returned by IStreamRecorder getRecorderState() to indicate the recorder is actively recording data to file. Value: 2
public static final	<u>RECORDER_STATE_RECORDING_STRING</u> String value returned by IStreamRecorder getRecorderStateString() when the recorder state is RECORDER_STATE_RECORDING. Value: Recording
public static final	<u>RECORDER_STATE_WAITING</u> Value returned by IStreamRecorder getRecorderState() to indicate the recorder has been started but the associated stream is not currently being received by the server (for example the stream is not being published from the encoder) Value: 1
public static final	<u>RECORDER_STATE_WAITING_STRING</u> String value returned by IStreamRecorder getRecorderStateString() when the recorder state is RECORDER_STATE_WAITING. Value: Waiting for Stream
public static final	<u>SEGMENT_BY_DURATION</u> Value: duration

public static final	SEGMENT_BY_SCHEDULE Value: schedule
public static final	SEGMENT_BY_SIZE Value: size
public static final	SEGMENT_NONE Value: none
public static final	VERSION_FILE Uses the file name returned by the configured fileVersionDelegate to version the new file Value: version

Fields

PROPERTY_DEBUG_ENABLE

```
public static final java.lang.String PROPERTY_DEBUG_ENABLE
```

Application Property "streamRecorderDebugEnabled": for enabling additional log statements throughout StreamRecorder

Add the property to the StreamRecorder Properties section of Application.xml.

Default value is false

Valid values are true or false

Constant value: **streamRecorderDebugEnabled**

PROPERTY_USE_SIMPLE_FILE_VERSION_NAMING

```
public static final java.lang.String PROPERTY_USE_SIMPLE_FILE_VERSION_NAMING
```

Application Property "streamRecorderUseSimpleFileVersionNaming": for using StreamRecorderSimpleFileVersionDelegate instead of StreamRecorderFileVersionDelegate as the default file version naming delegate

Add the property to the StreamRecorder Properties section of Application.xml.

Default value is false

Valid values are true, false

Constant value: **streamRecorderUseSimpleFileVersionNaming**

See Also:

[StreamRecorderSimpleFileVersionDelegate](#)

PROPERTY_SEGMENTATION_TYPE

```
public static final java.lang.String PROPERTY_SEGMENTATION_TYPE
```

Application Property "streamRecorderSegmentationType": for specifying the default segmentation type used by StreamRecorder.

Add the property to the StreamRecorder Properties section of Application.xml.

Valid values are: none, schedule, size, duration

(continued from last page)

Constant value: **streamRecorderSegmentationType**

See Also:

[StreamRecorderParameters.segmentationType](#)

SEGMENT_NONE

```
public static final java.lang.String SEGMENT_NONE
```

Constant value: **none**

SEGMENT_BY_SCHEDULE

```
public static final java.lang.String SEGMENT_BY_SCHEDULE
```

Constant value: **schedule**

SEGMENT_BY_SIZE

```
public static final java.lang.String SEGMENT_BY_SIZE
```

Constant value: **size**

SEGMENT_BY_DURATION

```
public static final java.lang.String SEGMENT_BY_DURATION
```

Constant value: **duration**

PROPERTY_FILE_FORMAT

```
public static final java.lang.String PROPERTY_FILE_FORMAT
```

Application Property "streamRecorderFileFormat": for specifying the default file format used by StreamRecorder.

Add the property to the StreamRecorder Properties section of Application.xml.

Valid values are: flv, mp4

Constant value: **streamRecorderFileFormat**

See Also:

[StreamRecorderParameters.fileFormat](#)

FORMAT_FLV

```
public static final java.lang.String FORMAT_FLV
```

Constant value: **flv**

FORMAT_MP4

```
public static final java.lang.String FORMAT_MP4
```

(continued from last page)

Constant value: **mp4**

PROPERTY_VERSIONING_OPTION

```
public static final java.lang.String PROPERTY_VERSIONING_OPTION
```

Application Property "streamRecorderVersioningOption": for specifying the default file versioning method used by StreamRecorders

Add the property to the StreamRecorder Properties section of Application.xml.

Valid values are: version, append, overwrite

Constant value: **streamRecorderVersioningOption**

See Also:

[StreamRecorderParameters.versioningOption](#)

VERSION_FILE

```
public static final java.lang.String VERSION_FILE
```

Uses the file name returned by the configured fileVersionDelegate to version the new file

Constant value: **version**

See Also:

[IStreamRecorderFileVersionDelegate](#)

[StreamRecorderFileVersionDelegate](#)

[StreamRecorderSimpleFileVersionDelegate](#)

APPEND_FILE

```
public static final java.lang.String APPEND_FILE
```

Appends new stream data to the existing file. NOTE: To ensure playback compatibility, it is highly recommended that this option only be used if all streams recorded to the file will be using identical video and audio codecs and configurations.

Constant value: **append**

OVERWRITE_FILE

```
public static final java.lang.String OVERWRITE_FILE
```

Overwrites the existing file with the new file

Constant value: **overwrite**

PROPERTY_START_ON_KEYFRAME

```
public static final java.lang.String PROPERTY_START_ON_KEYFRAME
```

Application Property "streamRecorderStartOnKeyFrame": for specifying the default startOnKeyFrame value used by StreamRecorder.

Add the property to the StreamRecorder Properties section of Application.xml.

Constant value: **streamRecorderStartOnKeyFrame**

See Also:

[StreamRecorderParameters.startOnKeyFrame](#)

PROPERTY_RECORD_DATA

```
public static final java.lang.String PROPERTY_RECORD_DATA
```

Application Property "streamRecorderRecordData": for specifying the default recordData value used by StreamRecorders.

Add the property to the StreamRecorder Properties section of Application.xml.

Constant value: **streamRecorderRecordData**

See Also:

[StreamRecorderParameters.recordData](#)

PROPERTY_MOVE_FIRST_VIDEO_FRAME_TO_ZERO

```
public static final java.lang.String PROPERTY_MOVE_FIRST_VIDEO_FRAME_TO_ZERO
```

Application Property "streamRecorderMoveFirstVideoFrameToZero": for specifying the default moveFirstVideoFrameToZero value used by StreamRecorder.

Add the property to the StreamRecorder Properties section of Application.xml.

Constant value: **streamRecorderMoveFirstVideoFrameToZero**

See Also:

[StreamRecorderParameters.moveFirstVideoFrameToZero](#)

PROPERTY_SPLIT_ON_TC_DISCONTINUITY

```
public static final java.lang.String PROPERTY_SPLIT_ON_TC_DISCONTINUITY
```

Application Property "streamRecorderSplitOnTcDiscontinuity": for specifying the default splitOnTcDiscontinuity value used by StreamRecorder.

Add the property to the StreamRecorder Properties section of Application.xml.

Constant value: **streamRecordersSplitOnTcDiscontinuity**

See Also:

[StreamRecorderParameters.splitOnTcDiscontinuity](#)

PROPERTY_SEGMENT_SIZE

```
public static final java.lang.String PROPERTY_SEGMENT_SIZE
```

Application Property "streamRecorderSegmentSize": for specifying the default SegmentSize used by StreamRecorder when the segmentationType is set to SEGMENT_BY_SIZE.

Add the property to the StreamRecorder Properties section of Application.xml.

Constant value: **streamRecorderSegmentSize**

See Also:

[StreamRecorderParameters.segmentSize](#)

(continued from last page)

PROPERTY_SEGMENT_DURATION

```
public static final java.lang.String PROPERTY_SEGMENT_DURATION
```

Application Property "streamRecorderSegmentDuration": for specifying the default SegmentDuration used by StreamRecorder when the segmentationType is set to SEGMENT_BY_DURATION.

Add the property to the StreamRecorder Properties section of Application.xml.

Constant value: **streamRecorderSegmentDuration**

See Also:

[StreamRecorderParameters.segmentDuration](#)

PROPERTY_SEGMENT_SCHEDULE

```
public static final java.lang.String PROPERTY_SEGMENT_SCHEDULE
```

Application Property "streamRecorderSegmentSchedule": for specifying the default SegmentSchedule string used by StreamRecorder when the segmentationType is set to SEGMENT_BY_SCHEDULE.

Add the property to the StreamRecorder Properties section of Application.xml.

Constant value: **streamRecorderSegmentSchedule**

See Also:

[StreamRecorderParameters.segmentSchedule](#)

PROPERTY_OUTPUT_PATH

```
public static final java.lang.String PROPERTY_OUTPUT_PATH
```

Application Property "streamRecorderOutputPath": for specifying the default path used by StreamRecorder to write recordings.

Add the property to the StreamRecorder Properties section of Application.xml.

Constant value: **streamRecorderOutputPath**

See Also:

[StreamRecorderParameters.outputPath](#)

PROPERTY_OUTPUT_FILE

```
public static final java.lang.String PROPERTY_OUTPUT_FILE
```

Application Property "streamRecorderOutputFile": for specifying the default filename used by StreamRecorder for recordings.

Add the property to the StreamRecorder Properties section of Application.xml.

Constant value: **streamRecorderOutputFile**

See Also:

[StreamRecorderParameters.outputFile](#)

(continued from last page)

PROPERTY_NOTIFY_LISTENER

```
public static final java.lang.String PROPERTY_NOTIFY_LISTENER
```

Application Property "streamRecorderNotifyListener": for specifying the default listener which will receive StreamRecorder notifications.

Add the property to the StreamRecorder Properties section of Application.xml.

Constant value: **streamRecorderNotifyListener**

See Also:

[StreamRecorderParameters.notifyListener](#)

PROPERTY_FILE_VERSION_DELEGATE

```
public static final java.lang.String PROPERTY_FILE_VERSION_DELEGATE
```

Application Property "streamRecorderFileVersionDelegate": for specifying a default custom file version delegate.

Add the property to the StreamRecorder Properties section of Application.xml.

Constant value: **streamRecorderFileVersionDelegate**

See Also:

[StreamRecorderParameters.fileVersionDelegate](#)

PROPERTY_FILE_VERSION_TEMPLATE

```
public static final java.lang.String PROPERTY_FILE_VERSION_TEMPLATE
```

Application Property "streamRecorderFileVersionTemplate": for specifying a default file version template string.
NOTE: only applicable when using the default fileVersionDelegate

Add the property to the StreamRecorder Properties section of Application.xml.

Constant value: **streamRecorderFileVersionTemplate**

See Also:

[StreamRecorderParameters.fileTemplate](#)

[StreamRecorderParameters.fileVersionDelegate](#)

RECORDER_STATE_ERROR

```
public static final int RECORDER_STATE_ERROR
```

Value returned by IStreamRecorder getRecorderState() to indicate an error has occurred in the recorder.

Constant value: **-1**

See Also:

[RECORDER_STATE_ERROR_STRING](#)

RECORDER_STATE_PENDING

```
public static final int RECORDER_STATE_PENDING
```

(continued from last page)

Value returned by IStreamRecorder getRecorderState() to indicate the recorder has been created but it's IStreamRecorder startRecording() method has not yet been called.

Constant value: **0**

See Also:

[RECORDER_STATE_PENDING_STRING](#)

RECORDER_STATE_WAITING

public static final int **RECORDER_STATE_WAITING**

Value returned by IStreamRecorder getRecorderState() to indicate the recorder has been started but the associated stream is not currently being received by the server (for example the stream is not being published from the encoder)

Constant value: **1**

See Also:

[RECORDER_STATE_WAITING_STRING](#)

RECORDER_STATE_RECORDING

public static final int **RECORDER_STATE_RECORDING**

Value returned by IStreamRecorder getRecorderState() to indicate the recorder is actively recording data to file.

Constant value: **2**

See Also:

[RECORDER_STATE_RECORDING_STRING](#)

RECORDER_STATE_ERROR_STRING

public static final java.lang.String **RECORDER_STATE_ERROR_STRING**

String value returned by IStreamRecorder getRecorderStateString() when the recorder state is RECORDER_STATE_ERROR.

Constant value: **Recorder Error**

See Also:

[RECORDER_STATE_ERROR](#)

RECORDER_STATE_PENDING_STRING

public static final java.lang.String **RECORDER_STATE_PENDING_STRING**

String value returned by IStreamRecorder getRecorderStateString() when the recorder state is RECORDER_STATE_PENDING.

Constant value: **Pending**

See Also:

[RECORDER_STATE_PENDING](#)

(continued from last page)

RECORDER_STATE_WAITING_STRING

```
public static final java.lang.String RECORDER_STATE_WAITING_STRING
```

String value returned by IStreamRecorder getRecorderStateString() when the recorder state is RECORDER_STATE_WAITING.

Constant value: **Waiting for Stream**

See Also:

[RECORDER_STATE_PENDING](#)

RECORDER_STATE_RECORDING_STRING

```
public static final java.lang.String RECORDER_STATE_RECORDING_STRING
```

String value returned by IStreamRecorder getRecorderStateString() when the recorder state is RECORDER_STATE_RECORDING.

Constant value: **Recording**

See Also:

[RECORDER_STATE_RECORDING](#)

com.wowza.wms.livestreamrecord.manager Interface IStreamRecorderFileVersionDelegate

All Known Implementing Classes:

[StreamRecorderSimpleFileVersionDelegate](#), [StreamRecorderFileVersionDelegate](#)

public interface **IStreamRecorderFileVersionDelegate**
extends

This interface is used to implement a custom file versioning scheme during live stream recording.

The `getFilename` method is called under 2 conditions

- 1 - when a recorder needs to create a new output file, and an file already exists by the same name.
- 2 - when a recorder creates a new file and has been configured to use any `segmentationType` other than none

The string returned must be a valid and fully qualified filename, including the full path.

For example:

`c:\test\myFile.flv`

`usr/home/content/myFile.mp4`

See Also:

[StreamRecorderParameters.segmentationType](#)

Method Summary

String	getFilename (IStreamRecorder recContext)
--------	---

Methods

getFilename

public String **getFilename**([IStreamRecorder](#) recContext)

Parameters:

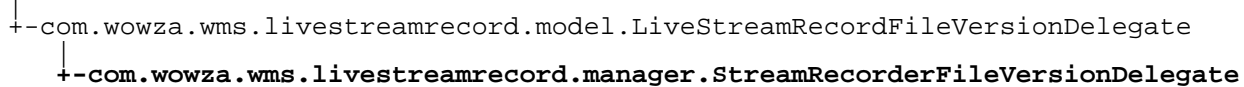
recContext - The IStreamRecorder requesting the file name

Returns:

String The versioned file name

com.wowza.wms.livestreamrecord.manager Class StreamRecorderFileVersionDelegate

java.lang.Object



All Implemented Interfaces:

[IStreamRecorderFileVersionDelegate](#), com.wowza.wms.livestreamrecord.model.ILiveStreamRecordFileVersionDelegate

```

public class StreamRecorderFileVersionDelegate
    extends com.wowza.wms.livestreamrecord.model.LiveStreamRecordFileVersionDelegate
    implements com.wowza.wms.livestreamrecord.model.ILiveStreamRecordFileVersionDelegate,
IStreamRecorderFileVersionDelegate
  
```

Default FileVersionDelegate class

This delegate allows the configuration of a custom FileName using a template string. The tags in the template string are replaced with their associated values, to generate a filename.

Valid template tags are;

`${SourceStreamName}` - The name of the stream

`${BaseFileName}` - The basefile name, which is the filename prior to any versioning changes

`${SegmentNumber}` - An incrementing value representing the current number of segments which have been created for this recording, starting at 0

`${RecordingStartTime}` - The time at which the recording was started formatted as yyyy-MM-dd-HH.mm.ss.SSS-zzz

`${SegmentTime}` - Time time at which the segment was created

The default template string is

`${BaseFileName}_${RecordingStartTime}_${SegmentNumber}`

Fields inherited from class

com.wowza.wms.livestreamrecord.model.LiveStreamRecordFileVersionDelegate

ALT_DATETIME_FORMAT, BASE_NAME_TAG, DATETIME_FORMAT, SEGMENT_NUMBER_TAG, SEGMENT_TIME_TAG, START_TIME_TAG, STREAM_NAME_TAG

Constructor Summary

public	StreamRecorderFileVersionDelegate()
--------	---

Method Summary

String	getFilename(IStreamRecorder recContext) Creates a new filename using the default template string,
--------	--

Methods inherited from class

com.wowza.wms.livestreamrecord.model.LiveStreamRecordFileVersionDelegate

createFileName, getFilename, getFileTemplate, setFileTemplate

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface`com.wowza.wms.livestreamrecord.model.ILiveStreamRecordFileVersionDelegate``getFilename`**Methods inherited from interface**`com.wowza.wms.livestreamrecord.manager.IStreamRecorderFileVersionDelegate``getFilename`

Constructors

StreamRecorderFileVersionDelegate

```
public StreamRecorderFileVersionDelegate()
```

Methods

getFilename

```
public String getFilename(IStreamRecorder recContext)
```

Creates a new filename using the default template string,

Parameters:

recContext - The IStreamRecorder requesting the file name

Returns:

String The versioned file name

com.wowza.wms.livestreamrecord.manager

Class StreamRecorderParameters

java.lang.Object

└─com.wowza.wms.livestreamrecord.manager.StreamRecorderParameters

public class **StreamRecorderParameters**
extends Object

This class is used to configure the recording parameters for the StreamRecorder class.
All members of this class are public and therefore can be accessed and modified directly.

Field Summary

public	fileFormat Configures the output file format
public	fileTemplate Configures the file version template string used by the default fileVersionDelegate, to generate a versioned file name for the recorder NOTE: if set to null then a default file template string, as defined by StreamRecorderFileVersionDelegate, will be used
public	fileVersionDelegate Configures the fileVersionDelegate for the recorder Valid values are the fully qualified class path to the class which implements the IStreamRecorderFileVersionDelegate interface. Default value is com.wowza.wms.livestreamrecord.manager.StreamRecorderFileVersionDele gate
public	moveFirstVideoFrameToZero Configures whether or not the recorder will set the first packet's timecode to 0, when recording fileFormat is FORMAT_MP4
public	notifyListener Configures the StreamRecorder listener for the recorder Valid values are the fully qualified class path to the class which implements the IStreamRecorderActionNotify interface.
public	outputFile Configures the output file name used for the recording If segmentationType is not SEGMENT_NONE, then this value is used as the base file name for the file segments when using either of the fileVersionDelegates NOTE 1: When set to null or "", the recorder will create the filename based upon the stream name and the value of fileFormat.
public	outputPath Configures the output directory where recorded file will be written NOTE: When set to null or "", the recorder will write files to the default content directory for the application.
public	recordData Configures whether or not the recorder will write data packets to the output file.
public	segmentationType Configures the type of recording segmentation the recorder will use Segmentation provides a means by which the recording of an incoming live stream can be split into a series of files in order to manage them more easily.

public	<u>segmentDuration</u> Configures the output file size in milliseconds for each recording segment, when segmentationType is SEGMENT_BY_DURATION. This value is specified in milliseconds
public	<u>segmentSchedule</u> Configures the crontab schedule used to segment a recording, when segmentationType is SEGMENT_BY_SCHEDULE.
public	<u>segmentSize</u> Configures the output file size in bytes for each recording segment, when segmentationType is SEGMENT_BY_SIZE. This value is specified in bytes
public	<u>splitOnTcDiscontinuity</u> Configures whether or not the recorder will create a new segment for a recording when packet timecode discontinuities are detected A discontinuity is said to occur when 10 consecutive packets have timecodes which jump backward in time.
public	<u>startOnKeyFrame</u> Configures whether or not the recorder will wait until a video key frame is received to start writing packets to the output file, thus ensuring the first packet in the file contains a video key frame.
public	<u>versioningOption</u> Configures the behavior when a new recording file is created with a name that already exists

Constructor Summary

public	<u>StreamRecorderParameters</u> (<u>IApplicationInstance</u> appInstance) The constructor will read any defaults set via properties in application.xml and will use those values as the default values for this class.
--------	--

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

segmentationType

public java.lang.String **segmentationType**

Configures the type of recording segmentation the recorder will use

Segmentation provides a means by which the recording of an incoming live stream can be split into a series of files in order to manage them more easily. The files can be limited by; size, duration, or split at a specific time (e.g. at the top of every hour). If SEGMENT_NONE is used, then the recording is written to a single file.

Valid values are defined in IStreamRecorderConstants

See Also:

[IStreamRecorderConstants.SEGMENT_NONE](#)
[IStreamRecorderConstants.SEGMENT_BY_DURATION](#)
[IStreamRecorderConstants.SEGMENT_BY_SIZE](#)
[IStreamRecorderConstants.SEGMENT_BY_SCHEDULE](#)

(continued from last page)

versioningOption

```
public java.lang.String versioningOption
```

Configures the behavior when a new recording file is created with a name that already exists

NOTE: APPEND_FILE and OVERWRITE_FILE are only applicable when the segmentationType is SEGMENT_NONE

Valid values are defined in IStreamRecorderConstants

See Also:

[IStreamRecorderConstants.VERSION_FILE](#)
[IStreamRecorderConstants.APPEND_FILE](#)
[IStreamRecorderConstants.OVERWRITE_FILE](#)
[segmentSize](#)
[segmentDuration](#)
[segmentSchedule](#)

fileFormat

```
public java.lang.String fileFormat
```

Configures the output file format

Valid values are defined in IStreamRecorderConstants

See Also:

[IStreamRecorderConstants.FORMAT_FLV](#)
[IStreamRecorderConstants.FORMAT_MP4](#)

startOnKeyFrame

```
public boolean startOnKeyFrame
```

Configures whether or not the recorder will wait until a video key frame is received to start writing packets to the output file, thus ensuring the first packet in the file contains a video key frame.

Valid values are true, false
default value is true

recordData

```
public boolean recordData
```

Configures whether or not the recorder will write data packets to the output file.

Valid values are true, false
default value is false

moveFirstVideoFrameToZero

```
public boolean moveFirstVideoFrameToZero
```

Configures whether or not the recorder will set the first packet's timecode to 0, when recording fileFormat is FORMAT_MP4

Valid values are true, false
default value is true

(continued from last page)

splitOnTcDiscontinuity

```
public boolean splitOnTcDiscontinuity
```

Configures whether or not the recorder will create a new segment for a recording when packet timecode discontinuities are detected

A discontinuity is said to occur when 10 consecutive packets have timecodes which jump backward in time. i.e. the current packet's timecode is less than the previous packet's timecode.

Valid values are true, false
default value is false

segmentSize

```
public long segmentSize
```

Configures the output file size in bytes for each recording segment, when segmentationType is SEGMENT_BY_SIZE.
This value is specified in bytes

Valid values are any positive long value greater than 0, however a minimum size of 1048576 bytes (1 MB) is highly recommended to ensure the recorded file segments can start on a video key frame if configured to do so.
default value is 10485760 bytes (10 MB)

segmentDuration

```
public long segmentDuration
```

Configures the output file size in milliseconds for each recording segment, when segmentationType is SEGMENT_BY_DURATION.
This value is specified in milliseconds

Valid values are any positive long value greater than 0, however a minimum size of 60000 ms (1 minute) is highly recommended to ensure the recorded file segments can start on a video key frame if configured to do so.
default value is 900000 ms (15 minutes)

segmentSchedule

```
public java.lang.String segmentSchedule
```

(continued from last page)

Configures the crontab schedule used to segment a recording, when segmentationType is SEGMENT_BY_SCHEDULE.

Crontab expressions

A crontab expression specifies the segmentation interval in minutes, hours, months, years, days of the month, or days of the week. The available crontab fields are (in order):

[Minute] [Hour] [Day_of_the_Month] [Month_of_the_Year] [Day_of_the_Week] [Year]

where:

Minute - Specifies the minute value (between 0 and 59) at which the segmentation occurs.

Hour - Specifies the hour (between 0 and 23 based on a 24-hour clock) at which the segmentation occurs.

Day_of_the_Month - Specifies the day (between 0 and 31) at which the split occurs. To specify that live recordings always be segmentation on the last day of the month, specify 31.

Month_of_the_Year - Specifies the month at which the segmentation occurs. You can specify the month by using the abbreviated name of the month (for example, Jan) or a numerical equivalent (for example, 1).

Day_of_the_Week - Specifies the day in a week at which the segmentation occurs. You can specify the day by using the abbreviated name of the day (for example, Mon) or a numerical equivalent (for example, 1).

Year - Specifies the year at which the segmentation occurs (for example, 2013).

Example

To segmentation a live recording every 30 minutes (at the top of each hour and every half-hour):

0,30 * * * * *

0,30 (simplified version)

for more details see (<http://www.nncron.ru/help/EN/working/cron-format.htm>).

Note: The nnCron nonstandard question mark character (?), which specifies that the server startup time is substituted into the cron expression, is NOT currently supported.

Valid values are any valid crontab expression exception for '?'

default value is "0 * * * * *" (segment every hour at the top of the hour)

outputFile

```
public java.lang.String outputFile
```

Configures the output file name used for the recording

If segmentationType is not SEGMENT_NONE, then this value is used as the base file name for the file segments when using either of the fileVersionDelegates

NOTE 1: When set to null or "", the recorder will create the filename based upon the stream name and the value of fileFormat.

NOTE 2: The fileFormat will be appended to the file name to ensure the file extension matches the actual file format written.

Example mytestRecording, with no fileFormat specified will result in the default fileFormat being used and a recorded file name of mytestRecording.mp4 myOtherTest.mp4, with the the fileFormat specified as 1 (FLV), will result in the recorded file name of myOtherTest.mp4.flv

Valid values are any valid filename

default value is null

outputPath

```
public java.lang.String outputPath
```

Configures the output directory where recorded file will be written

NOTE: When set to null or "", the recorder will write files to the default content directory for the application.

Examples c:\myfiles\recordings /home/user/myfiles/recordings

Valid values are any valid path.

default value is null

notifyListener

```
public com.wowza.wms.livestreamrecord.manager.IStreamRecorderActionNotify  
notifyListener
```

(continued from last page)

Configures the StreamRecorder listener for the recorder

Valid values are the fully qualified class path to the class which implements the IStreamRecorderActionNotify interface. default value is null

See Also:

[IStreamRecorderActionNotify](#)

fileVersionDelegate

```
public com.wowza.wms.livestreamrecord.manager.IStreamRecorderFileVersionDelegate  
fileVersionDelegate
```

Configures the fileVersionDelegate for the recorder

Valid values are the fully qualified class path to the class which implements the IStreamRecorderFileVersionDelegate interface.

Default value is com.wowza.wms.livestreamrecord.manager.StreamRecorderFileVersionDelegate

See Also:

[IStreamRecorderFileVersionDelegate](#)

[StreamRecorderFileVersionDelegate](#)

[StreamRecorderSimpleFileVersionDelegate](#)

fileTemplate

```
public java.lang.String fileTemplate
```

Configures the file version template string used by the default fileVersionDelegate, to generate a versioned file name for the recorder

NOTE: if set to null then a default file template string, as defined by StreamRecorderFileVersionDelegate, will be used

Valid values are null or a valid file version template string

Default value is null

See Also:

[StreamRecorderFileVersionDelegate](#)

Constructors

StreamRecorderParameters

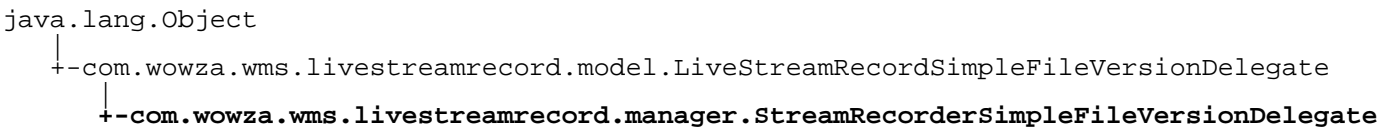
```
public StreamRecorderParameters(IApplicationInstance appInstance)
```

The constructor will read any defaults set via properties in application.xml and will use those values as the default values for this class.

Parameters:

appInstance

com.wowza.wms.livestreamrecord.manager
Class StreamRecorderSimpleFileVersionDelegate



All Implemented Interfaces:
[IStreamRecorderFileVersionDelegate](#), com.wowza.wms.livestreamrecord.model.ILiveStreamRecordFileVersionDelegate

```
public class StreamRecorderSimpleFileVersionDelegate
extends com.wowza.wms.livestreamrecord.model.LiveStreamRecordSimpleFileVersionDelegate
implements com.wowza.wms.livestreamrecord.model.ILiveStreamRecordFileVersionDelegate ,
IStreamRecorderFileVersionDelegate
```

StreamRecorderSimpleFileVersionDelegate class

This delegate uses the simple file versioning method to generate a file version name. The simple method is to version the existing file and always use the "basename" for the current file

Simple File Versioning:
The simple file versioning scheme attempts to resolve file name conflicts by appending "_1" to the end of the existing file, so that the current recording can use the requested file name. If a file by that name already exists the appended number will be incremented and renaming will be tried again. This will continue until a non-conflicting file name is found.

For example: if the file name is "myStream.flv", and that file already exists, "myStream_1.flv" will be tried. If that still conflicts with an existing file, "myStream_2.flv" will be tried. This process continues until a non conflicting name is found. The recorder will then record to "mystream.flv"

Constructor Summary	
public	StreamRecorderSimpleFileVersionDelegate()

Method Summary	
String	getFilename(IStreamRecorder recContext) Versions the existing file by appending _#, and returns the original file name for use as the current file.

Methods inherited from class com.wowza.wms.livestreamrecord.model.LiveStreamRecordSimpleFileVersionDelegate
createFileName, getFilename

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface com.wowza.wms.livestreamrecord.model.ILiveStreamRecordFileVersionDelegate
getFilename

Methods inherited from interface[com.wowza.wms.livestreamrecord.manager.IStreamRecorderFileVersionDelegate](#)[getFilename](#)

Constructors

StreamRecorderSimpleFileVersionDelegate

```
public StreamRecorderSimpleFileVersionDelegate()
```

Methods

getFilename

```
public String getFilename(IStreamRecorder recContext)
```

Versions the existing file by appending _#, and returns the original file name for use as the current file.

Parameters:

recContext - The IStreamRecorder requesting the file name

Returns:

String The original file name

Package

com.wowza.wms.logging

com.wowza.wms.logging Interface Constants

public interface **Constants**
extends

Constants used internally throughout log4j.

Field Summary

public static final	<u>ABSOLUTE_FORMAT</u> ABSOLUTE string literal. Value: ABSOLUTE
public static final	<u>ABSOLUTE_TIME_PATTERN</u> SimpleTimePattern for ABSOLUTE. Value: HH:mm:ss,SSS
public static final	<u>APPLICATION_KEY</u> application string literal. Value: application
public static final	<u>CODES_HREF</u> Codes URL string literal. Value: http://logging.apache.org/log4j/docs/codes.html
public static final	<u>CONFIGURATOR_CLASS_KEY</u> log4j.configuratorClass string literal. Value: log4j.configuratorClass
public static final	<u>DATE_AND_TIME_FORMAT</u> DATE string literal. Value: DATE
public static final	<u>DATE_AND_TIME_PATTERN</u> SimpleTimePattern for DATE. Value: dd MMM yyyy HH:mm:ss,SSS
public static final	<u>DEFAULT_CONFIGURATION_FILE</u> The default property file name for automatic configuration. Value: log4j.properties
public static final	<u>DEFAULT_CONFIGURATION_KEY</u> log4j.configuration string literal. Value: log4j.configuration
public static final	<u>DEFAULT_REPOSITORY_NAME</u> The name of the default repository is "default" (without the quotes). Value: default
public static final	<u>DEFAULT_XML_CONFIGURATION_FILE</u> The default XML configuration file name for automatic configuration. Value: log4j.xml

public static final	HOSTNAME_KEY hostname string literal. Value: hostname
public static final	ISO8601_FORMAT ISO8601 string literal. Value: ISO8601
public static final	ISO8601_PATTERN SimpleTimePattern for ISO8601. Value: yyyy-MM-dd HH:mm:ss,SSS
public static final	JNDI_CONTEXT_NAME JNDI context name string literal. Value: java:comp/env/log4j/context-name
public static final	LOG4J_ID_KEY log4jid string literal. Value: log4jid
public static final	LOG4J_PACKAGE_NAME log4j package name string literal. Value: org.apache.log4j
public static final	RECEIVER_NAME_KEY receiver string literal. Value: receiver
public static final	TEMP_CONSOLE_APPENDER_NAME TEMP_CONSOLE_APPENDER string literal. Value: TEMP_CONSOLE_APPENDER
public static final	TEMP_LIST_APPENDER_NAME TEMP_LIST_APPENDER string literal. Value: TEMP_LIST_APPENDER
public static final	TIMESTAMP_RULE_FORMAT time stamp pattern string literal. Value: yyyy/MM/dd HH:mm:ss

Fields

LOG4J_PACKAGE_NAME

public static final java.lang.String **LOG4J_PACKAGE_NAME**

log4j package name string literal.
Constant value: **org.apache.log4j**

DEFAULT_REPOSITORY_NAME

public static final java.lang.String **DEFAULT_REPOSITORY_NAME**

The name of the default repository is "default" (without the quotes).
Constant value: **default**

APPLICATION_KEY

```
public static final java.lang.String APPLICATION_KEY
```

application string literal.

Constant value: **application**

HOSTNAME_KEY

```
public static final java.lang.String HOSTNAME_KEY
```

hostname string literal.

Constant value: **hostname**

RECEIVER_NAME_KEY

```
public static final java.lang.String RECEIVER_NAME_KEY
```

receiver string literal.

Constant value: **receiver**

LOG4J_ID_KEY

```
public static final java.lang.String LOG4J_ID_KEY
```

log4jid string literal.

Constant value: **log4jid**

TIMESTAMP_RULE_FORMAT

```
public static final java.lang.String TIMESTAMP_RULE_FORMAT
```

time stamp pattern string literal.

Constant value: **yyyy/MM/dd HH:mm:ss**

DEFAULT_CONFIGURATION_FILE

```
public static final java.lang.String DEFAULT_CONFIGURATION_FILE
```

The default property file name for automatic configuration.

Constant value: **log4j.properties**

DEFAULT_XML_CONFIGURATION_FILE

```
public static final java.lang.String DEFAULT_XML_CONFIGURATION_FILE
```

The default XML configuration file name for automatic configuration.

Constant value: **log4j.xml**

DEFAULT_CONFIGURATION_KEY

```
public static final java.lang.String DEFAULT_CONFIGURATION_KEY
```

log4j.configuration string literal.

Constant value: **log4j.configuration**

(continued from last page)

CONFIGURATOR_CLASS_KEY

```
public static final java.lang.String CONFIGURATOR_CLASS_KEY
```

log4j.configuratorClass string literal.

Constant value: **log4j.configuratorClass**

JNDI_CONTEXT_NAME

```
public static final java.lang.String JNDI_CONTEXT_NAME
```

JNDI context name string literal.

Constant value: **java:comp/env/log4j/context-name**

TEMP_LIST_APPENDER_NAME

```
public static final java.lang.String TEMP_LIST_APPENDER_NAME
```

TEMP_LIST_APPENDER string literal.

Constant value: **TEMP_LIST_APPENDER**

TEMP_CONSOLE_APPENDER_NAME

```
public static final java.lang.String TEMP_CONSOLE_APPENDER_NAME
```

TEMP_CONSOLE_APPENDER string literal.

Constant value: **TEMP_CONSOLE_APPENDER**

CODES_HREF

```
public static final java.lang.String CODES_HREF
```

Codes URL string literal.

Constant value: **http://logging.apache.org/log4j/docs/codes.html**

ABSOLUTE_FORMAT

```
public static final java.lang.String ABSOLUTE_FORMAT
```

ABSOLUTE string literal.

Constant value: **ABSOLUTE**

ABSOLUTE_TIME_PATTERN

```
public static final java.lang.String ABSOLUTE_TIME_PATTERN
```

SimpleTimePattern for ABSOLUTE.

Constant value: **HH:mm:ss,SSS**

DATE_AND_TIME_FORMAT

```
public static final java.lang.String DATE_AND_TIME_FORMAT
```

DATE string literal.

Constant value: **DATE**

DATE_AND_TIME_PATTERN

```
public static final java.lang.String DATE_AND_TIME_PATTERN
```

(continued from last page)

SimpleTimePattern for DATE.

Constant value: **dd MMM yyyy HH:mm:ss,SSS**

ISO8601_FORMAT

```
public static final java.lang.String ISO8601_FORMAT
```

ISO8601 string literal.

Constant value: **ISO8601**

ISO8601_PATTERN

```
public static final java.lang.String ISO8601_PATTERN
```

SimpleTimePattern for ISO8601.

Constant value: **yyyy-MM-dd HH:mm:ss,SSS**

com.wowza.wms.logging Interface ILogNotify

All Known Implementing Classes:

[LogNotifyCalculateIncremental](#)

public interface **ILogNotify**
extends

ILogNotify: Interface to add custom fields to the Wowza Pro log files. To add your own custom log fields, define a class that implements this interface. The onLog method will be called each time the Wowza Pro server logs a message. Here is an example of a simple ILogNotify class that logs the current system time in milliseconds as a Long (systime-long) and as a String (systime-string).

```
package com.wowza.wms.plugin.newlogfields;

import org.apache.log4j.*;

import com.wowza.wms.logging.*;
import com.wowza.wms.stream.*;

public class LogNotifyDocs implements ILogNotify
{
    public void onLog(Level level, String comment, IMediaStream stream, String category,
String event, int status, String context)
    {
        long systime = System.currentTimeMillis();
        WMSLoggerFactory.putGlobalLogValue("systime-long", new Long(systime));
        WMSLoggerFactory.putGlobalLogValue("systime-string", systime+"");
    }
}
```

*Note: To get any of the values currently being logged use the logging API
WMSLoggerFactory.getGlobalLogValue(WMSLoggerIDs.FD_*)*

To add your class to Wowza Pro, compile your class into a .class file, bind the class into a .jar file and copy the .jar file into the Wowza Pro server /lib folder. Next, edit:

- [install-dir]/bin/setenv.bat (Windows)
- [install-dir]/bin/setenv.sh (Linux, OSX, Solaris)

Add **-Dcom.wowza.wms.logging.LogNotify=[full-path-to-your-ILogNotify-class]** to the **JAVA_OPTS**. For example for the class above the JAVA_OPTS would look like:

Linux/OSX

JAVA_OPTS="-Xmx768M -

Dcom.wowza.wms.logging.LogNotify=com.wowza.wms.plugin.newlogfields.NewLogFields"

Windows

JAVA_OPTS=-Xmx768M -

Dcom.wowza.wms.logging.LogNotify=com.wowza.wms.plugin.newlogfields.NewLogFields

Next, edit `[install-dir]/conf/log4j.properties` and add the new field names to any `log4j.appender.[appender-name].layout.Fields` fields lists to which you want to log these values. For example:

```
log4j.appender.stdout.layout.Fields=x-severity,x-category,x-event,x-ctx,x-comment,systemtime-long,systemtime-string
```

Method Summary

void

[onLog](#)(org.apache.log4j.Level level, String comment, [IMediaStream](#) stream, String category, String event, int status, String context)

Called each time the server logs a message.

Methods

onLog

```
public void onLog(org.apache.log4j.Level level,
    String comment,
    IMediaStream stream,
    String category,
    String event,
    int status,
    String context)
```

Called each time the server logs a message.

Parameters:

level - log level as defined by (org.apache.log4j.Level)
comment - comment part of the log statement
stream - if stream category log message it's the source stream
category - log category as defined by WMSLoggerIDs.CAT_*
event - log event as defined by WMSLoggerIDs.EVT_*
status - log status (same as HTTP status field) as defined by WMSLoggerIDs.STAT_*
context - log context value like stream name, vhost name, application name

com.wowza.wms.logging Class LogNotifyCalculateIncremental

java.lang.Object

└─com.wowza.wms.logging.LogNotifyCalculateIncremental

All Implemented Interfaces:

[ILogNotify](#)

public class **LogNotifyCalculateIncremental**
extends Object
implements [ILogNotify](#)

LogNotifyCalculateIncremental: pre-built implementation of ILogNotify that adds 5 new log fields:

- cs-bytes-inc (client to server bytes streamed since last log entry)
- sc-bytes-inc (server to client bytesstreamed since last log entry)
- cs-stream-bytes-inc (client to server stream bytesstreamed since last log entry)
- sc-stream-bytes-inc (server to client stream bytesstreamed since last log entry)
- x-duration-inc (time in seconds that have passed since the last log event)

These are incremental versions of the regular log values. To add this to Wowza Pro edit:

- [install-dir]/bin/setenv.bat (Windows)
- [install-dir]/bin/setenv.sh (Linux, OSX)

Add **-Dcom.wowza.wms.logging.LogNotify=com.wowza.wms.logging.LogNotifyCalculateIncremental** to the **JAVA_OPTS**. Also, edit [install-dir]/conf/log4j.properties and add these field names to any log4j.appender.[appender-name].layout.Fields fields lists to which you want to log these values.

*NOTE: These new log values are only accurate if all the events are included for the **session** and **stream** log categories.*

Field Summary

public static final	FD_cs_bytes_inc Value: cs-bytes-inc
public static final	FD_cs_stream_bytes_inc Value: cs-stream-bytes-inc
public static final	FD_sc_bytes_inc Value: sc-bytes-inc
public static final	FD_sc_stream_bytes_inc Value: sc-stream-bytes-inc
public static final	FD_x_duration_inc Value: x-duration-inc

Constructor Summary

public	LogNotifyCalculateIncremental()
--------	---

Method Summary

void	onLog (org.apache.log4j.Level level, String comment, IMediaStream stream, String category, String event, int status, String context)
------	--

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface [com.wowza.wms.logging.ILogNotify](#)

[onLog](#)

Fields

FD_cs_bytes_inc

public static final java.lang.String **FD_cs_bytes_inc**

Constant value: **cs-bytes-inc**

FD_sc_bytes_inc

public static final java.lang.String **FD_sc_bytes_inc**

Constant value: **sc-bytes-inc**

FD_cs_stream_bytes_inc

public static final java.lang.String **FD_cs_stream_bytes_inc**

Constant value: **cs-stream-bytes-inc**

FD_sc_stream_bytes_inc

public static final java.lang.String **FD_sc_stream_bytes_inc**

Constant value: **sc-stream-bytes-inc**

FD_x_duration_inc

public static final java.lang.String **FD_x_duration_inc**

Constant value: **x-duration-inc**

Constructors

(continued from last page)

LogNotifyCalculateIncremental

```
public LogNotifyCalculateIncremental()
```

Methods

onLog

```
public void onLog(org.apache.log4j.Level level,  
    String comment,  
    IMediaStream stream,  
    String category,  
    String event,  
    int status,  
    String context)
```

com.wowza.wms.logging Class UDPAppender

```

java.lang.Object
|
+-org.apache.log4j.AppenderSkeleton
|
+-com.wowza.wms.logging.UDPAppender

```

All Implemented Interfaces:

PortBased, org.apache.log4j.spi.OptionHandler, org.apache.log4j.Appender

```

public class UDPAppender
extends org.apache.log4j.AppenderSkeleton
implements org.apache.log4j.Appender, org.apache.log4j.spi.OptionHandler, PortBased

```

Sends log information as a UDP datagrams.

The UDPAppender is meant to be used as a diagnostic logging tool so that logging can be monitored by a simple UDP client.

Messages are not sent as LoggingEvent objects but as text after applying the designated Layout.

The port and remoteHost properties can be set in configuration properties. By setting the remoteHost to a broadcast address any number of clients can listen for log messages.

This was inspired and really extended/copied from SocketAppender. Please see the docs for the proper credit to the authors of that class.

Field Summary

public static final	DEFAULT_PORT The default port number for the UDP packets, 9991. Value: 9991
---------------------	--

Fields inherited from class org.apache.log4j.AppenderSkeleton

closed, errorHandler, headFilter, layout, name, tailFilter, threshold

Constructor Summary

public	UDPAppender ()
public	UDPAppender (java.net.InetAddress address, int port) Sends UDP packets to the address and port.
public	UDPAppender (String host, int port) Sends UDP packets to the address and port.

Method Summary

void	activateOptions () Open the UDP sender for the RemoteHost and Port .
void	append (org.apache.log4j.spi.LoggingEvent event)

void	<code>cleanup()</code> Close the UDP Socket and release the underlying connector thread if it has been created
void	<code>close()</code> Close this appender.
String	<code>getApplication()</code> Returns value of the App option.
String	<code>getEncoding()</code> Returns value of the Encoding option.
int	<code>getPort()</code> Returns value of the Port option.
String	<code>getRemoteHost()</code> Returns value of the RemoteHost option.
boolean	<code>isActive()</code>
boolean	<code>requiresLayout()</code> The UDPAppender uses layouts.
void	<code>setApplication(String app)</code> The App option takes a string value which should be the name of the application getting logged.
void	<code>setEncoding(String encoding)</code> The Encoding option specifies how the bytes are encoded.
void	<code>setPort(int port)</code> The Port option takes a positive integer representing the port where UDP packets will be sent.
void	<code>setRemoteHost(String host)</code> The RemoteHost option takes a string value which should be the host name or ipaddress to send the UDP packets.

Methods inherited from class org.apache.log4j.AppenderSkeleton

activateOptions, addFilter, append, clearFilters, doAppend, finalize, getErrorHandler, getFilter, getFirstFilter, getLayout, getName, getThreshold, isAsSevereAsThreshold, setErrorHandler, setLayout, setName, setThreshold

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface org.apache.log4j.Appender

addFilter, clearFilters, close, doAppend, getErrorHandler, getFilter, getLayout, getName, requiresLayout, setErrorHandler, setLayout, setName

Methods inherited from interface org.apache.log4j.spi.OptionHandler

activateOptions

Methods inherited from interface com.wowza.wms.logging.PortBased

```
getPort
```

Methods inherited from interface `com.wowza.wms.logging.NetworkBased`

```
getName, isActive
```

Fields

DEFAULT_PORT

```
public static final int DEFAULT_PORT
```

The default port number for the UDP packets, 9991.
Constant value: **9991**

Constructors

UDPAppender

```
public UDPAppender()
```

UDPAppender

```
public UDPAppender(java.net.InetAddress address,  
                    int port)
```

Sends UDP packets to the address and port.

UDPAppender

```
public UDPAppender(String host,  
                    int port)
```

Sends UDP packets to the address and port.

Methods

activateOptions

```
public void activateOptions()
```

Open the UDP sender for the **RemoteHost** and **Port**.

close

```
public void close()
```

Close this appender.

This will mark the appender as closed and call then [cleanUp\(\)](#) method.

(continued from last page)

cleanUp

```
public void cleanUp()
```

Close the UDP Socket and release the underlying connector thread if it has been created

append

```
public void append(org.apache.log4j.spi.LoggingEvent event)
```

isActive

```
public boolean isActive()
```

requiresLayout

```
public boolean requiresLayout()
```

The **UDPAppender** uses layouts. Hence, this method returns true.

setRemoteHost

```
public void setRemoteHost(String host)
```

The **RemoteHost** option takes a string value which should be the host name or ipaddress to send the UDP packets.

getRemoteHost

```
public String getRemoteHost()
```

Returns value of the **RemoteHost** option.

setApplication

```
public void setApplication(String app)
```

The **App** option takes a string value which should be the name of the application getting logged. If property was already set (via system property), don't set here.

getApplication

```
public String getApplication()
```

Returns value of the **App** option.

setEncoding

```
public void setEncoding(String encoding)
```

The **Encoding** option specifies how the bytes are encoded. If this option is not specified, the System encoding is used.

getEncoding

```
public String getEncoding()
```

(continued from last page)

Returns value of the **Encoding** option.

setPort

```
public void setPort(int port)
```

The **Port** option takes a positive integer representing the port where UDP packets will be sent.

getPort

```
public int getPort()
```

Returns value of the **Port** option.

com.wowza.wms.logging Class WMSLogger

```

java.lang.Object
  |
  +- org.apache.log4j.Category
    |
    +- org.apache.log4j.Logger
      |
      +- com.wowza.wms.logging.WMSLogger
  
```

All Implemented Interfaces:

org.apache.log4j.spi.AppenderAttachable

```

public class WMSLogger
extends org.apache.log4j.Logger
  
```

Fields inherited from class org.apache.log4j.Category

additive, level, name, parent, repository, resourceBundle

Constructor Summary

public	WMSLogger (String name, org.apache.log4j.Logger tmpLogger)
public	WMSLogger (String name)

Method Summary

void	debug (Object o)
void	debug (Object o, Throwable e)
void	debug (String comment)
void	debug (String comment, IMediaStream stream)
void	debug (String comment, IMediaStream stream, String category, String event, int status, String context)
void	debug (String comment, String category, String event)
void	debug (String comment, String category, String event, int status, String context)
void	debug (String comment, Throwable e)
void	error (Object o)
void	error (Object o, Throwable e)

void	<u>error</u> (String comment)
void	<u>error</u> (String comment, <u>IMediaStream</u> stream)
void	<u>error</u> (String comment, <u>IMediaStream</u> stream, String category, String event, int status, String context)
void	<u>error</u> (String comment, String category, String event)
void	<u>error</u> (String comment, String category, String event, int status, String context)
void	<u>error</u> (String comment, Throwable e)
void	<u>error</u> (String comment, Throwable e, int stackDepth)
void	<u>fatal</u> (Object o)
void	<u>fatal</u> (Object o, Throwable e)
void	<u>fatal</u> (String comment)
void	<u>fatal</u> (String comment, <u>IMediaStream</u> stream)
void	<u>fatal</u> (String comment, <u>IMediaStream</u> stream, String category, String event, int status, String context)
void	<u>fatal</u> (String comment, String category, String event)
void	<u>fatal</u> (String comment, String category, String event, int status, String context)
void	<u>fatal</u> (String comment, Throwable e)
static <u>WMSLogger</u>	<u>getLogger</u> (String name)
void	<u>info</u> (Object o)
void	<u>info</u> (Object o, Throwable e)
void	<u>info</u> (String comment)
void	<u>info</u> (String comment, <u>IMediaStream</u> stream)
void	<u>info</u> (String comment, <u>IMediaStream</u> stream, String category, String event, int status, String context)
void	<u>info</u> (String comment, String category, String event)

void	<code>info</code> (String comment, String category, String event, int status, String context)
void	<code>info</code> (String comment, Throwable e)
boolean	<code>isDebugEnabled</code> ()
boolean	<code>isEnabledFor</code> (org.apache.log4j.Priority level)
boolean	<code>isErrorEnabled</code> ()
boolean	<code>isInfoEnabled</code> ()
boolean	<code>isTraceEnabled</code> ()
boolean	<code>isWarnEnabled</code> ()
void	<code>log</code> (org.apache.log4j.Level level, String comment)
void	<code>log</code> (org.apache.log4j.Level level, String comment, <code>IMediaStream</code> stream)
void	<code>log</code> (org.apache.log4j.Level level, String comment, <code>IMediaStream</code> stream, String category, String event)
void	<code>log</code> (org.apache.log4j.Level level, String comment, <code>IMediaStream</code> stream, String category, String event, int status, String context)
void	<code>log</code> (org.apache.log4j.Level level, String comment, <code>IMediaStream</code> stream, String category, String event, int status, String context, Throwable e)
void	<code>log</code> (org.apache.log4j.Level level, String comment, <code>IMediaStream</code> stream, String category, String event, int status, String context, Throwable e, int stackDepth)
void	<code>log</code> (org.apache.log4j.Level level, String comment, String category, String event)
void	<code>warn</code> (Object o)
void	<code>warn</code> (Object o, Throwable e)
void	<code>warn</code> (String comment)
void	<code>warn</code> (String comment, <code>IMediaStream</code> stream)
void	<code>warn</code> (String comment, <code>IMediaStream</code> stream, String category, String event, int status, String context)
void	<code>warn</code> (String comment, String category, String event)

void	warn (String comment, String category, String event, int status, String context)
void	warn (String comment, Throwable e)

Methods inherited from class org.apache.log4j.Logger

getLogger, getLogger, getLogger, getLogger, isTraceEnabled, trace, trace

Methods inherited from class org.apache.log4j.Category

addAppender, assertLog, callAppenders, debug, debug, error, error, exists, fatal, fatal, forcedLog, getAdditivity, getAllAppenders, getAppender, getChainedPriority, getCurrentCategories, getDefaultHierarchy, getEffectiveLevel, getHierarchy, getInstance, getInstance, getLevel, getLoggerRepository, getName, getParent, getPriority, getResourceBundle, getResourceBundleString, getRoot, info, info, isAttached, isDebugEnabled, isEnabledFor, isInfoEnabled, l7dlog, l7dlog, log, log, log, removeAllAppenders, removeAppender, removeAppender, setAdditivity, setLevel, setPriority, setResourceBundle, shutdown, warn, warn

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface org.apache.log4j.spi.AppenderAttachable

addAppender, getAllAppenders, getAppender, isAttached, removeAllAppenders, removeAppender, removeAppender

Constructors

WMSLogger

```
public WMSLogger(String name,  
                  org.apache.log4j.Logger tmpLogger)
```

WMSLogger

```
public WMSLogger(String name)
```

Methods

getLogger

```
public static WMSLogger getLogger(String name)
```

(continued from last page)

isWarnEnabled

```
public boolean isWarnEnabled()
```

isErrorEnabled

```
public boolean isErrorEnabled()
```

isTraceEnabled

```
public boolean isTraceEnabled()
```

isDebugEnabled

```
public boolean isDebugEnabled()
```

isInfoEnabled

```
public boolean isInfoEnabled()
```

isEnabledFor

```
public boolean isEnabledFor(org.apache.log4j.Priority level)
```

log

```
public void log(org.apache.log4j.Level level,  
               String comment,  
               IMediaStream stream,  
               String category,  
               String event,  
               int status,  
               String context,  
               Throwable e)
```

log

```
public void log(org.apache.log4j.Level level,  
               String comment,  
               IMediaStream stream,  
               String category,  
               String event,  
               int status,  
               String context,  
               Throwable e,  
               int stackDepth)
```

(continued from last page)

log

```
public void log(org.apache.log4j.Level level,
               String comment,
               IMediaStream stream,
               String category,
               String event,
               int status,
               String context)
```

log

```
public void log(org.apache.log4j.Level level,
               String comment,
               IMediaStream stream,
               String category,
               String event)
```

log

```
public void log(org.apache.log4j.Level level,
               String comment,
               String category,
               String event)
```

log

```
public void log(org.apache.log4j.Level level,
               String comment)
```

log

```
public void log(org.apache.log4j.Level level,
               String comment,
               IMediaStream stream)
```

info

```
public void info(Object o)
```

info

```
public void info(Object o,
               Throwable e)
```

info

```
public void info(String comment)
```

info

```
public void info(String comment,  
                 Throwable e)
```

info

```
public void info(String comment,  
                 IMediaStream stream)
```

info

```
public void info(String comment,  
                 String category,  
                 String event)
```

info

```
public void info(String comment,  
                 String category,  
                 String event,  
                 int status,  
                 String context)
```

info

```
public void info(String comment,  
                 IMediaStream stream,  
                 String category,  
                 String event,  
                 int status,  
                 String context)
```

error

```
public void error(Object o)
```

error

```
public void error(Object o,  
                 Throwable e)
```

(continued from last page)

error

```
public void error(String comment)
```

error

```
public void error(String comment,  
                  Throwable e)
```

error

```
public void error(String comment,  
                  Throwable e,  
                  int stackDepth)
```

error

```
public void error(String comment,  
                  IMediaStream stream)
```

error

```
public void error(String comment,  
                  String category,  
                  String event)
```

error

```
public void error(String comment,  
                  String category,  
                  String event,  
                  int status,  
                  String context)
```

error

```
public void error(String comment,  
                  IMediaStream stream,  
                  String category,  
                  String event,  
                  int status,  
                  String context)
```

(continued from last page)

fatal

```
public void fatal(Object o)
```

fatal

```
public void fatal(Object o,  
                  Throwable e)
```

fatal

```
public void fatal(String comment)
```

fatal

```
public void fatal(String comment,  
                  Throwable e)
```

fatal

```
public void fatal(String comment,  
                  IMediaStream stream)
```

fatal

```
public void fatal(String comment,  
                  String category,  
                  String event)
```

fatal

```
public void fatal(String comment,  
                  String category,  
                  String event,  
                  int status,  
                  String context)
```

fatal

```
public void fatal(String comment,  
                  IMediaStream stream,  
                  String category,  
                  String event,  
                  int status,  
                  String context)
```

debug

```
public void debug(Object o)
```

debug

```
public void debug(Object o,  
                  Throwable e)
```

debug

```
public void debug(String comment)
```

debug

```
public void debug(String comment,  
                  Throwable e)
```

debug

```
public void debug(String comment,  
                  IMediaStream stream)
```

debug

```
public void debug(String comment,  
                  String category,  
                  String event)
```

debug

```
public void debug(String comment,  
                  String category,  
                  String event,  
                  int status,  
                  String context)
```

debug

```
public void debug(String comment,  
                  IMediaStream stream,  
                  String category,  
                  String event,  
                  int status,  
                  String context)
```

(continued from last page)

warn

```
public void warn(Object o)
```

warn

```
public void warn(Object o,  
                Throwable e)
```

warn

```
public void warn(String comment)
```

warn

```
public void warn(String comment,  
                Throwable e)
```

warn

```
public void warn(String comment,  
                IMediaStream stream)
```

warn

```
public void warn(String comment,  
                String category,  
                String event)
```

warn

```
public void warn(String comment,  
                String category,  
                String event,  
                int status,  
                String context)
```

(continued from last page)

warn

```
public void warn(String comment,  
    IMediaStream stream,  
    String category,  
    String event,  
    int status,  
    String context)
```

com.wowza.wms.logging
Class WMSLoggerFactory

java.lang.Object
└─com.wowza.wms.logging.WMSLoggerFactory

All Implemented Interfaces:
org.apache.log4j.spi.LoggerFactory

public class **WMSLoggerFactory**
extends Object
implements org.apache.log4j.spi.LoggerFactory

Field Summary	
public static final	LOGGERNAME_SERVER Value: _Server_

Constructor Summary	
public	WMSLoggerFactory()

Method Summary	
WMSLogger	forceNewLoggerInstance (String name, org.apache.log4j.Logger tmpLogger)
static Object	getGlobalLogValue (String key)
static WMSLoggerFactory	getInstance ()
static WMSLogger	getLogger (Class classObj)
static WMSLogger	getLoggerObj (IApplication application)
static WMSLogger	getLoggerObj (IApplicationInstance appInstance)
static WMSLogger	getLoggerObj (IVHost vhost)
WMSLogger	getLoggerObj (String name)
static void	initContextLogging (IApplication application)
static void	initContextLogging (IApplicationInstance appInstance)
static void	initContextLogging (IVHost vhost)

static WMSLogger	initializeLogging (String loggingConfigURL)
static WMSLogger	initializeLogging (String loggingConfigURL, IVHost vhost)
static boolean	isGlobalLogValueSet (String key)
org.apache.log4j.Logger	makeNewLoggerInstance (String name)
static void	putGlobalLogValue (String key, Object obj)
static void	removeGlobalLogValue (String key)
static void	removeGlobalLogValues ()
static void	resetMDC ()

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface org.apache.log4j.spi.LoggerFactory

makeNewLoggerInstance

Fields

LOGGERNAME_SERVER

```
public static final java.lang.String LOGGERNAME_SERVER
```

Constant value: **_Server_**

Constructors

WMSLoggerFactory

```
public WMSLoggerFactory()
```

Methods

getLoggerObj

```
public WMSLogger getLoggerObj(String name)
```

(continued from last page)

forceNewLoggerInstance

```
public WMSLogger forceNewLoggerInstance(String name,  
    org.apache.log4j.Logger tmpLogger)
```

getInstance

```
public static WMSLoggerFactory getInstance()
```

initContextLogging

```
public static void initContextLogging(IVHost vhost)
```

initContextLogging

```
public static void initContextLogging(IApplication application)
```

initContextLogging

```
public static void initContextLogging(IApplicationInstance appInstance)
```

initializeLogging

```
public static WMSLogger initializeLogging(String loggingConfigURL)
```

initializeLogging

```
public static WMSLogger initializeLogging(String loggingConfigURL,  
    IVHost vhost)
```

makeNewLoggerInstance

```
public org.apache.log4j.Logger makeNewLoggerInstance(String name)
```

resetMDC

```
public static void resetMDC()
```

(continued from last page)

removeGlobalLogValues

```
public static void removeGlobalLogValues()
```

removeGlobalLogValue

```
public static void removeGlobalLogValue(String key)
```

isGlobalLogValueSet

```
public static boolean isGlobalLogValueSet(String key)
```

putGlobalLogValue

```
public static void putGlobalLogValue(String key,  
                                     Object obj)
```

getGlobalLogValue

```
public static Object getGlobalLogValue(String key)
```

getLoggerObj

```
public static WMSLogger getLoggerObj(IVHost vhost)
```

getLoggerObj

```
public static WMSLogger getLoggerObj(IApplication application)
```

getLoggerObj

```
public static WMSLogger getLoggerObj(IApplicationInstance appInstance)
```

getLogger

```
public static WMSLogger getLogger(Class classObj)
```

com.wowza.wms.logging

Class WMSLoggerIDs

java.lang.Object

└-com.wowza.wms.logging.WMSLoggerIDs

public class **WMSLoggerIDs**
extends Object

Field Summary

public static final	CAT_ALL
public static final	CAT_application Value: application
public static final	CAT_cupertino Value: cupertino
public static final	CAT_dvrchunk Value: dvrchunk
public static final	CAT_mpegdash Value: mpegdash
public static final	CAT_rtsp Value: rtsp
public static final	CAT_sanjose Value: sanjose
public static final	CAT_server Value: server
public static final	CAT_session Value: session
public static final	CAT_smoothstreaming Value: smoothstreaming
public static final	CAT_stream Value: stream
public static final	CAT_transcoder Value: transcoder

public static final	CAT_vhost Value: vhost
public static final	CAT_webm Value: webm
public static final	CTRL_playlist_node Value: CTRL-playlist-node
public static final	EVT_ALL
public static final	EVT_announce Value: announce
public static final	EVT_app_start Value: app-start
public static final	EVT_app_stop Value: app-stop
public static final	EVT_comment Value: comment
public static final	EVT_connect Value: connect
public static final	EVT_connect_burst Value: connect-burst
public static final	EVT_connect_pending Value: connect-pending
public static final	EVT_create Value: create
public static final	EVT_decoderaudiostart Value: decoder-audio-start
public static final	EVT_decoderaudiostop Value: decoder-audio-stop
public static final	EVT_decodervideostart Value: decoder-video-start
public static final	EVT_decodervideostop Value: decoder-video-stop

public static final	EVT_describe Value: describe
public static final	EVT_destroy Value: destroy
public static final	EVT_disconnect Value: disconnect
public static final	EVT_encoderaudiostart Value: encoder-audio-start
public static final	EVT_encoderaudiostop Value: encoder-audio-stop
public static final	EVT_encodervideostart Value: encoder-video-start
public static final	EVT_encodervideostop Value: encoder-video-stop
public static final	EVT_pause Value: pause
public static final	EVT_play Value: play
public static final	EVT_publish Value: publish
public static final	EVT_record Value: record
public static final	EVT_recordstop Value: recordstop
public static final	EVT_seek Value: seek
public static final	EVT_server_start Value: server-start
public static final	EVT_server_stop Value: server-stop
public static final	EVT_setbuffertime Value: setbuffertime

public static final	EVT_setstreamtype Value: setstreamtype
public static final	EVT_stop Value: stop
public static final	EVT_unpause Value: unpause
public static final	EVT_unpublish Value: unpublish
public static final	EVT_vhost_start Value: vhost-start
public static final	EVT_vhost_stop Value: vhost-stop
public static final	FD_ALL
public static final	FD_c_client_id Value: c-client-id
public static final	FD_c_ip Value: c-ip
public static final	FD_c_proto Value: c-proto
public static final	FD_c_referrer Value: c-referrer
public static final	FD_c_user_agent Value: c-user-agent
public static final	FD_cs_bytes Value: cs-bytes
public static final	FD_cs_stream_bytes Value: cs-stream-bytes
public static final	FD_cs_uri_query Value: cs-uri-query
public static final	FD_cs_uri_stem Value: cs-uri-stem

public static final	FD_date Value: date
public static final	FD_s_ip Value: s-ip
public static final	FD_s_port Value: s-port
public static final	FD_s_uri Value: s-uri
public static final	FD_sc_bytes Value: sc-bytes
public static final	FD_sc_stream_bytes Value: sc-stream-bytes
public static final	FD_time Value: time
public static final	FD_tz Value: tz
public static final	FD_x_app Value: x-app
public static final	FD_x_appinst Value: x-appinst
public static final	FD_x_category Value: x-category
public static final	FD_x_comment Value: x-comment
public static final	FD_x_ctx Value: x-ctx
public static final	FD_x_ctx_override Value: x-ctx-override
public static final	FD_x_duration Value: x-duration
public static final	FD_x_event Value: x-event

public static final	FD_x_file_ext Value: x-file-ext
public static final	FD_x_file_length Value: x-file-length
public static final	FD_x_file_name Value: x-file-name
public static final	FD_x_file_size Value: x-file-size
public static final	FD_x_severity Value: x-severity
public static final	FD_x_sname Value: x-sname
public static final	FD_x_sname_query Value: x-sname-query
public static final	FD_x_spos Value: x-spos
public static final	FD_x_status Value: x-status
public static final	FD_x_stream_id Value: x-stream-id
public static final	FD_x_suri Value: x-suri
public static final	FD_x_suri_query Value: x-suri-query
public static final	FD_x_suri_stem Value: x-suri-stem
public static final	FD_x_vhost Value: x-vhost
public static final	PROTO_HTTPCUPERTINO Value: http (cupertino)
public static final	PROTO_HTTPDVRCHUNK Value: http (dvr)

public static final	PROTO_HTTPMPEGDASH Value: http (mpegdash)
public static final	PROTO_HTTPSANJOSE Value: http (sanjose)
public static final	PROTO_HTTPSCUPERTINO Value: https (cupertino)
public static final	PROTO_HTTPSDVRCHUNK Value: https (dvr)
public static final	PROTO_HTTPSMOOTH Value: http (smooth)
public static final	PROTO_HTTPSSANJOSE Value: https (sanjose)
public static final	PROTO_HTTPSMOOTH Value: https (smooth)
public static final	PROTO_HTTPSTREAMER Value: https (streamer)
public static final	PROTO_HTTPSTREAMER Value: http (streamer)
public static final	PROTO_RTMP Value: rtmp
public static final	PROTO_RTMPE Value: rtmpe
public static final	PROTO_RTMPs Value: rtmps
public static final	PROTO_RTMPt Value: rtmpt (HTTP-1.1)
public static final	PROTO_RTMPtE Value: rtmpte (HTTP-1.1)
public static final	PROTO_RTMPts Value: rtmpts (HTTP-1.1)
public static final	PROTO_RTSP Value: rtsp

public static final	PROTO_WOWZ Value: wowz
public static final	PROTO_WOWZE Value: wowze
public static final	PROTO_WOWZS Value: wowzs
public static final	STAT_connect_application_not_available Value: 302
public static final	STAT_connect_application_not_found Value: 404
public static final	STAT_connect_bad_gateway Value: 502
public static final	STAT_connect_internal_error Value: 500
public static final	STAT_connect_license_limit Value: 413
public static final	STAT_connect_pending_wating Value: 100
public static final	STAT_connect_redirect Value: 302
public static final	STAT_connect_rejected_by_application Value: 401
public static final	STAT_connect_rejected_by_module Value: 403
public static final	STAT_connect_resource_limit Value: 409
public static final	STAT_connect_service_unavailable Value: 503
public static final	STAT_connect_successful Value: 200
public static final	STAT_connect_unknown_protocol Value: 400

public static final	<u>STAT_general_internal_error</u> Value: 500
public static final	<u>STAT_general_successful</u> Value: 200
public static final	<u>STAT_play_bad_request</u> Value: 400
public static final	<u>STAT_play_internal_error</u> Value: 500
public static final	<u>STAT_play_rejected_by_application</u> Value: 401
public static final	<u>STAT_play_rejected_by_module</u> Value: 403
public static final	<u>STAT_play_stream_not_found</u> Value: 404
public static final	<u>STAT_play_successful</u> Value: 200
public static final	<u>STAT_play_unsupported_media_type</u> Value: 415
public static final	<u>STAT_publish_bad_request</u> Value: 400
public static final	<u>STAT_publish_in_use</u> Value: 409
public static final	<u>STAT_publish_internal_error</u> Value: 500
public static final	<u>STAT_publish_rejected_by_application</u> Value: 401
public static final	<u>STAT_publish_successful</u> Value: 200
public static final	<u>STAT_publish_unsupported_media_type</u> Value: 415
public static final	<u>STAT_stop_client_disconnect</u> Value: 408

```
public static final
```

```
STAT\_stop\_successful
```

```
Value: 200
```

Constructor Summary

```
public
```

```
WMSLoggerIDs( )
```

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

FD_x_event

```
public static final java.lang.String FD_x_event
```

Constant value: **x-event**

FD_x_category

```
public static final java.lang.String FD_x_category
```

Constant value: **x-category**

FD_date

```
public static final java.lang.String FD_date
```

Constant value: **date**

FD_time

```
public static final java.lang.String FD_time
```

Constant value: **time**

FD_tz

```
public static final java.lang.String FD_tz
```

Constant value: **tz**

FD_x_ctx

```
public static final java.lang.String FD_x_ctx
```

Constant value: **x-ctx**

FD_x_ctx_override

```
public static final java.lang.String FD_x_ctx_override
```

Constant value: **x-ctx-override**

FD_x_vhost

```
public static final java.lang.String FD_x_vhost
```

Constant value: **x-vhost**

FD_x_app

```
public static final java.lang.String FD_x_app
```

Constant value: **x-app**

FD_x_appinst

```
public static final java.lang.String FD_x_appinst
```

Constant value: **x-appinst**

FD_c_ip

```
public static final java.lang.String FD_c_ip
```

Constant value: **c-ip**

FD_c_proto

```
public static final java.lang.String FD_c_proto
```

Constant value: **c-proto**

FD_s_uri

```
public static final java.lang.String FD_s_uri
```

Constant value: **s-uri**

FD_c_referrer

```
public static final java.lang.String FD_c_referrer
```

Constant value: **c-referrer**

(continued from last page)

FD_c_user_agent

```
public static final java.lang.String FD_c_user_agent
```

Constant value: **c-user-agent**

FD_c_client_id

```
public static final java.lang.String FD_c_client_id
```

Constant value: **c-client-id**

FD_cs_bytes

```
public static final java.lang.String FD_cs_bytes
```

Constant value: **cs-bytes**

FD_sc_bytes

```
public static final java.lang.String FD_sc_bytes
```

Constant value: **sc-bytes**

FD_x_sname

```
public static final java.lang.String FD_x_sname
```

Constant value: **x-sname**

FD_x_file_size

```
public static final java.lang.String FD_x_file_size
```

Constant value: **x-file-size**

FD_x_file_length

```
public static final java.lang.String FD_x_file_length
```

Constant value: **x-file-length**

FD_x_spos

```
public static final java.lang.String FD_x_spos
```

Constant value: **x-spos**

FD_cs_stream_bytes

```
public static final java.lang.String FD_cs_stream_bytes
```

(continued from last page)

Constant value: **cs-stream-bytes**

FD_sc_stream_bytes

```
public static final java.lang.String FD_sc_stream_bytes
```

Constant value: **sc-stream-bytes**

FD_s_ip

```
public static final java.lang.String FD_s_ip
```

Constant value: **s-ip**

FD_x_duration

```
public static final java.lang.String FD_x_duration
```

Constant value: **x-duration**

FD_x_status

```
public static final java.lang.String FD_x_status
```

Constant value: **x-status**

FD_cs_uri_stem

```
public static final java.lang.String FD_cs_uri_stem
```

Constant value: **cs-uri-stem**

FD_cs_uri_query

```
public static final java.lang.String FD_cs_uri_query
```

Constant value: **cs-uri-query**

FD_x_sname_query

```
public static final java.lang.String FD_x_sname_query
```

Constant value: **x-sname-query**

FD_x_file_name

```
public static final java.lang.String FD_x_file_name
```

Constant value: **x-file-name**

FD_x_file_ext

```
public static final java.lang.String FD_x_file_ext
```

Constant value: **x-file-ext**

FD_x_suri_query

```
public static final java.lang.String FD_x_suri_query
```

Constant value: **x-suri-query**

FD_x_suri_stem

```
public static final java.lang.String FD_x_suri_stem
```

Constant value: **x-suri-stem**

FD_x_suri

```
public static final java.lang.String FD_x_suri
```

Constant value: **x-suri**

FD_x_severity

```
public static final java.lang.String FD_x_severity
```

Constant value: **x-severity**

FD_x_comment

```
public static final java.lang.String FD_x_comment
```

Constant value: **x-comment**

FD_s_port

```
public static final java.lang.String FD_s_port
```

Constant value: **s-port**

FD_x_stream_id

```
public static final java.lang.String FD_x_stream_id
```

Constant value: **x-stream-id**

(continued from last page)

CAT_server

```
public static final java.lang.String CAT_server
```

Constant value: **server**

CAT_vhost

```
public static final java.lang.String CAT_vhost
```

Constant value: **vhost**

CAT_application

```
public static final java.lang.String CAT_application
```

Constant value: **application**

CAT_session

```
public static final java.lang.String CAT_session
```

Constant value: **session**

CAT_stream

```
public static final java.lang.String CAT_stream
```

Constant value: **stream**

CAT_rtsp

```
public static final java.lang.String CAT_rtsp
```

Constant value: **rtsp**

CAT_cupertino

```
public static final java.lang.String CAT_cupertino
```

Constant value: **cupertino**

CAT_sanjose

```
public static final java.lang.String CAT_sanjose
```

Constant value: **sanjose**

CAT_smoothstreaming

```
public static final java.lang.String CAT_smoothstreaming
```

(continued from last page)

Constant value: **smoothstreaming**

CAT_dvrchunk

```
public static final java.lang.String CAT_dvrchunk
```

Constant value: **dvrchunk**

CAT_webm

```
public static final java.lang.String CAT_webm
```

Constant value: **webm**

CAT_transcoder

```
public static final java.lang.String CAT_transcoder
```

Constant value: **transcoder**

CAT_mpegdash

```
public static final java.lang.String CAT_mpegdash
```

Constant value: **mpegdash**

EVT_connect_pending

```
public static final java.lang.String EVT_connect_pending
```

Constant value: **connect-pending**

EVT_connect

```
public static final java.lang.String EVT_connect
```

Constant value: **connect**

EVT_connect_burst

```
public static final java.lang.String EVT_connect_burst
```

Constant value: **connect-burst**

EVT_disconnect

```
public static final java.lang.String EVT_disconnect
```

Constant value: **disconnect**

EVT_publish

```
public static final java.lang.String EVT_publish
```

Constant value: **publish**

EVT_unpublish

```
public static final java.lang.String EVT_unpublish
```

Constant value: **unpublish**

EVT_play

```
public static final java.lang.String EVT_play
```

Constant value: **play**

EVT_pause

```
public static final java.lang.String EVT_pause
```

Constant value: **pause**

EVT_setbuffertime

```
public static final java.lang.String EVT_setbuffertime
```

Constant value: **setbuffertime**

EVT_create

```
public static final java.lang.String EVT_create
```

Constant value: **create**

EVT_destroy

```
public static final java.lang.String EVT_destroy
```

Constant value: **destroy**

EVT_setstreamtype

```
public static final java.lang.String EVT_setstreamtype
```

Constant value: **setstreamtype**

(continued from last page)

EVT_unpause

```
public static final java.lang.String EVT_unpause
```

Constant value: **unpause**

EVT_seek

```
public static final java.lang.String EVT_seek
```

Constant value: **seek**

EVT_stop

```
public static final java.lang.String EVT_stop
```

Constant value: **stop**

EVT_record

```
public static final java.lang.String EVT_record
```

Constant value: **record**

EVT_recordstop

```
public static final java.lang.String EVT_recordstop
```

Constant value: **recordstop**

EVT_server_start

```
public static final java.lang.String EVT_server_start
```

Constant value: **server-start**

EVT_server_stop

```
public static final java.lang.String EVT_server_stop
```

Constant value: **server-stop**

EVT_vhost_start

```
public static final java.lang.String EVT_vhost_start
```

Constant value: **vhost-start**

EVT_vhost_stop

```
public static final java.lang.String EVT_vhost_stop
```

(continued from last page)

Constant value: **vhost-stop**

EVT_app_start

```
public static final java.lang.String EVT_app_start
```

Constant value: **app-start**

EVT_app_stop

```
public static final java.lang.String EVT_app_stop
```

Constant value: **app-stop**

EVT_comment

```
public static final java.lang.String EVT_comment
```

Constant value: **comment**

EVT_announce

```
public static final java.lang.String EVT_announce
```

Constant value: **announce**

EVT_describe

```
public static final java.lang.String EVT_describe
```

Constant value: **describe**

EVT_decoderaudiostart

```
public static final java.lang.String EVT_decoderaudiostart
```

Constant value: **decoder-audio-start**

EVT_decoderaudiostop

```
public static final java.lang.String EVT_decoderaudiostop
```

Constant value: **decoder-audio-stop**

EVT_decodervideostart

```
public static final java.lang.String EVT_decodervideostart
```

Constant value: **decoder-video-start**

EVT_decodervideostop

```
public static final java.lang.String EVT_decodervideostop
```

Constant value: **decoder-video-stop**

EVT_encoderaudiostart

```
public static final java.lang.String EVT_encoderaudiostart
```

Constant value: **encoder-audio-start**

EVT_encoderaudiostop

```
public static final java.lang.String EVT_encoderaudiostop
```

Constant value: **encoder-audio-stop**

EVT_encodervideostart

```
public static final java.lang.String EVT_encodervideostart
```

Constant value: **encoder-video-start**

EVT_encodervideostop

```
public static final java.lang.String EVT_encodervideostop
```

Constant value: **encoder-video-stop**

CTRL_playlist_node

```
public static final java.lang.String CTRL_playlist_node
```

Constant value: **CTRL-playlist-node**

STAT_connect_pending_wating

```
public static final int STAT_connect_pending_wating
```

Constant value: **100**

STAT_connect_successful

```
public static final int STAT_connect_successful
```

Constant value: **200**

(continued from last page)

STAT_connect_application_not_available

```
public static final int STAT_connect_application_not_available
```

Constant value: **302**

STAT_connect_unknown_protocol

```
public static final int STAT_connect_unknown_protocol
```

Constant value: **400**

STAT_connect_rejected_by_application

```
public static final int STAT_connect_rejected_by_application
```

Constant value: **401**

STAT_connect_rejected_by_module

```
public static final int STAT_connect_rejected_by_module
```

Constant value: **403**

STAT_connect_application_not_found

```
public static final int STAT_connect_application_not_found
```

Constant value: **404**

STAT_connect_resource_limit

```
public static final int STAT_connect_resource_limit
```

Constant value: **409**

STAT_connect_license_limit

```
public static final int STAT_connect_license_limit
```

Constant value: **413**

STAT_connect_redirect

```
public static final int STAT_connect_redirect
```

Constant value: **302**

STAT_connect_internal_error

```
public static final int STAT_connect_internal_error
```

(continued from last page)

Constant value: **500**

STAT_connect_bad_gateway

```
public static final int STAT_connect_bad_gateway
```

Constant value: **502**

STAT_connect_service_unavailable

```
public static final int STAT_connect_service_unavailable
```

Constant value: **503**

STAT_play_successful

```
public static final int STAT_play_successful
```

Constant value: **200**

STAT_play_bad_request

```
public static final int STAT_play_bad_request
```

Constant value: **400**

STAT_play_rejected_by_application

```
public static final int STAT_play_rejected_by_application
```

Constant value: **401**

STAT_play_rejected_by_module

```
public static final int STAT_play_rejected_by_module
```

Constant value: **403**

STAT_play_stream_not_found

```
public static final int STAT_play_stream_not_found
```

Constant value: **404**

STAT_play_unsupported_media_type

```
public static final int STAT_play_unsupported_media_type
```

Constant value: **415**

STAT_play_internal_error

public static final int **STAT_play_internal_error**

Constant value: **500**

STAT_publish_successful

public static final int **STAT_publish_successful**

Constant value: **200**

STAT_publish_bad_request

public static final int **STAT_publish_bad_request**

Constant value: **400**

STAT_publish_rejected_by_application

public static final int **STAT_publish_rejected_by_application**

Constant value: **401**

STAT_publish_in_use

public static final int **STAT_publish_in_use**

Constant value: **409**

STAT_publish_unsupported_media_type

public static final int **STAT_publish_unsupported_media_type**

Constant value: **415**

STAT_publish_internal_error

public static final int **STAT_publish_internal_error**

Constant value: **500**

STAT_stop_successful

public static final int **STAT_stop_successful**

Constant value: **200**

(continued from last page)

STAT_stop_client_disconnect

```
public static final int STAT_stop_client_disconnect
```

Constant value: **408**

STAT_general_successful

```
public static final int STAT_general_successful
```

Constant value: **200**

STAT_general_internal_error

```
public static final int STAT_general_internal_error
```

Constant value: **500**

PROTO_RTMP

```
public static final java.lang.String PROTO_RTMP
```

Constant value: **rtmp**

PROTO_RTMPs

```
public static final java.lang.String PROTO_RTMPs
```

Constant value: **rtmps**

PROTO_RTMPT

```
public static final java.lang.String PROTO_RTMPT
```

Constant value: **rtmpt (HTTP-1.1)**

PROTO_RTMPTs

```
public static final java.lang.String PROTO_RTMPTs
```

Constant value: **rtmpts (HTTP-1.1)**

PROTO_RTMPE

```
public static final java.lang.String PROTO_RTMPE
```

Constant value: **rtmpe**

PROTO_RTMPTE

```
public static final java.lang.String PROTO_RTMPTE
```

(continued from last page)

Constant value: **rtmpte (HTTP-1.1)**

PROTO_WOWZ

```
public static final java.lang.String PROTO_WOWZ
```

Constant value: **wowz**

PROTO_WOWZE

```
public static final java.lang.String PROTO_WOWZE
```

Constant value: **wowze**

PROTO_WOWZS

```
public static final java.lang.String PROTO_WOWZS
```

Constant value: **wowzs**

PROTO_RTSP

```
public static final java.lang.String PROTO_RTSP
```

Constant value: **rtsp**

PROTO_HTTPSTREAMER

```
public static final java.lang.String PROTO_HTTPSTREAMER
```

Constant value: **http (streamer)**

PROTO_HTTPCUPERTINO

```
public static final java.lang.String PROTO_HTTPCUPERTINO
```

Constant value: **http (cupertino)**

PROTO_HTTPSMOOTH

```
public static final java.lang.String PROTO_HTTPSMOOTH
```

Constant value: **http (smooth)**

PROTO_HTTPSANJOSE

```
public static final java.lang.String PROTO_HTTPSANJOSE
```

Constant value: **http (sanjose)**

PROTO_HTTPDVARCHUNK

public static final java.lang.String **PROTO_HTTPDVARCHUNK**

Constant value: **http (dvr)**

PROTO_HTTPMPEGDASH

public static final java.lang.String **PROTO_HTTPMPEGDASH**

Constant value: **http (mpegdash)**

PROTO_HTTPSSTREAMER

public static final java.lang.String **PROTO_HTTPSSTREAMER**

Constant value: **https (streamer)**

PROTO_HTTPSCUPERTINO

public static final java.lang.String **PROTO_HTTPSCUPERTINO**

Constant value: **https (cupertino)**

PROTO_HTTPSSMOOTH

public static final java.lang.String **PROTO_HTTPSSMOOTH**

Constant value: **https (smooth)**

PROTO_HTTPSSANJOSE

public static final java.lang.String **PROTO_HTTPSSANJOSE**

Constant value: **https (sanjose)**

PROTO_HTTPSDVARCHUNK

public static final java.lang.String **PROTO_HTTPSDVARCHUNK**

Constant value: **https (dvr)**

FD_ALL

public static final java.lang.String **FD_ALL**

CAT_ALL

public static final java.lang.String **CAT_ALL**

(continued from last page)

EVT_ALL

```
public static final java.lang.String EVT_ALL
```

Constructors

WMSLoggerIDs

```
public WMSLoggerIDs()
```

Package

com.wowza.wms.media.mp3.model.idtags

com.wowza.wms.media.mp3.model.idtags Class ID3DeserializeFrameResult

java.lang.Object

└─com.wowza.wms.media.mp3.model.idtags.ID3DeserializeFrameResult

public class **ID3DeserializeFrameResult**
extends Object

Field Summary

protected	frame
protected	len

Constructor Summary

public	ID3DeserializeFrameResult()
--------	---

Method Summary

IID3V2Frame	getFrame()
int	getLen()
void	setFrame(IID3V2Frame frame)
void	setLen(int len)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

frame

protected com.wowza.wms.media.mp3.model.idtags.IID3V2Frame **frame**

len

protected int **len**

Constructors

ID3DeserializeFrameResult

```
public ID3DeserializeFrameResult()
```

Methods

getFrame

```
public IID3V2Frame getFrame()
```

setFrame

```
public void setFrame(IID3V2Frame frame)
```

getLen

```
public int getLen()
```

setLen

```
public void setLen(int len)
```

com.wowza.wms.media.mp3.model.idtags Class ID3DeserializeFramesResult

java.lang.Object

└-com.wowza.wms.media.mp3.model.idtags.ID3DeserializeFramesResult

public class **ID3DeserializeFramesResult**
extends Object

Field Summary

protected	frames
protected	len

Constructor Summary

public	ID3DeserializeFramesResult()
--------	--

Method Summary

ID3Frames	getFrames()
int	getLen()
void	setFrames(ID3Frames frames)
void	setLen(int len)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

frames

protected com.wowza.wms.media.mp3.model.idtags.ID3Frames **frames**

len

protected int **len**

Constructors

ID3DeserializeFramesResult

```
public ID3DeserializeFramesResult()
```

Methods

getFrames

```
public ID3Frames getFrames()
```

setFrames

```
public void setFrames(ID3Frames frames)
```

getLen

```
public int getLen()
```

setLen

```
public void setLen(int len)
```


com.wowza.wms.media.mp3.model.idtags Class ID3Frames

java.lang.Object

└--com.wowza.wms.media.mp3.model.idtags.ID3Frames

public class **ID3Frames**
extends Object

Field Summary

public static final	<u>ID3FOOTER_SIZE</u> Value: 10
public static final	<u>ID3HEADER_SIZE</u> Value: 10
public static final	<u>ID3HEADER_VERSION</u> Value: 1024
public static final	<u>ID3HEADERFLAGS_DEFAULT</u> Value: 0
public static final	<u>ID3HEADERFLAGS_EXPERIMENTAL</u> Value: 32
public static final	<u>ID3HEADERFLAGS_EXTENDED</u> Value: 64
public static final	<u>ID3HEADERFLAGS_FOOTERPRESENT</u> Value: 16
public static final	<u>ID3HEADERFLAGS_UNSYNC</u> Value: 128
public static final	<u>STARTCODE_3DI</u>
public static final	<u>STARTCODE_ID3</u>

Constructor Summary

public	<u>ID3Frames</u> ()
--------	--------------------------------------

Method Summary

void	<u>clear()</u>
static <u>ID3DeserializeFramesResult</u>	<u>deserialize</u> (byte[] buffer, int offset, int len)
static int	<u>deserializeGetSize</u> (byte[] buffer, int offset, int len)
java.util.List	<u>getFrameMapIds</u> ()
java.util.List	<u>getFrames</u> ()
java.util.List	<u>getFramesById</u> (String idStr)
Object	<u>getLock</u> ()
int	<u>getSize</u> ()
boolean	<u>isEmpty</u> ()
static boolean	<u>isID3Footer</u> (byte[] buffer, int offset, int len)
static boolean	<u>isID3Header</u> (byte[] buffer, int offset, int len)
void	<u>putFrame</u> (<u>IID3V2Frame</u> frame)
<u>IID3V2Frame</u>	<u>removeFrame</u> (<u>IID3V2Frame</u> frame)
byte[]	<u>serialize</u> ()
byte[]	<u>serialize</u> (boolean includeHeader, boolean includeFooter, int flags)
static int	<u>serializeFooter</u> (byte[] buffer, int offset, int flags, int size)
static int	<u>serializeHeader</u> (byte[] buffer, int offset, int flags, int size)
int	<u>serializeTags</u> (byte[] buffer, int offset)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

STARTCODE_ID3

public static final byte **STARTCODE_ID3**

(continued from last page)

STARTCODE_3DI

```
public static final byte STARTCODE_3DI
```

ID3HEADER_VERSION

```
public static final int ID3HEADER_VERSION
```

Constant value: **1024**

ID3HEADER_SIZE

```
public static final int ID3HEADER_SIZE
```

Constant value: **10**

ID3FOOTER_SIZE

```
public static final int ID3FOOTER_SIZE
```

Constant value: **10**

ID3HEADERFLAGS_DEFAULT

```
public static final int ID3HEADERFLAGS_DEFAULT
```

Constant value: **0**

ID3HEADERFLAGS_UNSYNC

```
public static final int ID3HEADERFLAGS_UNSYNC
```

Constant value: **128**

ID3HEADERFLAGS_EXTENDED

```
public static final int ID3HEADERFLAGS_EXTENDED
```

Constant value: **64**

ID3HEADERFLAGS_EXPERIMENTAL

```
public static final int ID3HEADERFLAGS_EXPERIMENTAL
```

Constant value: **32**

(continued from last page)

ID3HEADERFLAGS_FOOTERPRESENT

```
public static final int ID3HEADERFLAGS_FOOTERPRESENT
```

Constant value: **16**

Constructors

ID3Frames

```
public ID3Frames()
```

Methods

getLock

```
public Object getLock()
```

clear

```
public void clear()
```

isEmpty

```
public boolean isEmpty()
```

putFrame

```
public void putFrame(IID3V2Frame frame)
```

removeFrame

```
public IID3V2Frame removeFrame(IID3V2Frame frame)
```

getFrames

```
public java.util.List getFrames()
```

getFrameMapIds

```
public java.util.List getFrameMapIds()
```

getFramesById

```
public java.util.List getFramesById(String idStr)
```

getSize

```
public int getSize()
```

serializeTags

```
public int serializeTags(byte[] buffer,  
    int offset)
```

serializeHeader

```
public static int serializeHeader(byte[] buffer,  
    int offset,  
    int flags,  
    int size)
```

serializeFooter

```
public static int serializeFooter(byte[] buffer,  
    int offset,  
    int flags,  
    int size)
```

serialize

```
public byte[] serialize()
```

serialize

```
public byte[] serialize(boolean includeHeader,  
    boolean includeFooter,  
    int flags)
```

isID3Header

```
public static boolean isID3Header(byte[] buffer,  
    int offset,  
    int len)
```

isID3Footer

```
public static boolean isID3Footer(byte[] buffer,  
    int offset,  
    int len)
```

deserializeGetSize

```
public static int deserializeGetSize(byte[] buffer,  
    int offset,  
    int len)
```

deserialize

```
public static ID3DeserializeFramesResult deserialize(byte[] buffer,  
    int offset,  
    int len)
```

com.wowza.wms.media.mp3.model.idtags Class ID3V2FrameAttachedPicture

java.lang.Object

```

+--com.wowza.wms.media.mp3.model.idtags.ID3V2FrameBase
    +--com.wowza.wms.media.mp3.model.idtags.ID3V2FrameAttachedPicture
  
```

All Implemented Interfaces:

[IID3V2Frame](#)

```

public class ID3V2FrameAttachedPicture
extends ID3V2FrameBase
  
```

Field Summary

public static final	MIMETYPES_JPEG Value: image/jpeg
public static final	MIMETYPES_PNG Value: image/png
public static final	MIMETYPES_URL Value: -->
public static	PICTUREMAXFILESIZE
public static final	PICTURETYPE_ARTISTLOGO Value: 19
public static final	PICTURETYPE_COVERBACK Value: 4
public static final	PICTURETYPE_COVERFRONT Value: 3
public static final	PICTURETYPE_FILEICON Value: 1
public static final	PICTURETYPE_ILLUSTRATION Value: 18
public static final	PICTURETYPE_MOVIESCREENCAPTURE Value: 16
public static final	PICTURETYPE_OTHER Value: 0

public static final	PICTURETYPE_OTHERFILEICON Value: 2
public static final	PICTURETYPE_PUBLISHERLOGO Value: 20

Fields inherited from class [com.wowza.wms.media.mp3.model.idtags.ID3V2FrameBase](#)

[DEFAULT_LANGUAGE](#), [FLAGS_DEFAULT](#), [HEADERSIZE](#), [TAG_APIC](#), [TAG_COMM](#), [TAG_LINK](#), [TAG_PRIV](#), [TAG_RBUF](#), [TAG_SYLT](#), [TAG_TALB](#), [TAG_TBPM](#), [TAG_TCOM](#), [TAG_TCON](#), [TAG_TCOP](#), [TAG_TDEN](#), [TAG_TDLY](#), [TAG_TDOR](#), [TAG_TDRC](#), [TAG_TDRL](#), [TAG_TDTG](#), [TAG_TENC](#), [TAG_TEXT](#), [TAG_TFLT](#), [TAG_TIPL](#), [TAG_TIT1](#), [TAG_TIT2](#), [TAG_TIT3](#), [TAG_TKEY](#), [TAG_TLAN](#), [TAG_TLEN](#), [TAG_TMCL](#), [TAG_TMED](#), [TAG_TMOO](#), [TAG_TOAL](#), [TAG_TOFN](#), [TAG_TOLY](#), [TAG_TOPE](#), [TAG_TOWN](#), [TAG_TPE1](#), [TAG_TPE2](#), [TAG_TPE3](#), [TAG_TPE4](#), [TAG_TPOS](#), [TAG_TPRO](#), [TAG_TPUB](#), [TAG_TRCK](#), [TAG_TRSN](#), [TAG_TRSO](#), [TAG_TSOA](#), [TAG_TSOP](#), [TAG_TSOT](#), [TAG_TSRC](#), [TAG_TSSE](#), [TAG_TSST](#), [TAG_TXXX](#), [TAG_UNKN](#), [TAG_WCOM](#), [TAG_WCOP](#), [TAG_WOAF](#), [TAG_WOAR](#), [TAG_WOAS](#), [TAG_WORS](#), [TAG_WPAY](#), [TAG_WPUB](#), [TAG_WXXX](#), [TAGS_TEXTINFORMATION](#), [TEXTENCODING_DEFAULT](#), [TEXTENCODING_ISO_8859_1](#), [TEXTENCODING_URL](#), [TEXTENCODING_UTF16](#), [TEXTENCODING_UTF16BE](#), [TEXTENCODING_UTF8](#)

Constructor Summary

public	ID3V2FrameAttachedPicture (String idStr, int flags)
public	ID3V2FrameAttachedPicture ()

Method Summary

void	deserializeBody (byte[] buffer, int offset, int len)
int	getBodySize ()
String	getDescription ()
String	getMapIdStr ()
String	getMimeType ()
byte[]	getPictureData ()
int	getPictureType ()
int	getTextEncoding ()
boolean	loadFile (java.io.File file)
int	serializeBody (byte[] buffer, int offset)
void	setDescription (String description)
void	setMimeType (String mimeType)

void	setPictureData (byte[] pictureData)
void	setPictureDataAsURL (String urlStr)
void	setPictureType (int pictureType)
void	setTextEncoding (int textEncoding)

Methods inherited from class [com.wowza.wms.media.mp3.model.idtags.ID3V2FrameBase](#)

[byteStringLen](#), [byteStringLen](#), [deserializeFrame](#), [deserializeString](#), [deserializeString](#), [getFlags](#), [getIdStr](#), [getMapIdStr](#), [getSize](#), [serialize](#), [serializeString](#), [serializeString](#), [serializeStringLen](#), [serializeStringLen](#), [setFlags](#), [setIdStr](#), [textEncodingToCharsetName](#), [trimTrailingZero](#)

Methods inherited from class java.lang.Object

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

Methods inherited from interface [com.wowza.wms.media.mp3.model.idtags.IID3V2Frame](#)

[deserializeBody](#), [getBodySize](#), [getFlags](#), [getIdStr](#), [getMapIdStr](#), [getSize](#), [serialize](#), [serializeBody](#), [setFlags](#), [setIdStr](#)

Fields

PICTURETYPE_OTHER

```
public static final int PICTURETYPE_OTHER
```

Constant value: **0**

PICTURETYPE_FILEICON

```
public static final int PICTURETYPE_FILEICON
```

Constant value: **1**

PICTURETYPE_OTHERFILEICON

```
public static final int PICTURETYPE_OTHERFILEICON
```

Constant value: **2**

PICTURETYPE_COVERFRONT

```
public static final int PICTURETYPE_COVERFRONT
```

Constant value: **3**

PICTURETYPE_COVERBACK

```
public static final int PICTURETYPE_COVERBACK
```

Constant value: **4**

PICTURETYPE_MOVIESCREENCAPTURE

```
public static final int PICTURETYPE_MOVIESCREENCAPTURE
```

Constant value: **16**

PICTURETYPE_ILLUSTRATION

```
public static final int PICTURETYPE_ILLUSTRATION
```

Constant value: **18**

PICTURETYPE_ARTISTLOGO

```
public static final int PICTURETYPE_ARTISTLOGO
```

Constant value: **19**

PICTURETYPE_PUBLISHERLOGO

```
public static final int PICTURETYPE_PUBLISHERLOGO
```

Constant value: **20**

MIMETYPES_JPEG

```
public static final java.lang.String MIMETYPES_JPEG
```

Constant value: **image/jpeg**

MIMETYPES_PNG

```
public static final java.lang.String MIMETYPES_PNG
```

Constant value: **image/png**

MIMETYPES_URL

```
public static final java.lang.String MIMETYPES_URL
```

Constant value: **-->**

(continued from last page)

PICTUREMAXFILESIZE

```
public static long PICTUREMAXFILESIZE
```

Constructors

ID3V2FrameAttachedPicture

```
public ID3V2FrameAttachedPicture(String idStr,  
                                int flags)
```

ID3V2FrameAttachedPicture

```
public ID3V2FrameAttachedPicture()
```

Methods

serializeBody

```
public int serializeBody(byte[] buffer,  
                        int offset)
```

deserializeBody

```
public void deserializeBody(byte[] buffer,  
                          int offset,  
                          int len)
```

getMapIdStr

```
public String getMapIdStr()
```

getBodySize

```
public int getBodySize()
```

getTextEncoding

```
public int getTextEncoding()
```

(continued from last page)

setTextEncoding

```
public void setTextEncoding(int textEncoding)
```

getMimeType

```
public String getMimeType()
```

setMimeType

```
public void setMimeType(String mimeType)
```

getPictureType

```
public int getPictureType()
```

setPictureType

```
public void setPictureType(int pictureType)
```

getDescription

```
public String getDescription()
```

setDescription

```
public void setDescription(String description)
```

getPictureData

```
public byte[] getPictureData()
```

setPictureData

```
public void setPictureData(byte[] pictureData)
```

setPictureDataAsURL

```
public void setPictureDataAsURL(String urlStr)
```

(continued from last page)

loadFile

```
public boolean loadFile(java.io.File file)
```

com.wowza.wms.media.mp3.model.idtags

Class ID3V2FrameBase

java.lang.Object

└─com.wowza.wms.media.mp3.model.idtags.ID3V2FrameBase

All Implemented Interfaces:

[IID3V2Frame](#)

Direct Known Subclasses:

[ID3V2FrameURLLinkUserDefined](#), [ID3V2FrameURLLink](#), [ID3V2FrameTextInformationUserDefined](#),
[ID3V2FrameTextInformation](#), [ID3V2FrameSynchronizedText](#), [ID3V2FrameRecommendedBufferSize](#),
[ID3V2FrameRawBytes](#), [ID3V2FramePrivate](#), [ID3V2FrameLinkedInformation](#), [ID3V2FrameComment](#),
[ID3V2FrameAttachedPicture](#)

public abstract class **ID3V2FrameBase**

extends Object

implements [IID3V2Frame](#)

Nested Class Summary

class	ID3V2FrameBase.DeserializeResult ID3V2FrameBase.DeserializeResult
-------	--

Field Summary

public static final	DEFAULT_LANGUAGE Value: eng
public static final	FLAGS_DEFAULT Value: 0
public static final	HEADERSIZE Value: 10
public static final	TAG_APIC Value: APIC
public static final	TAG_COMM Value: COMM
public static final	TAG_LINK Value: LINK
public static final	TAG_PRIV Value: PRIV

public static final	TAG_RBUF Value: RBUF
public static final	TAG_SYLT Value: SYLT
public static final	TAG_TALB Value: TALB
public static final	TAG_TBPM Value: TBPM
public static final	TAG_TCOM Value: TCOM
public static final	TAG_TCON Value: TCON
public static final	TAG_TCOP Value: TCOP
public static final	TAG_TDEN Value: TDEN
public static final	TAG_TDLY Value: TDLY
public static final	TAG_TDOR Value: TDOR
public static final	TAG_TDRC Value: TDRC
public static final	TAG_TDRL Value: TDRL
public static final	TAG_TDTG Value: TDTG
public static final	TAG_TENC Value: TENC
public static final	TAG_TEXT Value: TEXT
public static final	TAG_TFLT Value: TFLT

public static final	TAG_TIPL Value: TIPL
public static final	TAG_TIT1 Value: TIT1
public static final	TAG_TIT2 Value: TIT2
public static final	TAG_TIT3 Value: TIT3
public static final	TAG_TKEY Value: TKEY
public static final	TAG_TLAN Value: TLAN
public static final	TAG_TLEN Value: TLEN
public static final	TAG_TMCL Value: TMCL
public static final	TAG_TMED Value: TMED
public static final	TAG_TMOO Value: TMOO
public static final	TAG_TOAL Value: TOAL
public static final	TAG_TOFN Value: TOFN
public static final	TAG_TOLY Value: TOLY
public static final	TAG_TOPE Value: TOPE
public static final	TAG_TOWN Value: TOWN
public static final	TAG_TPE1 Value: TPE1

public static final	TAG_TPE2 Value: TPE2
public static final	TAG_TPE3 Value: TPE3
public static final	TAG_TPE4 Value: TPE4
public static final	TAG_TPOS Value: TPOS
public static final	TAG_TPRO Value: TPRO
public static final	TAG_TPUB Value: TPUB
public static final	TAG_TRCK Value: TRCK
public static final	TAG_TRSN Value: TRSN
public static final	TAG_TRSO Value: TRSO
public static final	TAG_TSOA Value: TSOA
public static final	TAG_TSOP Value: TSOP
public static final	TAG_TSOT Value: TSOT
public static final	TAG_TSRC Value: TSRC
public static final	TAG_TSSE Value: TSSE
public static final	TAG_TSST Value: TSST
public static final	TAG_TXXX Value: TXXX

public static final	TAG_UNKN Value: UNKN
public static final	TAG_WCOM Value: WCOM
public static final	TAG_WCOP Value: WCOP
public static final	TAG_WOAF Value: WOAF
public static final	TAG_WOAR Value: WOAR
public static final	TAG_WOAS Value: WOAS
public static final	TAG_WORS Value: WORS
public static final	TAG_WPAY Value: WPAY
public static final	TAG_WPUB Value: WPUB
public static final	TAG_WXXX Value: WXXX
public static final	TAGS_TEXTINFORMATION
public static final	TEXTENCODING_DEFAULT Value: UTF-8
public static final	TEXTENCODING_ISO_8859_1 Value: 0
public static final	TEXTENCODING_URL Value: ISO-8859-1
public static final	TEXTENCODING_UTF16 Value: 1
public static final	TEXTENCODING_UTF16BE Value: 2

public static final	TEXTENCODING_UTF8 Value: 3
---------------------	--

Constructor Summary

public	ID3V2FrameBase (String idStr, int flags)
--------	--

Method Summary

static int	byteStringLen (String value, int textEncoding)
static int	byteStringLen (String value, String encodingStr)
static ID3DeserializeFrameResult	deserializeFrame (byte[] buffer, int offset, int len)
static ID3V2FrameBase.DeserializeResult	deserializeString (byte[] buffer, int offset, int len, int textEncoding)
static ID3V2FrameBase.DeserializeResult	deserializeString (byte[] buffer, int offset, int len, String charsetName)
int	getFlags ()
String	getIdStr ()
String	getMapIdStr ()
int	getSize ()
int	serialize (byte[] buffer, int offset)
static int	serializeString (String value, byte[] buffer, int offset, int textEncoding, boolean addZero)
static int	serializeString (String value, byte[] buffer, int offset, String charsetName, boolean addZero)
static int	serializeStringLen (String value, int textEncoding, boolean addZero)
static int	serializeStringLen (String value, String encodingStr, boolean addZero)
void	setFlags (int flags)
void	setIdStr (String idStr)
static String	textEncodingToCharsetName (int textEncoding)
static String	trimTrailingZero (String value)

Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Methods inherited from interface [`com.wowza.wms.media.mp3.model.idtags.ID3V2Frame`](#)

[`deserializeBody`](#), [`getBodySize`](#), [`getFlags`](#), [`getIdStr`](#), [`getMapIdStr`](#), [`getSize`](#), [`serialize`](#), [`serializeBody`](#), [`setFlags`](#), [`setIdStr`](#)

Fields

HEADERSIZE

```
public static final int HEADERSIZE
```

Constant value: **10**

TEXTENCODING_ISO_8859_1

```
public static final int TEXTENCODING_ISO_8859_1
```

Constant value: **0**

TEXTENCODING_UTF16

```
public static final int TEXTENCODING_UTF16
```

Constant value: **1**

TEXTENCODING_UTF16BE

```
public static final int TEXTENCODING_UTF16BE
```

Constant value: **2**

TEXTENCODING_UTF8

```
public static final int TEXTENCODING_UTF8
```

Constant value: **3**

TEXTENCODING_URL

```
public static final java.lang.String TEXTENCODING_URL
```

Constant value: **ISO-8859-1**

(continued from last page)

TEXTENCODING_DEFAULT

```
public static final java.lang.String TEXTENCODING_DEFAULT
```

Constant value: **UTF-8**

DEFAULT_LANGUAGE

```
public static final java.lang.String DEFAULT_LANGUAGE
```

Constant value: **eng**

TAG_UNKN

```
public static final java.lang.String TAG_UNKN
```

Constant value: **UNKN**

TAG_PRIV

```
public static final java.lang.String TAG_PRIV
```

Constant value: **PRIV**

TAG_APIC

```
public static final java.lang.String TAG_APIC
```

Constant value: **APIC**

TAG_LINK

```
public static final java.lang.String TAG_LINK
```

Constant value: **LINK**

TAG_SYLT

```
public static final java.lang.String TAG_SYLT
```

Constant value: **SYLT**

TAG_RBUF

```
public static final java.lang.String TAG_RBUF
```

Constant value: **RBUF**

TAG_TALB

```
public static final java.lang.String TAG_TALB
```

(continued from last page)

Constant value: **TALB**

TAG_TBPM

```
public static final java.lang.String TAG_TBPM
```

Constant value: **TBPM**

TAG_TCOM

```
public static final java.lang.String TAG_TCOM
```

Constant value: **TCOM**

TAG_TCON

```
public static final java.lang.String TAG_TCON
```

Constant value: **TCON**

TAG_TCOP

```
public static final java.lang.String TAG_TCOP
```

Constant value: **TCOP**

TAG_TDEN

```
public static final java.lang.String TAG_TDEN
```

Constant value: **TDEN**

TAG_TDLY

```
public static final java.lang.String TAG_TDLY
```

Constant value: **TDLY**

TAG_TDOR

```
public static final java.lang.String TAG_TDOR
```

Constant value: **TDOR**

TAG_TDRC

```
public static final java.lang.String TAG_TDRC
```

Constant value: **TDRC**

TAG_TDRL

```
public static final java.lang.String TAG_TDRL
```

Constant value: **TDRL**

TAG_TDTG

```
public static final java.lang.String TAG_TDTG
```

Constant value: **TDTG**

TAG_TENC

```
public static final java.lang.String TAG_TENC
```

Constant value: **TENC**

TAG_TEXT

```
public static final java.lang.String TAG_TEXT
```

Constant value: **TEXT**

TAG_TFLT

```
public static final java.lang.String TAG_TFLT
```

Constant value: **TFLT**

TAG_TIPL

```
public static final java.lang.String TAG_TIPL
```

Constant value: **TIPL**

TAG_TIT1

```
public static final java.lang.String TAG_TIT1
```

Constant value: **TIT1**

TAG_TIT2

```
public static final java.lang.String TAG_TIT2
```

Constant value: **TIT2**

(continued from last page)

TAG_TIT3

```
public static final java.lang.String TAG_TIT3
```

Constant value: **TIT3**

TAG_TKEY

```
public static final java.lang.String TAG_TKEY
```

Constant value: **TKEY**

TAG_TLAN

```
public static final java.lang.String TAG_TLAN
```

Constant value: **TLAN**

TAG_TLEN

```
public static final java.lang.String TAG_TLEN
```

Constant value: **TLEN**

TAG_TMCL

```
public static final java.lang.String TAG_TMCL
```

Constant value: **TMCL**

TAG_TMED

```
public static final java.lang.String TAG_TMED
```

Constant value: **TMED**

TAG_TMOO

```
public static final java.lang.String TAG_TMOO
```

Constant value: **TMOO**

TAG_TOAL

```
public static final java.lang.String TAG_TOAL
```

Constant value: **TOAL**

TAG_TOFN

```
public static final java.lang.String TAG_TOFN
```

(continued from last page)

Constant value: **TOFN**

TAG_TOLY

```
public static final java.lang.String TAG_TOLY
```

Constant value: **TOLY**

TAG_TOPE

```
public static final java.lang.String TAG_TOPE
```

Constant value: **TOPE**

TAG_TOWN

```
public static final java.lang.String TAG_TOWN
```

Constant value: **TOWN**

TAG_TPE1

```
public static final java.lang.String TAG_TPE1
```

Constant value: **TPE1**

TAG_TPE2

```
public static final java.lang.String TAG_TPE2
```

Constant value: **TPE2**

TAG_TPE3

```
public static final java.lang.String TAG_TPE3
```

Constant value: **TPE3**

TAG_TPE4

```
public static final java.lang.String TAG_TPE4
```

Constant value: **TPE4**

TAG_TPOS

```
public static final java.lang.String TAG_TPOS
```

Constant value: **TPOS**

TAG_TPRO

```
public static final java.lang.String TAG_TPRO
```

Constant value: **TPRO**

TAG_TPUB

```
public static final java.lang.String TAG_TPUB
```

Constant value: **TPUB**

TAG_TRCK

```
public static final java.lang.String TAG_TRCK
```

Constant value: **TRCK**

TAG_TRSN

```
public static final java.lang.String TAG_TRSN
```

Constant value: **TRSN**

TAG_TRSO

```
public static final java.lang.String TAG_TRSO
```

Constant value: **TRSO**

TAG_TSOA

```
public static final java.lang.String TAG_TSOA
```

Constant value: **TSOA**

TAG_TSOP

```
public static final java.lang.String TAG_TSOP
```

Constant value: **TSOP**

TAG_TSOT

```
public static final java.lang.String TAG_TSOT
```

Constant value: **TSOT**

(continued from last page)

TAG_TSRC

```
public static final java.lang.String TAG_TSRC
```

Constant value: **TSRC**

TAG_TSSE

```
public static final java.lang.String TAG_TSSE
```

Constant value: **TSSE**

TAG_TSST

```
public static final java.lang.String TAG_TSST
```

Constant value: **TSST**

TAGS_TEXTINFORMATION

```
public static final java.lang.String TAGS_TEXTINFORMATION
```

TAG_WCOM

```
public static final java.lang.String TAG_WCOM
```

Constant value: **WCOM**

TAG_WCOP

```
public static final java.lang.String TAG_WCOP
```

Constant value: **WCOP**

TAG_WOAF

```
public static final java.lang.String TAG_WOAF
```

Constant value: **WOAF**

TAG_WOAR

```
public static final java.lang.String TAG_WOAR
```

Constant value: **WOAR**

TAG_WOAS

```
public static final java.lang.String TAG_WOAS
```

(continued from last page)

Constant value: **WOAS**

TAG_WORS

```
public static final java.lang.String TAG_WORS
```

Constant value: **WORS**

TAG_WPAY

```
public static final java.lang.String TAG_WPAY
```

Constant value: **WPAY**

TAG_WPUB

```
public static final java.lang.String TAG_WPUB
```

Constant value: **WPUB**

TAG_TXXX

```
public static final java.lang.String TAG_TXXX
```

Constant value: **TXXX**

TAG_COMM

```
public static final java.lang.String TAG_COMM
```

Constant value: **COMM**

TAG_WXXX

```
public static final java.lang.String TAG_WXXX
```

Constant value: **WXXX**

FLAGS_DEFAULT

```
public static final int FLAGS_DEFAULT
```

Constant value: **0**

Constructors

ID3V2FrameBase

```
public ID3V2FrameBase(String idStr,  
                      int flags)
```

Methods

textEncodingToCharsetName

```
public static String textEncodingToCharsetName(int textEncoding)
```

byteStringLen

```
public static int byteStringLen(String value,  
    int textEncoding)
```

byteStringLen

```
public static int byteStringLen(String value,  
    String encodingStr)
```

serializeStringLen

```
public static int serializeStringLen(String value,  
    int textEncoding,  
    boolean addZero)
```

serializeStringLen

```
public static int serializeStringLen(String value,  
    String encodingStr,  
    boolean addZero)
```

trimTrailingZero

```
public static String trimTrailingZero(String value)
```

serializeString

```
public static int serializeString(String value,  
    byte[] buffer,  
    int offset,  
    int textEncoding,  
    boolean addZero)
```

(continued from last page)

serializeString

```
public static int serializeString(String value,  
    byte[] buffer,  
    int offset,  
    String charsetName,  
    boolean addZero)
```

deserializeString

```
public static ID3V2FrameBase.DeserializeResult deserializeString(byte[] buffer,  
    int offset,  
    int len,  
    int textEncoding)
```

deserializeString

```
public static ID3V2FrameBase.DeserializeResult deserializeString(byte[] buffer,  
    int offset,  
    int len,  
    String charsetName)
```

serialize

```
public int serialize(byte[] buffer,  
    int offset)
```

getIdStr

```
public String getIdStr()
```

setIdStr

```
public void setIdStr(String idStr)
```

getFlags

```
public int getFlags()
```

setFlags

```
public void setFlags(int flags)
```

(continued from last page)

getSize

```
public int getSize()
```

getMapIdStr

```
public String getMapIdStr()
```

deserializeFrame

```
public static ID3DeserializeFrameResult deserializeFrame(byte[] buffer,  
    int offset,  
    int len)
```

com.wowza.wms.media.mp3.model.idtags Class ID3V2FrameBase.DeserializeResult

java.lang.Object

└─com.wowza.wms.media.mp3.model.idtags.ID3V2FrameBase.DeserializeResult

public static class **ID3V2FrameBase.DeserializeResult**
extends Object

Constructor Summary

public	ID3V2FrameBase.DeserializeResult()
--------	--

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

ID3V2FrameBase.DeserializeResult

public **ID3V2FrameBase.DeserializeResult()**

com.wowza.wms.media.mp3.model.idtags

Class ID3V2FrameComment

java.lang.Object

```

  |
+- com.wowza.wms.media.mp3.model.idtags.ID3V2FrameBase
  |
+- com.wowza.wms.media.mp3.model.idtags.ID3V2FrameComment

```

All Implemented Interfaces:

[IID3V2Frame](#)

public class **ID3V2FrameComment**
 extends [ID3V2FrameBase](#)

Field Summary

protected	language
protected	shortDescription
protected	text
protected	textEncoding

Fields inherited from class [com.wowza.wms.media.mp3.model.idtags.ID3V2FrameBase](#)

[DEFAULT_LANGUAGE](#), [FLAGS_DEFAULT](#), [HEADERSIZE](#), [TAG_APIC](#), [TAG_COMM](#), [TAG_LINK](#), [TAG_PRIV](#), [TAG_RBUF](#), [TAG_SYLT](#), [TAG_TALB](#), [TAG_TBPM](#), [TAG_TCOM](#), [TAG_TCON](#), [TAG_TCOPI](#), [TAG_TDEN](#), [TAG_TDLY](#), [TAG_TDOR](#), [TAG_TDRC](#), [TAG_TDRL](#), [TAG_TDTG](#), [TAG_TENC](#), [TAG_TEXT](#), [TAG_TFLT](#), [TAG_TIPL](#), [TAG_TIT1](#), [TAG_TIT2](#), [TAG_TIT3](#), [TAG_TKEY](#), [TAG_TLAN](#), [TAG_TLEN](#), [TAG_TMCL](#), [TAG_TMED](#), [TAG_TMOO](#), [TAG_TOAL](#), [TAG_TOFN](#), [TAG_TOLY](#), [TAG_TOPE](#), [TAG_TOWN](#), [TAG_TPE1](#), [TAG_TPE2](#), [TAG_TPE3](#), [TAG_TPE4](#), [TAG_TPOS](#), [TAG_TPRO](#), [TAG_TPUB](#), [TAG_TRCK](#), [TAG_TRSN](#), [TAG_TRSO](#), [TAG_TSOA](#), [TAG_TSOP](#), [TAG_TSOT](#), [TAG_TSRC](#), [TAG_TSSE](#), [TAG_TSST](#), [TAG_TXXX](#), [TAG_UNKN](#), [TAG_WCOM](#), [TAG_WCOP](#), [TAG_WOAF](#), [TAG_WOAR](#), [TAG_WOAS](#), [TAG_WORS](#), [TAG_WPAY](#), [TAG_WPUB](#), [TAG_WXXX](#), [TAGS_TEXTINFORMATION](#), [TEXTENCODING_DEFAULT](#), [TEXTENCODING_ISO_8859_1](#), [TEXTENCODING_URL](#), [TEXTENCODING_UTF16](#), [TEXTENCODING_UTF16BE](#), [TEXTENCODING_UTF8](#)

Constructor Summary

public	ID3V2FrameComment (String idStr, int flags)
public	ID3V2FrameComment (String idStr)
public	ID3V2FrameComment ()

Method Summary

void	deserializeBody (byte[] buffer, int offset, int len)
------	--

int	getBodySize()
String	getLanguage()
String	getShortDescription()
String	getText()
int	getTextEncoding()
int	serializeBody (byte[] buffer, int offset)
void	setLanguage (String language)
void	setShortDescription (String shortDescription)
void	setText (String text)
void	setTextEncoding (int textEncoding)

Methods inherited from class [com.wowza.wms.media.mp3.model.idtags.ID3V2FrameBase](#)

[byteStringLen](#), [byteStringLen](#), [deserializeFrame](#), [deserializeString](#), [deserializeString](#), [getFlags](#), [getIdStr](#), [getMapIdStr](#), [getSize](#), [serialize](#), [serializeString](#), [serializeString](#), [serializeStringLen](#), [serializeStringLen](#), [setFlags](#), [setIdStr](#), [textEncodingToCharsetName](#), [trimTrailingZero](#)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface [com.wowza.wms.media.mp3.model.idtags.IID3V2Frame](#)

[deserializeBody](#), [getBodySize](#), [getFlags](#), [getIdStr](#), [getMapIdStr](#), [getSize](#), [serialize](#), [serializeBody](#), [setFlags](#), [setIdStr](#)

Fields

textEncoding

protected int **textEncoding**

language

protected java.lang.String **language**

shortDescription

protected java.lang.String **shortDescription**

text

protected java.lang.String **text**

Constructors

ID3V2FrameComment

```
public ID3V2FrameComment(String idStr,  
                           int flags)
```

ID3V2FrameComment

```
public ID3V2FrameComment(String idStr)
```

ID3V2FrameComment

```
public ID3V2FrameComment()
```

Methods

serializeBody

```
public int serializeBody(byte[] buffer,  
                           int offset)
```

deserializeBody

```
public void deserializeBody(byte[] buffer,  
                             int offset,  
                             int len)
```

getBodySize

```
public int getBodySize()
```

(continued from last page)

getTextEncoding

```
public int getTextEncoding()
```

setTextEncoding

```
public void setTextEncoding(int textEncoding)
```

getLanguage

```
public String getLanguage()
```

setLanguage

```
public void setLanguage(String language)
```

getShortDescription

```
public String getShortDescription()
```

setShortDescription

```
public void setShortDescription(String shortDescription)
```

getText

```
public String getText()
```

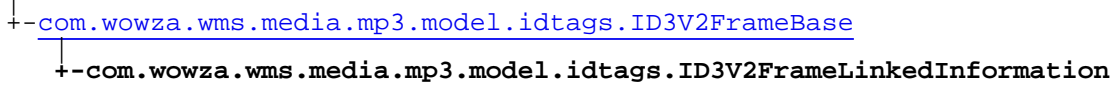
setText

```
public void setText(String text)
```

com.wowza.wms.media.mp3.model.idtags

Class ID3V2FrameLinkedInformation

java.lang.Object



All Implemented Interfaces:

[IID3V2Frame](#)

public class **ID3V2FrameLinkedInformation**

extends [ID3V2FrameBase](#)

Fields inherited from class [com.wowza.wms.media.mp3.model.idtags.ID3V2FrameBase](#)

[DEFAULT_LANGUAGE](#), [FLAGS_DEFAULT](#), [HEADERSIZE](#), [TAG_APIC](#), [TAG_COMM](#), [TAG_LINK](#), [TAG_PRIV](#), [TAG_RBUF](#), [TAG_SYLT](#), [TAG_TALB](#), [TAG_TBPM](#), [TAG_TCOM](#), [TAG_TCON](#), [TAG_TCOP](#), [TAG_TDEN](#), [TAG_TDLY](#), [TAG_TDOR](#), [TAG_TDRC](#), [TAG_TDRL](#), [TAG_TDTG](#), [TAG_TENC](#), [TAG_TEXT](#), [TAG_TFLT](#), [TAG_TIPL](#), [TAG_TIT1](#), [TAG_TIT2](#), [TAG_TIT3](#), [TAG_TKEY](#), [TAG_TLAN](#), [TAG_TLEN](#), [TAG_TMCL](#), [TAG_TMED](#), [TAG_TMOO](#), [TAG_TOAL](#), [TAG_TOFN](#), [TAG_TOLY](#), [TAG_TOPE](#), [TAG_TOWN](#), [TAG_TPE1](#), [TAG_TPE2](#), [TAG_TPE3](#), [TAG_TPE4](#), [TAG_TPOS](#), [TAG_TPRO](#), [TAG_TPUB](#), [TAG_TRCK](#), [TAG_TRSN](#), [TAG_TRSO](#), [TAG_TSOA](#), [TAG_TSOP](#), [TAG_TSOT](#), [TAG_TSRC](#), [TAG_TSSE](#), [TAG_TSST](#), [TAG_TXXX](#), [TAG_UNKN](#), [TAG_WCOM](#), [TAG_WCOP](#), [TAG_WOAF](#), [TAG_WOAR](#), [TAG_WOAS](#), [TAG_WORS](#), [TAG_WPAY](#), [TAG_WPUB](#), [TAG_WXXX](#), [TAGS_TEXTINFORMATION](#), [TEXTENCODING_DEFAULT](#), [TEXTENCODING_ISO_8859_1](#), [TEXTENCODING_URL](#), [TEXTENCODING_UTF16](#), [TEXTENCODING_UTF16BE](#), [TEXTENCODING_UTF8](#)

Constructor Summary

public	ID3V2FrameLinkedInformation (String idStr, int flags)
public	ID3V2FrameLinkedInformation ()

Method Summary

void	deserializeBody (byte[] buffer, int offset, int len)
int	getBodySize ()
byte[]	getData ()
String	getDescription ()
long	getFrameIdentifier ()
String	getURL ()
int	serializeBody (byte[] buffer, int offset)

void	<u>setData</u> (byte[] data)
void	<u>setDescription</u> (String description)
void	<u>setFrameIdentifier</u> (long frameIdentifier)
void	<u>setURL</u> (String url)

Methods inherited from class [com.wowza.wms.media.mp3.model.idtags.ID3V2FrameBase](#)

[byteStringLen](#), [byteStringLen](#), [deserializeFrame](#), [deserializeString](#), [deserializeString](#), [getFlags](#), [getIdStr](#), [getMapIdStr](#), [getSize](#), [serialize](#), [serializeString](#), [serializeString](#), [serializeStringLen](#), [serializeStringLen](#), [setFlags](#), [setIdStr](#), [textEncodingToCharsetName](#), [trimTrailingZero](#)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface [com.wowza.wms.media.mp3.model.idtags.IID3V2Frame](#)

[deserializeBody](#), [getBodySize](#), [getFlags](#), [getIdStr](#), [getMapIdStr](#), [getSize](#), [serialize](#), [serializeBody](#), [setFlags](#), [setIdStr](#)

Constructors

ID3V2FrameLinkedInformation

```
public ID3V2FrameLinkedInformation(String idStr,
                                   int flags)
```

ID3V2FrameLinkedInformation

```
public ID3V2FrameLinkedInformation()
```

Methods

serializeBody

```
public int serializeBody(byte[] buffer,
                          int offset)
```

deserializeBody

```
public void deserializeBody(byte[] buffer,
                              int offset,
                              int len)
```

(continued from last page)

getBodySize

```
public int getBodySize()
```

getDescription

```
public String getDescription()
```

setDescription

```
public void setDescription(String description)
```

getFrameIdentifier

```
public long getFrameIdentifier()
```

setFrameIdentifier

```
public void setFrameIdentifier(long frameIdentifier)
```

getURL

```
public String getURL()
```

setURL

```
public void setURL(String url)
```

getData

```
public byte[] getData()
```

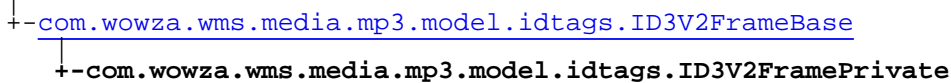
setData

```
public void setData(byte[] data)
```

com.wowza.wms.media.mp3.model.idtags

Class ID3V2FramePrivate

java.lang.Object



All Implemented Interfaces:

[IID3V2Frame](#)

public class **ID3V2FramePrivate**

extends [ID3V2FrameBase](#)

Fields inherited from class [com.wowza.wms.media.mp3.model.idtags.ID3V2FrameBase](#)

[DEFAULT_LANGUAGE](#), [FLAGS_DEFAULT](#), [HEADERSIZE](#), [TAG_APIC](#), [TAG_COMM](#), [TAG_LINK](#), [TAG_PRIV](#), [TAG_RBUF](#), [TAG_SYLT](#), [TAG_TALB](#), [TAG_TBPM](#), [TAG_TCOM](#), [TAG_TCON](#), [TAG_TCOP](#), [TAG_TDEN](#), [TAG_TDLY](#), [TAG_TDOR](#), [TAG_TDRC](#), [TAG_TDRL](#), [TAG_TDTG](#), [TAG_TENC](#), [TAG_TEXT](#), [TAG_TFLT](#), [TAG_TIPL](#), [TAG_TIT1](#), [TAG_TIT2](#), [TAG_TIT3](#), [TAG_TKEY](#), [TAG_TLAN](#), [TAG_TLEN](#), [TAG_TMCL](#), [TAG_TMED](#), [TAG_TMOO](#), [TAG_TOAL](#), [TAG_TOFN](#), [TAG_TOLY](#), [TAG_TOPE](#), [TAG_TOWN](#), [TAG_TPE1](#), [TAG_TPE2](#), [TAG_TPE3](#), [TAG_TPE4](#), [TAG_TPOS](#), [TAG_TPRO](#), [TAG_TPUB](#), [TAG_TRCK](#), [TAG_TRSN](#), [TAG_TRSO](#), [TAG_TSOA](#), [TAG_TSOP](#), [TAG_TSOT](#), [TAG_TSRC](#), [TAG_TSSE](#), [TAG_TSST](#), [TAG_TXXX](#), [TAG_UNKN](#), [TAG_WCOM](#), [TAG_WCOP](#), [TAG_WOAF](#), [TAG_WOAR](#), [TAG_WOAS](#), [TAG_WORS](#), [TAG_WPAY](#), [TAG_WPUB](#), [TAG_WXXX](#), [TAGS_TEXTINFORMATION](#), [TEXTENCODING_DEFAULT](#), [TEXTENCODING_ISO_8859_1](#), [TEXTENCODING_URL](#), [TEXTENCODING_UTF16](#), [TEXTENCODING_UTF16BE](#), [TEXTENCODING_UTF8](#)

Constructor Summary

public	ID3V2FramePrivate (String idStr, int flags)
public	ID3V2FramePrivate ()

Method Summary

void	deserializeBody (byte[] buffer, int offset, int len)
int	getBodySize ()
byte[]	getData ()
String	getOwnerIdentifier ()
int	serializeBody (byte[] buffer, int offset)
void	setData (byte[] data)
void	setOwnerIdentifier (String ownerIdentifier)

Methods inherited from class [com.wowza.wms.media.mp3.model.idtags.ID3V2FrameBase](#)

[byteStringLen](#), [byteStringLen](#), [deserializeFrame](#), [deserializeString](#), [deserializeString](#), [getFlags](#), [getIdStr](#), [getMapIdStr](#), [getSize](#), [serialize](#), [serializeString](#), [serializeString](#), [serializeStringLen](#), [serializeStringLen](#), [setFlags](#), [setIdStr](#), [textEncodingToCharsetName](#), [trimTrailingZero](#)

Methods inherited from class [java.lang.Object](#)

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

Methods inherited from interface [com.wowza.wms.media.mp3.model.idtags.IID3V2Frame](#)

[deserializeBody](#), [getBodySize](#), [getFlags](#), [getIdStr](#), [getMapIdStr](#), [getSize](#), [serialize](#), [serializeBody](#), [setFlags](#), [setIdStr](#)

Constructors

ID3V2FramePrivate

```
public ID3V2FramePrivate(String idStr,  
                          int flags)
```

ID3V2FramePrivate

```
public ID3V2FramePrivate()
```

Methods

serializeBody

```
public int serializeBody(byte[] buffer,  
                        int offset)
```

deserializeBody

```
public void deserializeBody(byte[] buffer,  
                          int offset,  
                          int len)
```

getBodySize

```
public int getBodySize()
```

(continued from last page)

getOwnerIdentifier

```
public String getOwnerIdentifier()
```

setOwnerIdentifier

```
public void setOwnerIdentifier(String ownerIdentifier)
```

getData

```
public byte[] getData()
```

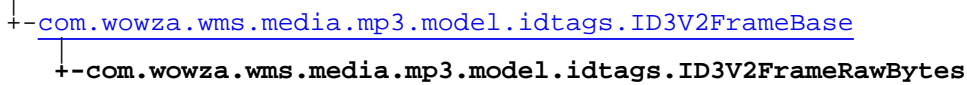
setData

```
public void setData(byte[] data)
```

com.wowza.wms.media.mp3.model.idtags

Class ID3V2FrameRawBytes

java.lang.Object



All Implemented Interfaces:

[IID3V2Frame](#)

public class **ID3V2FrameRawBytes**

extends [ID3V2FrameBase](#)

Fields inherited from class [com.wowza.wms.media.mp3.model.idtags.ID3V2FrameBase](#)

[DEFAULT_LANGUAGE](#), [FLAGS_DEFAULT](#), [HEADERSIZE](#), [TAG_APIC](#), [TAG_COMM](#), [TAG_LINK](#), [TAG_PRIV](#), [TAG_RBUF](#), [TAG_SYLT](#), [TAG_TALB](#), [TAG_TBPM](#), [TAG_TCOM](#), [TAG_TCON](#), [TAG_TCOPI](#), [TAG_TDEN](#), [TAG_TDLY](#), [TAG_TDOR](#), [TAG_TDRC](#), [TAG_TDRL](#), [TAG_TDTG](#), [TAG_TENC](#), [TAG_TEXT](#), [TAG_TFLT](#), [TAG_TIPL](#), [TAG_TIT1](#), [TAG_TIT2](#), [TAG_TIT3](#), [TAG_TKEY](#), [TAG_TLAN](#), [TAG_TLEN](#), [TAG_TMCL](#), [TAG_TMED](#), [TAG_TMOO](#), [TAG_TOAL](#), [TAG_TOFN](#), [TAG_TOLY](#), [TAG_TOPE](#), [TAG_TOWN](#), [TAG_TPE1](#), [TAG_TPE2](#), [TAG_TPE3](#), [TAG_TPE4](#), [TAG_TPOS](#), [TAG_TPRO](#), [TAG_TPUB](#), [TAG_TRCK](#), [TAG_TRSN](#), [TAG_TRSO](#), [TAG_TSOA](#), [TAG_TSOP](#), [TAG_TSOT](#), [TAG_TSRC](#), [TAG_TSSE](#), [TAG_TSST](#), [TAG_TXXX](#), [TAG_UNKN](#), [TAG_WCOM](#), [TAG_WCOP](#), [TAG_WOAF](#), [TAG_WOAR](#), [TAG_WOAS](#), [TAG_WORS](#), [TAG_WPAY](#), [TAG_WPUB](#), [TAG_WXXX](#), [TAGS_TEXTINFORMATION](#), [TEXTENCODING_DEFAULT](#), [TEXTENCODING_ISO_8859_1](#), [TEXTENCODING_URL](#), [TEXTENCODING_UTF16](#), [TEXTENCODING_UTF16BE](#), [TEXTENCODING_UTF8](#)

Constructor Summary

public	ID3V2FrameRawBytes (String idStr, int flags)
public	ID3V2FrameRawBytes (String idStr)

Method Summary

void	deserializeBody (byte[] buffer, int offset, int len)
int	getBodySize ()
int	serializeBody (byte[] buffer, int offset)

Methods inherited from class [com.wowza.wms.media.mp3.model.idtags.ID3V2FrameBase](#)

[byteStringLen](#), [byteStringLen](#), [deserializeFrame](#), [deserializeString](#), [deserializeString](#), [getFlags](#), [getIdStr](#), [getMapIdStr](#), [getSize](#), [serialize](#), [serializeString](#), [serializeString](#), [serializeStringLen](#), [serializeStringLen](#), [setFlags](#), [setIdStr](#), [textEncodingToCharsetName](#), [trimTrailingZero](#)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface [com.wowza.wms.media.mp3.model.idtags.IID3V2Frame](#)

[deserializeBody](#), [getBodySize](#), [getFlags](#), [getIdStr](#), [getMapIdStr](#), [getSize](#), [serialize](#), [serializeBody](#), [setFlags](#), [setIdStr](#)

Constructors

ID3V2FrameRawBytes

```
public ID3V2FrameRawBytes(String idStr,
                           int flags)
```

ID3V2FrameRawBytes

```
public ID3V2FrameRawBytes(String idStr)
```

Methods

serializeBody

```
public int serializeBody(byte[] buffer,
                          int offset)
```

deserializeBody

```
public void deserializeBody(byte[] buffer,
                             int offset,
                             int len)
```

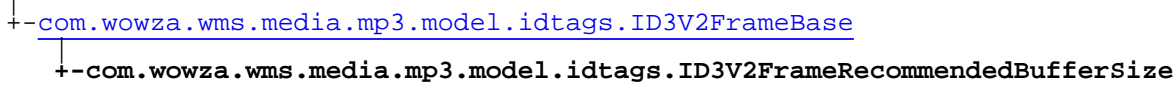
getBodySize

```
public int getBodySize()
```

com.wowza.wms.media.mp3.model.idtags

Class ID3V2FrameRecommendedBufferSize

java.lang.Object



All Implemented Interfaces:

[IID3V2Frame](#)

public class **ID3V2FrameRecommendedBufferSize**

extends [ID3V2FrameBase](#)

Fields inherited from class [com.wowza.wms.media.mp3.model.idtags.ID3V2FrameBase](#)

[DEFAULT_LANGUAGE](#), [FLAGS_DEFAULT](#), [HEADERSIZE](#), [TAG_APIC](#), [TAG_COMM](#), [TAG_LINK](#), [TAG_PRIV](#), [TAG_RBUF](#), [TAG_SYLT](#), [TAG_TALB](#), [TAG_TBPM](#), [TAG_TCOM](#), [TAG_TCON](#), [TAG_TCOP](#), [TAG_TDEN](#), [TAG_TDLY](#), [TAG_TDOR](#), [TAG_TDRC](#), [TAG_TDRL](#), [TAG_TDTG](#), [TAG_TENC](#), [TAG_TEXT](#), [TAG_TFLT](#), [TAG_TIPL](#), [TAG_TIT1](#), [TAG_TIT2](#), [TAG_TIT3](#), [TAG_TKEY](#), [TAG_TLAN](#), [TAG_TLEN](#), [TAG_TMCL](#), [TAG_TMED](#), [TAG_TMOO](#), [TAG_TOAL](#), [TAG_TOFN](#), [TAG_TOLY](#), [TAG_TOPE](#), [TAG_TOWN](#), [TAG_TPE1](#), [TAG_TPE2](#), [TAG_TPE3](#), [TAG_TPE4](#), [TAG_TPOS](#), [TAG_TPRO](#), [TAG_TPUB](#), [TAG_TRCK](#), [TAG_TRSN](#), [TAG_TRSO](#), [TAG_TSOA](#), [TAG_TSOP](#), [TAG_TSOT](#), [TAG_TSRC](#), [TAG_TSSE](#), [TAG_TSST](#), [TAG_TXXX](#), [TAG_UNKN](#), [TAG_WCOM](#), [TAG_WCOP](#), [TAG_WOAF](#), [TAG_WOAR](#), [TAG_WOAS](#), [TAG_WORS](#), [TAG_WPAY](#), [TAG_WPUB](#), [TAG_WXXX](#), [TAGS_TEXTINFORMATION](#), [TEXTENCODING_DEFAULT](#), [TEXTENCODING_ISO_8859_1](#), [TEXTENCODING_URL](#), [TEXTENCODING_UTF16](#), [TEXTENCODING_UTF16BE](#), [TEXTENCODING_UTF8](#)

Constructor Summary

public	ID3V2FrameRecommendedBufferSize (String idStr, int flags)
public	ID3V2FrameRecommendedBufferSize ()

Method Summary

void	deserializeBody (byte[] buffer, int offset, int len)
int	getBodySize ()
int	getBufferSize ()
byte	getEmbeddedFlag ()
long	getOffsetToNextTag ()
int	serializeBody (byte[] buffer, int offset)
void	setBufferSize (int bufferSize)

void	<u>setEmbeddedFlag</u> (byte embeddedFlag)
void	<u>setOffsetToNextTag</u> (long offsetToNextTag)

Methods inherited from class [com.wowza.wms.media.mp3.model.idtags.ID3V2FrameBase](#)

[byteStringLen](#), [byteStringLen](#), [deserializeFrame](#), [deserializeString](#), [deserializeString](#), [getFlags](#), [getIdStr](#), [getMapIdStr](#), [getSize](#), [serialize](#), [serializeString](#), [serializeString](#), [serializeStringLen](#), [serializeStringLen](#), [setFlags](#), [setIdStr](#), [textEncodingToCharsetName](#), [trimTrailingZero](#)

Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Methods inherited from interface [com.wowza.wms.media.mp3.model.idtags.IID3V2Frame](#)

[deserializeBody](#), [getBodySize](#), [getFlags](#), [getIdStr](#), [getMapIdStr](#), [getSize](#), [serialize](#), [serializeBody](#), [setFlags](#), [setIdStr](#)

Constructors

ID3V2FrameRecommendedBufferSize

```
public ID3V2FrameRecommendedBufferSize(String idStr,
                                       int flags)
```

ID3V2FrameRecommendedBufferSize

```
public ID3V2FrameRecommendedBufferSize()
```

Methods

serializeBody

```
public int serializeBody(byte[] buffer,
                        int offset)
```

deserializeBody

```
public void deserializeBody(byte[] buffer,
                          int offset,
                          int len)
```

(continued from last page)

getBodySize

```
public int getBodySize()
```

getBufferSize

```
public int getBufferSize()
```

setBufferSize

```
public void setBufferSize(int bufferSize)
```

getEmbeddedFlag

```
public byte getEmbeddedFlag()
```

setEmbeddedFlag

```
public void setEmbeddedFlag(byte embeddedFlag)
```

getOffsetToNextTag

```
public long getOffsetToNextTag()
```

setOffsetToNextTag

```
public void setOffsetToNextTag(long offsetToNextTag)
```

com.wowza.wms.media.mp3.model.idtags Class ID3V2FrameSynchronizedText

java.lang.Object

└- [com.wowza.wms.media.mp3.model.idtags.ID3V2FrameBase](#)
└- [com.wowza.wms.media.mp3.model.idtags.ID3V2FrameSynchronizedText](#)

All Implemented Interfaces:

[IID3V2Frame](#)

public class **ID3V2FrameSynchronizedText**
extends [ID3V2FrameBase](#)

Field Summary

public static final	CONTENTTYPE_CHORD Value: 5
public static final	CONTENTTYPE_EVENTS Value: 4
public static final	CONTENTTYPE_LYRICS Value: 1
public static final	CONTENTTYPE_MOVEMENT Value: 3
public static final	CONTENTTYPE_OTHER Value: 0
public static final	CONTENTTYPE_TRANSCRIPTION Value: 2
public static final	CONTENTTYPE_TRIVIA Value: 6
public static final	CONTENTTYPE_URLIMAGES Value: 8
public static final	CONTENTTYPE_URLWEBPAGES Value: 7
public static final	TIMESTAMPFORMAT_MILLISECONDS Value: 2

public static final	<u>TIMESTAMPFORMAT_MPEG</u> Value: 1
---------------------	--

Fields inherited from class [com.wowza.wms.media.mp3.model.idtags.ID3V2FrameBase](#)

[DEFAULT_LANGUAGE](#), [FLAGS_DEFAULT](#), [HEADERSIZE](#), [TAG_APIC](#), [TAG_COMM](#), [TAG_LINK](#), [TAG_PRIV](#), [TAG_RBUF](#), [TAG_SYLT](#), [TAG_TALB](#), [TAG_TBPM](#), [TAG_TCOM](#), [TAG_TCON](#), [TAG_TCOP](#), [TAG_TDEN](#), [TAG_TDLY](#), [TAG_TDOR](#), [TAG_TDRC](#), [TAG_TDRL](#), [TAG_TDTG](#), [TAG_TENC](#), [TAG_TEXT](#), [TAG_TFLT](#), [TAG_TIPL](#), [TAG_TIT1](#), [TAG_TIT2](#), [TAG_TIT3](#), [TAG_TKEY](#), [TAG_TLAN](#), [TAG_TLEN](#), [TAG_TMCL](#), [TAG_TMED](#), [TAG_TMOO](#), [TAG_TOAL](#), [TAG_TOFN](#), [TAG_TOLY](#), [TAG_TOPE](#), [TAG_TOWN](#), [TAG_TPE1](#), [TAG_TPE2](#), [TAG_TPE3](#), [TAG_TPE4](#), [TAG_TPOS](#), [TAG_TPRO](#), [TAG_TPUB](#), [TAG_TRCK](#), [TAG_TRSN](#), [TAG_TRSO](#), [TAG_TSOA](#), [TAG_TSOP](#), [TAG_TSOT](#), [TAG_TSRC](#), [TAG_TSSE](#), [TAG_TSST](#), [TAG_TXXX](#), [TAG_UNKN](#), [TAG_WCOM](#), [TAG_WCOP](#), [TAG_WOAF](#), [TAG_WOAR](#), [TAG_WOAS](#), [TAG_WORS](#), [TAG_WPAY](#), [TAG_WPUB](#), [TAG_WXXX](#), [TAGS_TEXTINFORMATION](#), [TEXTENCODING_DEFAULT](#), [TEXTENCODING_ISO_8859_1](#), [TEXTENCODING_URL](#), [TEXTENCODING_UTF16](#), [TEXTENCODING_UTF16BE](#), [TEXTENCODING_UTF8](#)

Constructor Summary

public	<u>ID3V2FrameSynchronizedText</u> (String idStr, int flags)
public	<u>ID3V2FrameSynchronizedText</u> ()

Method Summary

void	<u>addContentDescriptor</u> (<u>ID3V2FrameSynchronizedTextDescriptor</u> contentDescriptor)
void	<u>addContentDescriptor</u> (long timecode, String value)
void	<u>deserializeBody</u> (byte[] buffer, int offset, int len)
int	<u>getBodySize</u> ()
java.util.List	<u>getContentDescriptors</u> ()
byte	<u>getContentType</u> ()
String	<u>getLanguage</u> ()
int	<u>getTextEncoding</u> ()
byte	<u>getTimeStampFormat</u> ()
int	<u>serializeBody</u> (byte[] buffer, int offset)
void	<u>setContentType</u> (byte contentType)
void	<u>setLanguage</u> (String language)
void	<u>setTextEncoding</u> (int textEncoding)

void	<u>setTimeStampFormat</u> (byte timeStampFormat)
------	--

Methods inherited from class [com.wowza.wms.media.mp3.model.idtags.ID3V2FrameBase](#)

[byteStringLen](#), [byteStringLen](#), [deserializeFrame](#), [deserializeString](#), [deserializeString](#), [getFlags](#), [getIdStr](#), [getMapIdStr](#), [getSize](#), [serialize](#), [serializeString](#), [serializeString](#), [serializeStringLen](#), [serializeStringLen](#), [setFlags](#), [setIdStr](#), [textEncodingToCharsetName](#), [trimTrailingZero](#)

Methods inherited from class java.lang.Object

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

Methods inherited from interface [com.wowza.wms.media.mp3.model.idtags.IID3V2Frame](#)

[deserializeBody](#), [getBodySize](#), [getFlags](#), [getIdStr](#), [getMapIdStr](#), [getSize](#), [serialize](#), [serializeBody](#), [setFlags](#), [setIdStr](#)

Fields

CONTENTTYPE_OTHER

```
public static final int CONTENTTYPE_OTHER
```

Constant value: **0**

CONTENTTYPE_LYRICS

```
public static final int CONTENTTYPE_LYRICS
```

Constant value: **1**

CONTENTTYPE_TRANSCRIPTION

```
public static final int CONTENTTYPE_TRANSCRIPTION
```

Constant value: **2**

CONTENTTYPE_MOVEMENT

```
public static final int CONTENTTYPE_MOVEMENT
```

Constant value: **3**

CONTENTTYPE_EVENTS

```
public static final int CONTENTTYPE_EVENTS
```

Constant value: **4**

CONTENTTYPE_CHORD

```
public static final int CONTENTTYPE_CHORD
```

Constant value: **5**

CONTENTTYPE_TRIVIA

```
public static final int CONTENTTYPE_TRIVIA
```

Constant value: **6**

CONTENTTYPE_URLWEBPAGES

```
public static final int CONTENTTYPE_URLWEBPAGES
```

Constant value: **7**

CONTENTTYPE_URLIMAGES

```
public static final int CONTENTTYPE_URLIMAGES
```

Constant value: **8**

TIMESTAMPFORMAT_MPEG

```
public static final int TIMESTAMPFORMAT_MPEG
```

Constant value: **1**

TIMESTAMPFORMAT_MILLISECONDS

```
public static final int TIMESTAMPFORMAT_MILLISECONDS
```

Constant value: **2**

Constructors

ID3V2FrameSynchronizedText

```
public ID3V2FrameSynchronizedText(String idStr,  
                                   int flags)
```

ID3V2FrameSynchronizedText

```
public ID3V2FrameSynchronizedText()
```

Methods

(continued from last page)

serializeBody

```
public int serializeBody(byte[] buffer,  
    int offset)
```

deserializeBody

```
public void deserializeBody(byte[] buffer,  
    int offset,  
    int len)
```

getBodySize

```
public int getBodySize()
```

getTextEncoding

```
public int getTextEncoding()
```

setTextEncoding

```
public void setTextEncoding(int textEncoding)
```

getLanguage

```
public String getLanguage()
```

setLanguage

```
public void setLanguage(String language)
```

getTimeStampFormat

```
public byte getTimeStampFormat()
```

setTimeStampFormat

```
public void setTimeStampFormat(byte timeStampFormat)
```

(continued from last page)

getContentType

```
public byte getContentType()
```

setContentType

```
public void setContentType(byte contentType)
```

getContentDescriptors

```
public java.util.List getContentDescriptors()
```

addContentDescriptor

```
public void addContentDescriptor(ID3V2FrameSynchronizedTextDescriptor  
contentDescriptor)
```

addContentDescriptor

```
public void addContentDescriptor(long timecode,  
    String value)
```

com.wowza.wms.media.mp3.model.idtags Class ID3V2FrameSynchronizedTextDescriptor

java.lang.Object

└-com.wowza.wms.media.mp3.model.idtags.ID3V2FrameSynchronizedTextDescriptor

public class **ID3V2FrameSynchronizedTextDescriptor**
extends Object

Constructor Summary

public	ID3V2FrameSynchronizedTextDescriptor()
public	ID3V2FrameSynchronizedTextDescriptor(long timecode, String value)

Method Summary

long	getTimecode()
String	getValue()
void	setTimecode(long timecode)
void	setValue(String value)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

ID3V2FrameSynchronizedTextDescriptor

public **ID3V2FrameSynchronizedTextDescriptor()**

ID3V2FrameSynchronizedTextDescriptor

public **ID3V2FrameSynchronizedTextDescriptor**(long timecode,
String value)

Methods

(continued from last page)

getTimecode

```
public long getTimecode()
```

setTimecode

```
public void setTimecode(long timecode)
```

getValue

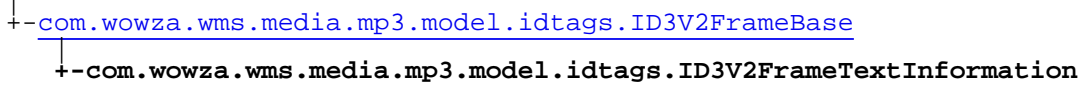
```
public String getValue()
```

setValue

```
public void setValue(String value)
```

com.wowza.wms.media.mp3.model.idtags Class ID3V2FrameTextInformation

java.lang.Object



All Implemented Interfaces:

[IID3V2Frame](#)

public class **ID3V2FrameTextInformation**
extends [ID3V2FrameBase](#)

Fields inherited from class [com.wowza.wms.media.mp3.model.idtags.ID3V2FrameBase](#)

[DEFAULT_LANGUAGE](#), [FLAGS_DEFAULT](#), [HEADERSIZE](#), [TAG_APIC](#), [TAG_COMM](#), [TAG_LINK](#), [TAG_PRIV](#), [TAG_RBUF](#), [TAG_SYLT](#), [TAG_TALB](#), [TAG_TBPM](#), [TAG_TCOM](#), [TAG_TCON](#), [TAG_TCOP](#), [TAG_TDEN](#), [TAG_TDLY](#), [TAG_TDOR](#), [TAG_TDRC](#), [TAG_TDRL](#), [TAG_TDTG](#), [TAG_TENC](#), [TAG_TEXT](#), [TAG_TFLT](#), [TAG_TIPL](#), [TAG_TIT1](#), [TAG_TIT2](#), [TAG_TIT3](#), [TAG_TKEY](#), [TAG_TLAN](#), [TAG_TLEN](#), [TAG_TMCL](#), [TAG_TMED](#), [TAG_TMOO](#), [TAG_TOAL](#), [TAG_TOFN](#), [TAG_TOLY](#), [TAG_TOPE](#), [TAG_TOWN](#), [TAG_TPE1](#), [TAG_TPE2](#), [TAG_TPE3](#), [TAG_TPE4](#), [TAG_TPOS](#), [TAG_TPRO](#), [TAG_TPUB](#), [TAG_TRCK](#), [TAG_TRSN](#), [TAG_TRSO](#), [TAG_TSOA](#), [TAG_TSOP](#), [TAG_TSOT](#), [TAG_TSRC](#), [TAG_TSSE](#), [TAG_TSST](#), [TAG_TXXX](#), [TAG_UNKN](#), [TAG_WCOM](#), [TAG_WCOP](#), [TAG_WOAF](#), [TAG_WOAR](#), [TAG_WOAS](#), [TAG_WORS](#), [TAG_WPAY](#), [TAG_WPUB](#), [TAG_WXXX](#), [TAGS_TEXTINFORMATION](#), [TEXTENCODING_DEFAULT](#), [TEXTENCODING_ISO_8859_1](#), [TEXTENCODING_URL](#), [TEXTENCODING_UTF16](#), [TEXTENCODING_UTF16BE](#), [TEXTENCODING_UTF8](#)

Constructor Summary

public	ID3V2FrameTextInformation (String idStr, int flags)
public	ID3V2FrameTextInformation (String idStr)

Method Summary

void	deserializeBody (byte[] buffer, int offset, int len)
int	getBodySize ()
int	getTextEncoding ()
String	getValue ()
int	serializeBody (byte[] buffer, int offset)
void	setTextEncoding (int textEncoding)
void	setValue (String value)

Methods inherited from class [com.wowza.wms.media.mp3.model.idtags.ID3V2FrameBase](#)

[byteStringLen](#), [byteStringLen](#), [deserializeFrame](#), [deserializeString](#), [deserializeString](#), [getFlags](#), [getIdStr](#), [getMapIdStr](#), [getSize](#), [serialize](#), [serializeString](#), [serializeString](#), [serializeStringLen](#), [serializeStringLen](#), [setFlags](#), [setIdStr](#), [textEncodingToCharsetName](#), [trimTrailingZero](#)

Methods inherited from class [java.lang.Object](#)

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

Methods inherited from interface [com.wowza.wms.media.mp3.model.idtags.IID3V2Frame](#)

[deserializeBody](#), [getBodySize](#), [getFlags](#), [getIdStr](#), [getMapIdStr](#), [getSize](#), [serialize](#), [serializeBody](#), [setFlags](#), [setIdStr](#)

Constructors

ID3V2FrameTextInformation

```
public ID3V2FrameTextInformation(String idStr,  
                                int flags)
```

ID3V2FrameTextInformation

```
public ID3V2FrameTextInformation(String idStr)
```

Methods

serializeBody

```
public int serializeBody(byte[] buffer,  
                        int offset)
```

deserializeBody

```
public void deserializeBody(byte[] buffer,  
                          int offset,  
                          int len)
```

getBodySize

```
public int getBodySize()
```

(continued from last page)

getValue

```
public String getValue()
```

setValue

```
public void setValue(String value)
```

getTextEncoding

```
public int getTextEncoding()
```

setTextEncoding

```
public void setTextEncoding(int textEncoding)
```

com.wowza.wms.media.mp3.model.idtags

Class ID3V2FrameTextInformationUserDefined

java.lang.Object



All Implemented Interfaces:

[IID3V2Frame](#)

public class **ID3V2FrameTextInformationUserDefined**
 extends [ID3V2FrameBase](#)

Fields inherited from class [com.wowza.wms.media.mp3.model.idtags.ID3V2FrameBase](#)

[DEFAULT_LANGUAGE](#), [FLAGS_DEFAULT](#), [HEADERSIZE](#), [TAG_APIC](#), [TAG_COMM](#), [TAG_LINK](#), [TAG_PRIV](#), [TAG_RBUF](#),
[TAG_SYLT](#), [TAG_TALB](#), [TAG_TBPM](#), [TAG_TCOM](#), [TAG_TCON](#), [TAG_TCOP](#), [TAG_TDEN](#), [TAG_TDLY](#), [TAG_TDOR](#),
[TAG_TDRC](#), [TAG_TDRL](#), [TAG_TDTG](#), [TAG_TENC](#), [TAG_TEXT](#), [TAG_TFLT](#), [TAG_TIPL](#), [TAG_TIT1](#), [TAG_TIT2](#),
[TAG_TIT3](#), [TAG_TKEY](#), [TAG_TLAN](#), [TAG_TLEN](#), [TAG_TMCL](#), [TAG_TMED](#), [TAG_TMOO](#), [TAG_TOAL](#), [TAG_TOFN](#),
[TAG_TOLY](#), [TAG_TOPE](#), [TAG_TOWN](#), [TAG_TPE1](#), [TAG_TPE2](#), [TAG_TPE3](#), [TAG_TPE4](#), [TAG_TPOS](#), [TAG_TPRO](#),
[TAG_TPUB](#), [TAG_TRCK](#), [TAG_TRSN](#), [TAG_TRSO](#), [TAG_TSOA](#), [TAG_TSOP](#), [TAG_TSOT](#), [TAG_TSRC](#), [TAG_TSSE](#),
[TAG_TSST](#), [TAG_TXXX](#), [TAG_UNKN](#), [TAG_WCOM](#), [TAG_WCOP](#), [TAG_WOAF](#), [TAG_WOAR](#), [TAG_WOAS](#), [TAG_WORS](#),
[TAG_WPAY](#), [TAG_WPUB](#), [TAG_WXXX](#), [TAGS_TEXTINFORMATION](#), [TEXTENCODING_DEFAULT](#),
[TEXTENCODING_ISO_8859_1](#), [TEXTENCODING_URL](#), [TEXTENCODING_UTF16](#), [TEXTENCODING_UTF16BE](#),
[TEXTENCODING_UTF8](#)

Constructor Summary

public	ID3V2FrameTextInformationUserDefined (String idStr, int flags)
public	ID3V2FrameTextInformationUserDefined (String idStr)
public	ID3V2FrameTextInformationUserDefined ()

Method Summary

void	deserializeBody (byte[] buffer, int offset, int len)
int	getBodySize ()
String	getDescription ()
int	getTextEncoding ()
String	getValue ()
int	serializeBody (byte[] buffer, int offset)

void	setDescription (String description)
void	setTextEncoding (int textEncoding)
void	setValue (String value)

Methods inherited from class [com.wowza.wms.media.mp3.model.idtags.ID3V2FrameBase](#)

[byteStringLen](#), [byteStringLen](#), [deserializeFrame](#), [deserializeString](#), [deserializeString](#), [getFlags](#), [getIdStr](#), [getMapIdStr](#), [getSize](#), [serialize](#), [serializeString](#), [serializeString](#), [serializeStringLen](#), [serializeStringLen](#), [setFlags](#), [setIdStr](#), [textEncodingToCharsetName](#), [trimTrailingZero](#)

Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Methods inherited from interface [com.wowza.wms.media.mp3.model.idtags.IID3V2Frame](#)

[deserializeBody](#), [getBodySize](#), [getFlags](#), [getIdStr](#), [getMapIdStr](#), [getSize](#), [serialize](#), [serializeBody](#), [setFlags](#), [setIdStr](#)

Constructors

ID3V2FrameTextInformationUserDefined

```
public ID3V2FrameTextInformationUserDefined(String idStr,
                                             int flags)
```

ID3V2FrameTextInformationUserDefined

```
public ID3V2FrameTextInformationUserDefined(String idStr)
```

ID3V2FrameTextInformationUserDefined

```
public ID3V2FrameTextInformationUserDefined()
```

Methods

serializeBody

```
public int serializeBody(byte[] buffer,
                          int offset)
```

(continued from last page)

deserializeBody

```
public void deserializeBody(byte[] buffer,  
    int offset,  
    int len)
```

getBodySize

```
public int getBodySize()
```

getValue

```
public String getValue()
```

setValue

```
public void setValue(String value)
```

getTextEncoding

```
public int getTextEncoding()
```

setTextEncoding

```
public void setTextEncoding(int textEncoding)
```

getDescription

```
public String getDescription()
```

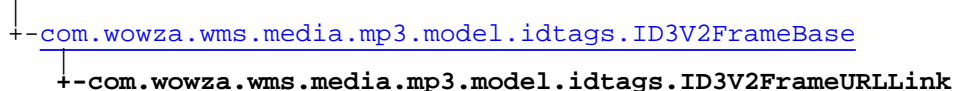
setDescription

```
public void setDescription(String description)
```

com.wowza.wms.media.mp3.model.idtags

Class ID3V2FrameURLLink

java.lang.Object



All Implemented Interfaces:

[IID3V2Frame](#)

public class **ID3V2FrameURLLink**

extends [ID3V2FrameBase](#)

Fields inherited from class [com.wowza.wms.media.mp3.model.idtags.ID3V2FrameBase](#)

[DEFAULT_LANGUAGE](#), [FLAGS_DEFAULT](#), [HEADERSIZE](#), [TAG_APIC](#), [TAG_COMM](#), [TAG_LINK](#), [TAG_PRIV](#), [TAG_RBUF](#), [TAG_SYLT](#), [TAG_TALB](#), [TAG_TBPM](#), [TAG_TCOM](#), [TAG_TCON](#), [TAG_TCOP](#), [TAG_TDEN](#), [TAG_TDLY](#), [TAG_TDOR](#), [TAG_TDRC](#), [TAG_TDRL](#), [TAG_TDTG](#), [TAG_TENC](#), [TAG_TEXT](#), [TAG_TFLT](#), [TAG_TIPL](#), [TAG_TIT1](#), [TAG_TIT2](#), [TAG_TIT3](#), [TAG_TKEY](#), [TAG_TLAN](#), [TAG_TLEN](#), [TAG_TMCL](#), [TAG_TMED](#), [TAG_TMOO](#), [TAG_TOAL](#), [TAG_TOFN](#), [TAG_TOLY](#), [TAG_TOPE](#), [TAG_TOWN](#), [TAG_TPE1](#), [TAG_TPE2](#), [TAG_TPE3](#), [TAG_TPE4](#), [TAG_TPOS](#), [TAG_TPRO](#), [TAG_TPUB](#), [TAG_TRCK](#), [TAG_TRSN](#), [TAG_TRSO](#), [TAG_TSOA](#), [TAG_TSOP](#), [TAG_TSOT](#), [TAG_TSRC](#), [TAG_TSSE](#), [TAG_TSST](#), [TAG_TXXX](#), [TAG_UNKN](#), [TAG_WCOM](#), [TAG_WCOP](#), [TAG_WOAF](#), [TAG_WOAR](#), [TAG_WOAS](#), [TAG_WORS](#), [TAG_WPAY](#), [TAG_WPUB](#), [TAG_WXXX](#), [TAGS_TEXTINFORMATION](#), [TEXTENCODING_DEFAULT](#), [TEXTENCODING_ISO_8859_1](#), [TEXTENCODING_URL](#), [TEXTENCODING_UTF16](#), [TEXTENCODING_UTF16BE](#), [TEXTENCODING_UTF8](#)

Constructor Summary

public	ID3V2FrameURLLink (String idStr, int flags)
public	ID3V2FrameURLLink (String idStr)

Method Summary

void	deserializeBody (byte[] buffer, int offset, int len)
int	getBodySize ()
String	getURL ()
int	serializeBody (byte[] buffer, int offset)
void	setURL (String value)

Methods inherited from class [com.wowza.wms.media.mp3.model.idtags.ID3V2FrameBase](#)

[byteStringLen](#), [byteStringLen](#), [deserializeFrame](#), [deserializeString](#), [deserializeString](#), [getFlags](#), [getIdStr](#), [getMapIdStr](#), [getSize](#), [serialize](#), [serializeString](#), [serializeString](#), [serializeStringLen](#), [serializeStringLen](#), [setFlags](#), [setIdStr](#), [textEncodingToCharsetName](#), [trimTrailingZero](#)

Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Methods inherited from interface [com.wowza.wms.media.mp3.model.idtags.IID3V2Frame](#)

[deserializeBody](#), [getBodySize](#), [getFlags](#), [getIdStr](#), [getMapIdStr](#), [getSize](#), [serialize](#), [serializeBody](#), [setFlags](#), [setIdStr](#)

Constructors

ID3V2FrameURLLink

```
public ID3V2FrameURLLink(String idStr,  
                           int flags)
```

ID3V2FrameURLLink

```
public ID3V2FrameURLLink(String idStr)
```

Methods

serializeBody

```
public int serializeBody(byte[] buffer,  
                          int offset)
```

deserializeBody

```
public void deserializeBody(byte[] buffer,  
                             int offset,  
                             int len)
```

getBodySize

```
public int getBodySize()
```

(continued from last page)

getURL

```
public String getURL()
```

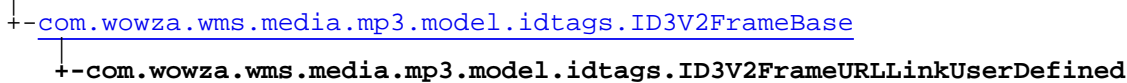
setURL

```
public void setURL(String value)
```


com.wowza.wms.media.mp3.model.idtags

Class ID3V2FrameURLLinkUserDefined

java.lang.Object



All Implemented Interfaces:

[IID3V2Frame](#)

public class **ID3V2FrameURLLinkUserDefined**

extends [ID3V2FrameBase](#)

Fields inherited from class [com.wowza.wms.media.mp3.model.idtags.ID3V2FrameBase](#)

[DEFAULT_LANGUAGE](#), [FLAGS_DEFAULT](#), [HEADERSIZE](#), [TAG_APIC](#), [TAG_COMM](#), [TAG_LINK](#), [TAG_PRIV](#), [TAG_RBUF](#), [TAG_SYLT](#), [TAG_TALB](#), [TAG_TBPM](#), [TAG_TCOM](#), [TAG_TCON](#), [TAG_TCOP](#), [TAG_TDEN](#), [TAG_TDLY](#), [TAG_TDOR](#), [TAG_TDRC](#), [TAG_TDRL](#), [TAG_TDTG](#), [TAG_TENC](#), [TAG_TEXT](#), [TAG_TFLT](#), [TAG_TIPL](#), [TAG_TIT1](#), [TAG_TIT2](#), [TAG_TIT3](#), [TAG_TKEY](#), [TAG_TLAN](#), [TAG_TLEN](#), [TAG_TMCL](#), [TAG_TMED](#), [TAG_TMOO](#), [TAG_TOAL](#), [TAG_TOFN](#), [TAG_TOLY](#), [TAG_TOPE](#), [TAG_TOWN](#), [TAG_TPE1](#), [TAG_TPE2](#), [TAG_TPE3](#), [TAG_TPE4](#), [TAG_TPOS](#), [TAG_TPRO](#), [TAG_TPUB](#), [TAG_TRCK](#), [TAG_TRSN](#), [TAG_TRSO](#), [TAG_TSOA](#), [TAG_TSOP](#), [TAG_TSOT](#), [TAG_TSRC](#), [TAG_TSSE](#), [TAG_TSST](#), [TAG_TXXX](#), [TAG_UNKN](#), [TAG_WCOM](#), [TAG_WCOP](#), [TAG_WOAF](#), [TAG_WOAR](#), [TAG_WOAS](#), [TAG_WORS](#), [TAG_WPAY](#), [TAG_WPUB](#), [TAG_WXXX](#), [TAGS_TEXTINFORMATION](#), [TEXTENCODING_DEFAULT](#), [TEXTENCODING_ISO_8859_1](#), [TEXTENCODING_URL](#), [TEXTENCODING_UTF16](#), [TEXTENCODING_UTF16BE](#), [TEXTENCODING_UTF8](#)

Constructor Summary

public	ID3V2FrameURLLinkUserDefined (String idStr, int flags)
public	ID3V2FrameURLLinkUserDefined (String idStr)
public	ID3V2FrameURLLinkUserDefined ()

Method Summary

void	deserializeBody (byte[] buffer, int offset, int len)
int	getBodySize ()
String	getDescription ()
String	getURL ()
int	serializeBody (byte[] buffer, int offset)
void	setDescription (String description)

void	setURL (String value)
------	---------------------------------------

Methods inherited from class [com.wowza.wms.media.mp3.model.idtags.ID3V2FrameBase](#)

[byteStringLen](#), [byteStringLen](#), [deserializeFrame](#), [deserializeString](#), [deserializeString](#), [getFlags](#), [getIdStr](#), [getMapIdStr](#), [getSize](#), [serialize](#), [serializeString](#), [serializeString](#), [serializeStringLen](#), [serializeStringLen](#), [setFlags](#), [setIdStr](#), [textEncodingToCharsetName](#), [trimTrailingZero](#)

Methods inherited from class [java.lang.Object](#)

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

Methods inherited from interface [com.wowza.wms.media.mp3.model.idtags.IID3V2Frame](#)

[deserializeBody](#), [getBodySize](#), [getFlags](#), [getIdStr](#), [getMapIdStr](#), [getSize](#), [serialize](#), [serializeBody](#), [setFlags](#), [setIdStr](#)

Constructors

ID3V2FrameURLLinkUserDefined

```
public ID3V2FrameURLLinkUserDefined(String idStr,
                                     int flags)
```

ID3V2FrameURLLinkUserDefined

```
public ID3V2FrameURLLinkUserDefined(String idStr)
```

ID3V2FrameURLLinkUserDefined

```
public ID3V2FrameURLLinkUserDefined()
```

Methods

serializeBody

```
public int serializeBody(byte[] buffer,
                        int offset)
```

deserializeBody

```
public void deserializeBody(byte[] buffer,
                          int offset,
                          int len)
```

(continued from last page)

getBodySize

```
public int getBodySize()
```

getURL

```
public String getURL()
```

setURL

```
public void setURL(String value)
```

getDescription

```
public String getDescription()
```

setDescription

```
public void setDescription(String description)
```

com.wowza.wms.media.mp3.model.idtags

Class ID3V2Utils

java.lang.Object

└─com.wowza.wms.media.mp3.model.idtags.ID3V2Utils

public class ID3V2Utils

extends Object

Constructor Summary	
public	ID3V2Utils()

Method Summary	
static int	byteArrayToIntSafeSync (byte[] b, int offset, int count)
static long	byteArrayToLongSafeSync (byte[] b, int offset, int count)
static void	intToByteArraySafeSync (int value, byte[] buffer, int offset, int size)

Methods inherited from class java.lang.Object	
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait	

Constructors

ID3V2Utils

public ID3V2Utils()

Methods

byteArrayToLongSafeSync

public static long **byteArrayToLongSafeSync**(byte[] b,
int offset,
int count)

(continued from last page)

byteArrayToIntSafeSync

```
public static int byteArrayToIntSafeSync(byte[] b,  
    int offset,  
    int count)
```

intToByteArraySafeSync

```
public static void intToByteArraySafeSync(int value,  
    byte[] buffer,  
    int offset,  
    int size)
```

com.wowza.wms.media.mp3.model.idtags Interface IID3V2Frame

All Known Implementing Classes:
[IID3V2FrameBase](#)

public interface **IID3V2Frame**
extends

Method Summary

void	<code>deserializeBody</code> (byte[] buffer, int offset, int len)
int	<code>getBodySize</code> ()
int	<code>getFlags</code> ()
String	<code>getIdStr</code> ()
String	<code>getMapIdStr</code> ()
int	<code>getSize</code> ()
int	<code>serialize</code> (byte[] buffer, int offset)
int	<code>serializeBody</code> (byte[] buffer, int offset)
void	<code>setFlags</code> (int flags)
void	<code>setIdStr</code> (String idStr)

Methods

getIdStr

public String **getIdStr**()

setIdStr

public void **setIdStr**(String idStr)

(continued from last page)

getMapIdStr

```
public String getMapIdStr()
```

getFlags

```
public int getFlags()
```

setFlags

```
public void setFlags(int flags)
```

getSize

```
public int getSize()
```

getBodySize

```
public int getBodySize()
```

deserializeBody

```
public void deserializeBody(byte[] buffer,  
                             int offset,  
                             int len)
```

serializeBody

```
public int serializeBody(byte[] buffer,  
                           int offset)
```

serialize

```
public int serialize(byte[] buffer,  
                     int offset)
```

Package

com.wowza.wms.mediacache.impl

com.wowza.wms.mediacache.impl Interface IMediaCacheHTTPByteReader

public interface **IMediaCacheHTTPByteReader**
extends

Method Summary

void	<u>close()</u>
MediaCacheHTTPByteReaderResult	<u>head</u> (String path)
void	<u>init</u> ()
MediaCacheHTTPByteReaderResult	<u>read</u> (String path, long pos, int flen)
void	<u>setConnectionTimeout</u> (int connectionTimeout)
void	<u>setDestinationAddress</u> (String host, int port)
void	<u>setMaxRetries</u> (int maxRetries)
void	<u>setProxyAddress</u> (String host, int port)
void	<u>setReadTimeout</u> (int readTimeout)
void	<u>setReceiveBufferSize</u> (int receiveBufferSize)
void	<u>setRequestFullURL</u> (boolean requestFullURL)
void	<u>setSendBufferSize</u> (int sendBufferSize)

Methods

head

```
public MediaCacheHTTPByteReaderResult head(String path)
```

read

```
public MediaCacheHTTPByteReaderResult read(String path,  
      long pos,  
      int flen)
```

(continued from last page)

close

```
public void close()
```

setDestinationAddress

```
public void setDestinationAddress(String host,  
    int port)
```

setProxyAddress

```
public void setProxyAddress(String host,  
    int port)
```

setRequestFullURL

```
public void setRequestFullURL(boolean requestFullURL)
```

setReadTimeout

```
public void setReadTimeout(int readTimeout)
```

setMaxRetries

```
public void setMaxRetries(int maxRetries)
```

setConnectionTimeout

```
public void setConnectionTimeout(int connectionTimeout)
```

setSendBufferSize

```
public void setSendBufferSize(int sendBufferSize)
```

setReceiveBufferSize

```
public void setReceiveBufferSize(int receiveBufferSize)
```

init

```
public void init()
```

com.wowza.wms.mediacache.impl Interface IMediaCacheItemHTTPReaderFactory

public interface **IMediaCacheItemHTTPReaderFactory**
extends **IGenericPoolFactory**

Method Summary

void	<u>addProperties</u> (<u>WMSProperties</u> properties)
<u>IMediaCacheHTTPByteReader</u>	<u>createInstance</u> ()
void	<u>destroyInstance</u> (<u>IMediaCacheHTTPByteReader</u> instance)
int	<u>getConnectionTimeout</u> ()
<u>IMediaCacheItem</u>	<u>getMediaCacheItem</u> ()
int	<u>getPort</u> ()
int	<u>getReadRetries</u> ()
int	<u>getReadTimeout</u> ()
int	<u>getReceiveBufferSize</u> ()
int	<u>getSendBufferSize</u> ()
void	<u>init</u> ()
boolean	<u>isRequestFullURL</u> ()
void	<u>setConnectionTimeout</u> (int connectionTimeout)
void	<u>setDestinationAddress</u> (String host, int port)
void	<u>setMediaCacheItem</u> (<u>IMediaCacheItem</u> mediaCacheItem)
void	<u>setPort</u> (int port)
void	<u>setProxyAddress</u> (String host, int port)
void	<u>setReadRetries</u> (int readRetries)
void	<u>setReadTimeout</u> (int readTimeout)

void	<code>setReceiveBufferSize</code> (int receiveBufferSize)
void	<code>setRequestFullURL</code> (boolean requestFullURL)
void	<code>setSendBufferSize</code> (int sendBufferSize)

Methods inherited from interface `com.wowza.util.IGenericPoolFactory`

`createInstance`, `destroyInstance`

Methods

`setProxyAddress`

```
public void setProxyAddress(String host,  
                           int port)
```

`createInstance`

```
public IMediaCacheHTTPByteReader createInstance()
```

`destroyInstance`

```
public void destroyInstance(IMediaCacheHTTPByteReader instance)
```

`isRequestFullURL`

```
public boolean isRequestFullURL()
```

`setRequestFullURL`

```
public void setRequestFullURL(boolean requestFullURL)
```

`getReadTimeout`

```
public int getReadTimeout()
```

`setReadTimeout`

```
public void setReadTimeout(int readTimeout)
```

getConnectionTimeout

```
public int getConnectionTimeout()
```

setConnectionTimeout

```
public void setConnectionTimeout(int connectionTimeout)
```

getPort

```
public int getPort()
```

setPort

```
public void setPort(int port)
```

getReadRetries

```
public int getReadRetries()
```

setReadRetries

```
public void setReadRetries(int readRetries)
```

getSendBufferSize

```
public int getSendBufferSize()
```

setSendBufferSize

```
public void setSendBufferSize(int sendBufferSize)
```

getReceiveBufferSize

```
public int getReceiveBufferSize()
```

setReceiveBufferSize

```
public void setReceiveBufferSize(int receiveBufferSize)
```

(continued from last page)

setDestinationAddress

```
public void setDestinationAddress(String host,  
    int port)
```

getMediaCacheItem

```
public IMediaCacheItem getMediaCacheItem()
```

setMediaCacheItem

```
public void setMediaCacheItem(IMediaCacheItem mediaCacheItem)
```

init

```
public void init()
```

addProperties

```
public void addProperties(WMSProperties properties)
```

com.wowza.wms.mediache.impl Interface IMediaCacheNameHandler

public interface **IMediaCacheNameHandler**
extends

Method Summary

void	<code>addToEnvMap</code> (String key, String value)
String	<code>getBasePath</code> ()
String	<code>getCanHandleFilter</code> ()
String	<code>getMediaNameExpression</code> ()
String	<code>getPathDelimiter</code> ()
void	<code>setBasePath</code> (String basePath)
void	<code>setCanHandleFilter</code> (String canHandleFilter)
void	<code>setMediaNameExpression</code> (String mediaNameExpression)
void	<code>setPathDelimiter</code> (String pathDelimiter)

Methods

addToEnvMap

```
public void addToEnvMap(String key,  
                        String value)
```

getBasePath

```
public String getBasePath()
```

setBasePath

```
public void setBasePath(String basePath)
```


(continued from last page)

getCanHandleFilter

```
public String getCanHandleFilter()
```

setCanHandleFilter

```
public void setCanHandleFilter(String canHandleFilter)
```

getMediaNameExpression

```
public String getMediaNameExpression()
```

setMediaNameExpression

```
public void setMediaNameExpression(String mediaNameExpression)
```

getPathDelimiter

```
public String getPathDelimiter()
```

setPathDelimiter

```
public void setPathDelimiter(String pathDelimiter)
```

com.wowza.wms.mediache.impl Interface IMediaCacheSourceBasicPerSourceInfo

public interface **IMediaCacheSourceBasicPerSourceInfo**
extends

Method Summary

String	getName()
void	setName(String name)

Methods

setName

public void **setName**(String name)

getName

public String **getName**()

com.wowza.wms.mediache.impl Interface IMediaCacheSourceBasicPerSourceInfoListener

public interface **IMediaCacheSourceBasicPerSourceInfoListener**
extends

Method Summary

void	onPerSourceInfoAdd (IMediaCacheSourceBasicPerSourceInfo perSourceInfo)
void	onPerSourceInfoRemove (IMediaCacheSourceBasicPerSourceInfo perSourceInfo)

Methods

onPerSourceInfoAdd

public void **onPerSourceInfoAdd**([IMediaCacheSourceBasicPerSourceInfo](#) perSourceInfo)

onPerSourceInfoRemove

public void **onPerSourceInfoRemove**([IMediaCacheSourceBasicPerSourceInfo](#) perSourceInfo)

Package

com.wowza.wms.mediacache.model

com.wowza.wms.mediacache.model

Interface IMediaCacheContextMapper

public interface **IMediaCacheContextMapper**
extends

Method Summary	
String	getMediaNameFromContext (IApplicationInstance appInstance, IMediaStream stream, String basePath, String mediaName, String mediaExtension)
void	init (MediaCache mediaCache)

Methods

init
public void **init**(MediaCache mediaCache)

getMediaNameFromContext
public String **getMediaNameFromContext**([IApplicationInstance](#) appInstance, [IMediaStream](#) stream, String basePath, String mediaName, String mediaExtension)

com.wowza.wms.mediacache.model Interface IMediaCacheItem

public interface **IMediaCacheItem**
extends

Field Summary

public static final	PROPS_ITEM_LASTMODIFIED Value: mediaCacheItem.lastModified
public static final	PROPS_ITEM_LENGTH Value: mediaCacheItem.length
public static final	PROPS_ITEM_MEDIANAME Value: mediaCacheItem.mediaName
public static final	PROPS_SOURCE_BLOCKSIZE Value: mediaCacheSource.blockSize
public static final	PROPS_SOURCE_NAME Value: mediaCacheSource.name

Method Summary

void	addPendingWrites (MediaCacheBlockPtr blockPtr)
long	blockSize ()
void	close ()
void	decPendingWriteCount ()
boolean	exists ()
long	getBlockCount ()
MediaCacheBlockPtr	getBlockPtr (long offset, long length)
long	getBlockSize ()
MediaCacheItemInfo	getInfo ()
String	getLocalPath ()

int	<u>getLockCount()</u>
long	<u>getMaxTimeToLive()</u>
<u>IMediaCacheSource</u>	<u>getMediaCacheSource()</u>
java.io.File	<u>getMediaFile()</u>
String	<u>getMediaName()</u>
long	<u>getMinTimeToLive()</u>
long	<u>getPendingWriteCount()</u>
MediaCacheBlockPtr	<u>getPendingWrites()</u> (long blockId)
<u>IMediaCacheStoreItem</u>	<u>getStoreItem()</u>
<u>IMediaCacheStoreItem</u>	<u>getStoreItem()</u> (boolean doLoad)
edu.emory.mathcs.backport.java.util.concurrent.locks.WMSReadWriteLock	<u>getSyncLock()</u>
void	<u>incPendingWriteCount()</u>
boolean	<u>init()</u> (String mediaName, <u>IMediaCacheStoreItem</u> storeItem)
boolean	<u>isBlockInCache()</u> (long block)
boolean	<u>isLocal()</u>
boolean	<u>isReadAheadTrigger()</u> (long pos, int threshold)
long	<u>lastModified()</u>
long	<u>length()</u>
int	<u>lock()</u>
long	<u>posToBlock()</u> (long pos)
void	<u>preload()</u>
long	<u>read()</u> (MediaCacheReadContext readContext, long foffset, byte[] buffer, long boffset, long length)
MediaCacheReadResult	<u>read()</u> (MediaCacheReadContext readContext, long foffset, long length)

void	<u>readAhead()</u>
MediaCacheBlockPtr	<u>removePendingWrites</u> (long blockId)
void	<u>saveBlock</u> (MediaCacheBlockPtr blockPtr)
void	<u>setLastModified</u> (long lastModified)
void	<u>setLength</u> (long length)
void	<u>setLocal</u> (boolean isLocal)
void	<u>setLocalPath</u> (String localPath)
void	<u>setMaxTimeToLive</u> (long maxTimeToLive)
void	<u>setMediaCacheSource</u> (<u>IMediaCacheSource</u> mediaCacheSource)
void	<u>setMinTimeToLive</u> (long minTimeToLive)
void	<u>tiggerReadAhead</u> (long pos)
int	<u>unlock</u> ()
boolean	<u>validate</u> ()
boolean	<u>validate</u> (java.util.Properties props)

Fields

PROPS_ITEM_LENGTH

```
public static final java.lang.String PROPS_ITEM_LENGTH
```

Constant value: **mediaCacheItem.length**

PROPS_ITEM_LASTMODIFIED

```
public static final java.lang.String PROPS_ITEM_LASTMODIFIED
```

Constant value: **mediaCacheItem.lastModified**

PROPS_ITEM_MEDIANAME

```
public static final java.lang.String PROPS_ITEM_MEDIANAME
```

Constant value: **mediaCacheItem.mediaName**

PROPS_SOURCE_NAME

```
public static final java.lang.String PROPS_SOURCE_NAME
```

Constant value: `mediaCacheSource.name`

PROPS_SOURCE_BLOCKSIZE

```
public static final java.lang.String PROPS_SOURCE_BLOCKSIZE
```

Constant value: `mediaCacheSource.blockSize`

Methods

getMediaCacheSource

```
public IMediaCacheSource getMediaCacheSource()
```

setMediaCacheSource

```
public void setMediaCacheSource(IMediaCacheSource mediaCacheSource)
```

lock

```
public int lock()
```

unlock

```
public int unlock()
```

getLockCount

```
public int getLockCount()
```

getSyncLock

```
public edu.emory.mathcs.backport.java.util.concurrent.locks.WMSReadWriteLock  
getSyncLock()
```

init

```
public boolean init(String mediaName,  
    IMediaCacheStoreItem storeItem)
```

(continued from last page)

close

```
public void close()
```

getBlockPtr

```
public MediaCacheBlockPtr getBlockPtr(long offset,  
    long length)
```

saveBlock

```
public void saveBlock(MediaCacheBlockPtr blockPtr)
```

readAhead

```
public void readAhead()
```

getMediaFile

```
public java.io.File getMediaFile()
```

isBlockInCache

```
public boolean isBlockInCache(long block)
```

blockSize

```
public long blockSize()
```

validate

```
public boolean validate()
```

validate

```
public boolean validate(java.util.Properties props)
```

(continued from last page)

getInfo

```
public MediaCacheItemInfo getInfo()
```

getMediaName

```
public String getMediaName()
```

getStoreItem

```
public IMediaCacheStoreItem getStoreItem()
```

getStoreItem

```
public IMediaCacheStoreItem getStoreItem(boolean doLoad)
```

addPendingWrites

```
public void addPendingWrites(MediaCacheBlockPtr blockPtr)
```

removePendingWrites

```
public MediaCacheBlockPtr removePendingWrites(long blockId)
```

getPendingWrites

```
public MediaCacheBlockPtr getPendingWrites(long blockId)
```

incPendingWriteCount

```
public void incPendingWriteCount()
```

decPendingWriteCount

```
public void decPendingWriteCount()
```

getPendingWriteCount

```
public long getPendingWriteCount()
```

(continued from last page)

getMinTimeToLive

```
public long getMinTimeToLive()
```

setMinTimeToLive

```
public void setMinTimeToLive(long minTimeToLive)
```

getMaxTimeToLive

```
public long getMaxTimeToLive()
```

setMaxTimeToLive

```
public void setMaxTimeToLive(long maxTimeToLive)
```

getBlockSize

```
public long getBlockSize()
```

getBlockCount

```
public long getBlockCount()
```

isReadAheadTrigger

```
public boolean isReadAheadTrigger(long pos,  
    int threshold)
```

tiggerReadAhead

```
public void tiggerReadAhead(long pos)
```

posToBlock

```
public long posToBlock(long pos)
```

(continued from last page)

preload

```
public void preload()
```

getLocalPath

```
public String getLocalPath()
```

setLocalPath

```
public void setLocalPath(String localPath)
```

isLocal

```
public boolean isLocal()
```

setLocal

```
public void setLocal(boolean isLocal)
```

exists

```
public boolean exists()
```

length

```
public long length()
```

setLength

```
public void setLength(long length)
```

lastModified

```
public long lastModified()
```

setLastModified

```
public void setLastModified(long lastModified)
```

(continued from last page)

read

```
public long read(MediaCacheReadContext readContext,  
                long foffset,  
                byte[] buffer,  
                long boffset,  
                long length)
```

read

```
public MediaCacheReadResult read(MediaCacheReadContext readContext,  
                                  long foffset,  
                                  long length)
```

com.wowza.wms.mediacache.model Interface IMediaCacheReader

public interface **IMediaCacheReader**
extends

Method Summary

void	<code>close()</code>
long	<code>getFilePointer()</code>
<code>IMediaCacheItem</code>	<code>getMediaCacheItem()</code>
Object	<code>getOpaqueObject()</code>
long	<code>lastModified()</code>
long	<code>length()</code>
void	<code>open()</code>
int	<code>read</code> (byte[] buffer, int offset, int len)
void	<code>seek</code> (long pos)
void	<code>setMediaCacheItem</code> (<code>IMediaCacheItem</code> mediaCacheItem)
void	<code>setOpaqueObject</code> (Object object)

Methods

seek

public void **seek**(long pos)

getFilePointer

public long **getFilePointer**()

(continued from last page)

read

```
public int read(byte[] buffer,  
               int offset,  
               int len)
```

setMediaCacheItem

```
public void setMediaCacheItem(IMediaCacheItem mediaCacheItem)
```

getMediaCacheItem

```
public IMediaCacheItem getMediaCacheItem()
```

open

```
public void open()
```

close

```
public void close()
```

length

```
public long length()
```

lastModified

```
public long lastModified()
```

getOpaqueObject

```
public Object getOpaqueObject()
```

setOpaqueObject

```
public void setOpaqueObject(Object object)
```

com.wowza.wms.mediacache.model

Interface IMediaCacheSource

public interface **IMediaCacheSource**
 extends

Method Summary

boolean	<u>canHandle</u> (String mediaName)
String	<u>getBaseClass</u> ()
int	<u>getDefaultBlockSize</u> ()
String	<u>getDescription</u> ()
long	<u>getMaxTimeToLive</u> ()
MediaCache	<u>getMediaCache</u> ()
long	<u>getMinTimeToLive</u> ()
String	<u>getName</u> ()
MediaCachePerformance Counter	<u>getPerformanceCount</u> ()
<u>WMSProperties</u>	<u>getProperties</u> ()
int	<u>getReadAheadThreshold</u> ()
String	<u>getReaderClass</u> ()
String	<u>getSourcePath</u> ()
boolean	<u>isPassThru</u> ()
boolean	<u>isReadAhead</u> ()
<u>IMediaCacheItem</u>	<u>loadCacheItem</u> (String mediaName, <u>IMediaCacheStoreItem</u> storeItem)
<u>IMediaCacheItem</u>	<u>loadCacheItem</u> (String mediaName, <u>IMediaCacheStoreItem</u> storeItem, long length, long lastModified)
<u>IMediaCacheItem</u>	<u>newCacheItem</u> (String mediaName)

void	<u>setBaseClass</u> (String baseClass)
void	<u>setDefaultBlockSize</u> (int defaultBlockSize)
void	<u>setDescription</u> (String description)
void	<u>setMaxTimeToLive</u> (long maxTimeToLive)
void	<u>setMediaCache</u> (MediaCache mediaCache)
void	<u>setMinTimeToLive</u> (long minTimeToLive)
void	<u>setName</u> (String name)
void	<u>setPassThru</u> (boolean isPassThru)
void	<u>setProperties</u> (<u>WMSProperties</u> properties)
void	<u>setReadAhead</u> (boolean readAhead)
void	<u>setReadAheadThreshold</u> (int readAheadThreshold)
void	<u>setReaderClass</u> (String readerClass)
void	<u>setSourcePath</u> (String sourcePath)

Methods

canHandle

```
public boolean canHandle(String mediaName)
```

newCacheItem

```
public IMediaCacheItem newCacheItem(String mediaName)
```

loadCacheItem

```
public IMediaCacheItem loadCacheItem(String mediaName,  
    IMediaCacheStoreItem storeItem)
```

(continued from last page)

loadCacheItem

```
public IMediaCacheItem loadCacheItem(String mediaName,  
    IMediaCacheStoreItem storeItem,  
    long length,  
    long lastModified)
```

getName

```
public String getName()
```

setName

```
public void setName(String name)
```

getDescription

```
public String getDescription()
```

setDescription

```
public void setDescription(String description)
```

getBaseClass

```
public String getBaseClass()
```

setBaseClass

```
public void setBaseClass(String baseClass)
```

getSourcePath

```
public String getSourcePath()
```

setSourcePath

```
public void setSourcePath(String sourcePath)
```

(continued from last page)

getDefaultBlockSize

```
public int getDefaultBlockSize()
```

setDefaultBlockSize

```
public void setDefaultBlockSize(int defaultBlockSize)
```

getProperties

```
public WMSProperties getProperties()
```

setProperties

```
public void setProperties(WMSProperties properties)
```

getReaderClass

```
public String getReaderClass()
```

setReaderClass

```
public void setReaderClass(String readerClass)
```

getMediaCache

```
public MediaCache getMediaCache()
```

setMediaCache

```
public void setMediaCache(MediaCache mediaCache)
```

getPerformanceCount

```
public MediaCachePerformanceCounter getPerformanceCount()
```

getMinTimeToLive

```
public long getMinTimeToLive()
```

(continued from last page)

setMinTimeToLive

```
public void setMinTimeToLive(long minTimeToLive)
```

getMaxTimeToLive

```
public long getMaxTimeToLive()
```

setMaxTimeToLive

```
public void setMaxTimeToLive(long maxTimeToLive)
```

isReadAhead

```
public boolean isReadAhead()
```

setReadAhead

```
public void setReadAhead(boolean readAhead)
```

getReadAheadThreshold

```
public int getReadAheadThreshold()
```

setReadAheadThreshold

```
public void setReadAheadThreshold(int readAheadThreshold)
```

isPassThru

```
public boolean isPassThru()
```

setPassThru

```
public void setPassThru(boolean isPassThru)
```

com.wowza.wms.mediacache.model Interface IMediaCacheSourceNotify

public interface **IMediaCacheSourceNotify**
extends

Method Summary

void	onMediaCacheSourceCreate (IMediaCacheSource mediaCacheSource)
void	onMediaCacheSourceDestory (IMediaCacheSource mediaCacheSource)

Methods

onMediaCacheSourceCreate

public void **onMediaCacheSourceCreate**([IMediaCacheSource](#) mediaCacheSource)

onMediaCacheSourceDestory

public void **onMediaCacheSourceDestory**([IMediaCacheSource](#) mediaCacheSource)

com.wowza.wms.mediacache.model Interface IMediaCacheStoreItem

public interface **IMediaCacheStoreItem**
extends

Method Summary

byte[]	<u>getBlockMask()</u>
String	<u>getBlockPath()</u>
String	<u>getMediaPath()</u>
java.util.Properties	<u>getProperties()</u>
String	<u>getPropertiesPath()</u>
MediaCacheStore	<u>getStore()</u>
void	<u>initMedia()</u> (long length)
long	<u>length()</u>
void	<u>setBlockMask()</u> (byte[] blockMask)
void	<u>setLength()</u> (long length)
void	<u>setProperties()</u> (java.util.Properties props)
String	<u>toString()</u>
void	<u>updateBlockMask()</u> (byte[] blockMask, long offset, long len)

Methods

getMediaPath

public String **getMediaPath()**

getPropertiesPath

public String **getPropertiesPath()**

(continued from last page)

getBlockPath

```
public String getBlockPath()
```

getProperties

```
public java.util.Properties getProperties()
```

setProperties

```
public void setProperties(java.util.Properties props)
```

getBlockMask

```
public byte[] getBlockMask()
```

updateBlockMask

```
public void updateBlockMask(byte[] blockMask,  
                             long offset,  
                             long len)
```

setBlockMask

```
public void setBlockMask(byte[] blockMask)
```

initMedia

```
public void initMedia(long length)
```

toString

```
public String toString()
```

getStore

```
public MediaCacheStore getStore()
```

length

public long **length**()

setLength

public void **setLength**(long length)

com.wowza.wms.mediacache.model
Interface IMediaCacheStoreNotify

public interface IMediaCacheStoreNotify
extends

Method Summary	
void	onMediaCacheStoreCreate (MediaCacheStore mediaCacheStore)
void	onMediaCacheStoreDestory (MediaCacheStore mediaCacheStore)

Methods

onMediaCacheStoreCreate

public void **onMediaCacheStoreCreate**(MediaCacheStore mediaCacheStore)

onMediaCacheStoreDestory

public void **onMediaCacheStoreDestory**(MediaCacheStore mediaCacheStore)

com.wowza.wms.mediacache.model Interface IMediaCacheStoreTraverseFile

public interface **IMediaCacheStoreTraverseFile**
extends

Method Summary

void	onFile (java.io.File item)
------	--

Methods

onFile

public void **onFile**(java.io.File item)

com.wowza.wms.mediacache.model Interface IMediaCacheStoreTraverseItem

public interface **IMediaCacheStoreTraverseItem**
extends

Method Summary

void	onItem (IMediaCacheStoreItem item)
------	---

Methods

onItem

public void **onItem**([IMediaCacheStoreItem](#) item)

Package

com.wowza.wms.mediacaster

com.wowza.wms.mediacaster

Interface IMediaCaster

public interface **IMediaCaster**
extends

Field Summary

public static final	<u>MEDIACASTERTYPE_LIVEREPEATER</u> Value: 1
public static final	<u>MEDIACASTERTYPE_RTPLIVE</u> Value: 3
public static final	<u>MEDIACASTERTYPE_SHOUTCAST</u> Value: 2
public static final	<u>MEDIACASTERTYPE_UNKNOWN</u> Value: 0
public static final	<u>STREAMTIMEOUTREASON_GOOD</u> Value: 100
public static final	<u>STREAMTIMEOUTREASON_MISSING</u> Value: 101
public static final	<u>STREAMTIMEOUTREASON_NORTSPSESSION</u> Value: 6
public static final	<u>STREAMTIMEOUTREASON_NOSESSION</u> Value: 2
public static final	<u>STREAMTIMEOUTREASON_NOSTREAM</u> Value: 4
public static final	<u>STREAMTIMEOUTREASON_NOTIMEOUT</u> Value: 1
public static final	<u>STREAMTIMEOUTREASON_NOURL</u> Value: 3
public static final	<u>STREAMTIMEOUTREASON_RECONNECTRUNNING</u> Value: 5
public static final	<u>STREAMTIMEOUTREASON_UNKNOWN</u> Value: 0

Method Summary

boolean	<u>doWatchdog()</u> Idle processor
void	<u>forceReset()</u> Force a reset/reconnect of this media caster
<u>IApplicationInstance</u>	<u>getAppInstance()</u> Get the application instance this media caster is associated with
long	<u>getConnectLastAttempt()</u> Get system time in milliseconds of last connection attempt
long	<u>getConnectLastForceReset()</u> Get system time in milliseconds of last time forceReset was called
long	<u>getConnectLastSuccess()</u> Get system time in milliseconds of last connection success
int	<u>getIdleTimeout()</u> Get the idle timeout for this media caster (milliseconds)
Object	<u>getLock()</u> Get the MediaCaster lock for an individual MediaCaster item.
<u>MediaCasterItem</u>	<u>getMediaCasterDef()</u> Get the media caster definition
String	<u>getMediaCasterId()</u> Get the media caster id
<u>MediaCasterStreamItem</u>	<u>getMediaCasterStreamItem()</u> Get the media caster item associated with this media caster
int	<u>getMediaCasterType()</u> Get the media caster type.
int	<u>getReconnectWaitTime()</u> Get the minimum time between reconnect attempts (milliseconds)
<u>IMediaStream</u>	<u>getStream()</u> Get the underlying stream being used by this media caster
long	<u>getStreamLastSeq()</u> Get the AMFPacket sequence number of last watchdog processed packet
long	<u>getStreamMissingTime()</u> Get the time in milliseconds the stream has been missing
long	<u>getStreamRunningIndex()</u> Each time a MediaCaster stream switches from not running to running this value is incremented
int	<u>getStreamTimeout()</u> Get the watchdog stream timeout (milliseconds)
long	<u>getStreamTimeoutLastReset()</u> Get system time in milliseconds of last time stream was reset due to stream timeout (debug)

long	<u>getStreamTimeoutLastTime()</u> Get system time in milliseconds of last time stream was considered in missing state (debug)
int	<u>getStreamTimeoutReason()</u> Get the reason the stream is in timeout condition (debug)
<u>IVHost</u>	<u>getVHost()</u> Get the virtual host associated with this media caster
void	<u>init(MediaCasterStreamItem mediaCasterStreamItem, MediaCasterItem mediaCasterDef, IApplicationInstance appInstance, String mediaCasterId, String streamExt)</u> Initialize the media caster
boolean	<u>isSession()</u> Is there current a session attached to this MediaCaster
boolean	<u>isStream()</u> Is there a stream associated with this MediaCaster
boolean	<u>isStreamIsRunning()</u> Return true if stream is currently running
void	<u>registerPlayer(IMediaStreamPlay player)</u> Register a player with this media caster
void	<u>sessionClosed(org.apache.mina.common.IoSession session)</u> sessionClosed callback
void	<u>sessionOpened(org.apache.mina.common.IoSession session)</u> sessionOpened callback
void	<u>setAppInstance(IApplicationInstance appInstance)</u> Set the application instance this media caster is assoicated with
void	<u>setMediaCasterDef(MediaCasterItem mediaCasterDef)</u> Set the media caster definition
void	<u>setMediaCasterId(String mediaCasterId)</u> Get the media caster id
void	<u>setMediaCasterType(int mediaCasterType)</u> Set the media caster type.
void	<u>setReconnectWaitTime(int reconnectWaitTime)</u> Set the minimum time between reconnect attempts (milliseconds)
void	<u>setStream(IMediaStream stream)</u> Set the underlying stream being used by this media caster
void	<u>setStreamRunning(boolean streamRunning)</u> Called internally to track when streams are running or are stopped.
void	<u>setStreamTimeout(int streamTimeout)</u> Set the watchdog stream timeout (milliseconds)
void	<u>shutdown(boolean isAppInstanceShutdown)</u> Shutdown media caster

void	<code>unregisterPlayer(IMediaStreamPlay player)</code> Unregister a player with this media caster
------	--

Fields

STREAMTIMEOUTREASON_UNKNOWN

`public static final int STREAMTIMEOUTREASON_UNKNOWN`

Constant value: **0**

STREAMTIMEOUTREASON_NOTIMEOUT

`public static final int STREAMTIMEOUTREASON_NOTIMEOUT`

Constant value: **1**

STREAMTIMEOUTREASON_NOSESSION

`public static final int STREAMTIMEOUTREASON_NOSESSION`

Constant value: **2**

STREAMTIMEOUTREASON_NOURL

`public static final int STREAMTIMEOUTREASON_NOURL`

Constant value: **3**

STREAMTIMEOUTREASON_NOSTREAM

`public static final int STREAMTIMEOUTREASON_NOSTREAM`

Constant value: **4**

STREAMTIMEOUTREASON_RECONNECTRUNNING

`public static final int STREAMTIMEOUTREASON_RECONNECTRUNNING`

Constant value: **5**

STREAMTIMEOUTREASON_NORTSPSESSION

`public static final int STREAMTIMEOUTREASON_NORTSPSESSION`

Constant value: **6**

(continued from last page)

STREAMTIMEOUTREASON_GOOD

```
public static final int STREAMTIMEOUTREASON_GOOD
```

Constant value: **100**

STREAMTIMEOUTREASON_MISSING

```
public static final int STREAMTIMEOUTREASON_MISSING
```

Constant value: **101**

MEDIACASTERTYPE_UNKNOWN

```
public static final int MEDIACASTERTYPE_UNKNOWN
```

Constant value: **0**

MEDIACASTERTYPE_LIVEREPEATER

```
public static final int MEDIACASTERTYPE_LIVEREPEATER
```

Constant value: **1**

MEDIACASTERTYPE_SHOUTCAST

```
public static final int MEDIACASTERTYPE_SHOUTCAST
```

Constant value: **2**

MEDIACASTERTYPE_RTPLIVE

```
public static final int MEDIACASTERTYPE_RTPLIVE
```

Constant value: **3**

Methods

init

```
public void init(MediaCasterStreamItem mediaCasterStreamItem,  
                MediaCasterItem mediaCasterDef,  
                ApplicationInstance appInstance,  
                String mediaCasterId,  
                String streamExt)
```

Initialize the media caster

Parameters:

`mediaCasterStreamItem` - media caster item
`mediaCasterDef` - media caster definition
`appInstance` - application instance
`mediaCasterId` - media caster id
`streamExt` - stream ext or prefix

getVHost

```
public IVHost getVHost( )
```

Get the virtual host associated with this media caster

Returns:

virtual host associated with this media caster

getMediaCasterId

```
public String getMediaCasterId( )
```

Get the media caster id

Returns:

media caster id

setMediaCasterId

```
public void setMediaCasterId(String mediaCasterId)
```

Get the media caster id

Parameters:

mediaCasterId - media caster id

getStream

```
public IMediaStream getStream( )
```

Get the underlying stream being used by this media caster

Returns:

underlying stream being used by this media caster

setStream

```
public void setStream(IMediaStream stream)
```

Set the underlying stream being used by this media caster

Parameters:

stream - underlying stream being used by this media caster

getAppInstance

```
public IApplicationInstance getAppInstance( )
```

Get the application instance this media caster is associated with

Returns:

application instance this media caster is associated with

setAppInstance

```
public void setAppInstance(IApplicationInstance appInstance)
```

(continued from last page)

Set the application instance this media caster is assoicated with

Parameters:

appInstance - application instance this media caster is assoicated with

getMediaCasterDef

```
public MediaCasterItem getMediaCasterDef()
```

Get the media caster definition

Returns:

media caster definition

setMediaCasterDef

```
public void setMediaCasterDef(MediaCasterItem mediaCasterDef)
```

Set the media caster definition

Parameters:

mediaCasterDef - media caster definition

shutdown

```
public void shutdown(boolean isAppInstanceShutdown)
```

Shutdown media caster

Parameters:

isAppInstanceShutdown - is server shutdown

registerPlayer

```
public void registerPlayer(IMediaStreamPlay player)
```

Register a player with this media caster

Parameters:

player - player to register

unregisterPlayer

```
public void unregisterPlayer(IMediaStreamPlay player)
```

Unregister a player with this media caster

Parameters:

player - player to unregister

getIdleTimeout

```
public int getIdleTimeout()
```

Get the idle timeout for this media caster (milliseconds)

Returns:

idle timeout for this media caster (milliseconds)

sessionOpened

```
public void sessionOpened(org.apache.mina.common.Session session)
```

sessionOpened callback

Parameters:

session - IO Session

sessionClosed

```
public void sessionClosed(org.apache.mina.common.Session session)
```

sessionClosed callback

Parameters:

session - IO Session

getMediaCasterStreamItem

```
public MediaCasterStreamItem getMediaCasterStreamItem()
```

Get the media caster item associated with this media caster

Returns:

media caster item associated with this media caster

forceReset

```
public void forceReset()
```

Force a reset/reconnect of this media caster

doWatchdog

```
public boolean doWatchdog()
```

Idle processor

Returns:

return true if media caster unloaded due to idle event

getStreamTimeout

```
public int getStreamTimeout()
```

Get the watchdog stream timeout (milliseconds)

Returns:

stream timeout

setStreamTimeout

```
public void setStreamTimeout(int streamTimeout)
```

Set the watchdog stream timeout (milliseconds)

(continued from last page)

Parameters:

streamTimeout - stream timeout

getStreamMissingTime

```
public long getStreamMissingTime()
```

Get the time in milliseconds the stream has been missing

Returns:

time in milliseconds the stream has been missing

getStreamLastSeq

```
public long getStreamLastSeq()
```

Get the AMFPacket sequence number of last watchdog processed packet

Returns:

AMFPacket sequence number

getStreamTimeoutReason

```
public int getStreamTimeoutReason()
```

Get the reason the stream is in timeout condition (debug)

Returns:

reason the stream is in timeout condition (debug)

getStreamTimeoutLastTime

```
public long getStreamTimeoutLastTime()
```

Get system time in milliseconds of last time stream was considered in missing state (debug)

Returns:

time in milliseconds of last time stream was considered in missing

getStreamTimeoutLastReset

```
public long getStreamTimeoutLastReset()
```

Get system time in milliseconds of last time stream was reset due to stream timeout (debug)

Returns:

time in milliseconds of last time stream was reset

isSession

```
public boolean isSession()
```

Is there current a session attached to this MediaCaster

Returns:

true is MediaCaster has session

(continued from last page)

isStream

```
public boolean isStream( )
```

Is there a stream associated with this MediaCaster

Returns:

true if stream associated with this MediaCaster

getConnectLastAttempt

```
public long getConnectLastAttempt( )
```

Get system time in milliseconds of last connection attempt

Returns:

system time in milliseconds of last connection attempt

getConnectLastSuccess

```
public long getConnectLastSuccess( )
```

Get system time in milliseconds of last connection success

Returns:

system time in milliseconds of last connection success

getConnectLastForceReset

```
public long getConnectLastForceReset( )
```

Get system time in milliseconds of last time forceReset was called

Returns:

system time in milliseconds of last time forceReset was called

isStreamIsRunning

```
public boolean isStreamIsRunning( )
```

Return true if stream is currently running

Returns:

true if stream is currently running

getReconnectWaitTime

```
public int getReconnectWaitTime( )
```

Get the minimum time between reconnect attempts (milliseconds)

Returns:

minimum time between reconnect attempts (milliseconds)

setReconnectWaitTime

```
public void setReconnectWaitTime(int reconnectWaitTime)
```

(continued from last page)

Set the minimum time between reconnect attempts (milliseconds)

Parameters:

reconnectWaitTime - minimum time between reconnect attempts (milliseconds)

getMediaCasterType

```
public int getMediaCasterType()
```

Get the media caster type. See IMediaCaster.MEDIACASTERTYPE_*

Returns:

media caster type

setMediaCasterType

```
public void setMediaCasterType(int mediaCasterType)
```

Set the media caster type. See IMediaCaster.MEDIACASTERTYPE_*

Parameters:

mediaCasterType - media caster type

getLock

```
public Object getLock()
```

Get the MediaCaster lock for an individual MediaCaster item.

setStreamRunning

```
public void setStreamRunning(boolean streamRunning)
```

Called internally to track when streams are running or are stopped.

Parameters:

streamRunning - set to true if stream is running

getStreamRunningIndex

```
public long getStreamRunningIndex()
```

Each time a MediaCaster stream switches from not running to running this value is incremented

Returns:

stream running index

com.wowza.wms.mediacaster

Interface IMediaCasterDataReceiver

public interface **IMediaCasterDataReceiver**
extends

IMediaCasterDataReceiver: For internal use only.

Method Summary	
void	onData (org.apache.mina.common.ByteBuffer data) Data callback

Methods

onData

public void **onData**(org.apache.mina.common.ByteBuffer data)

Data callback

Parameters:

data - data

com.wowza.wms.mediacaster Interface IMediaCasterNetConnection

public interface **IMediaCasterNetConnection**
extends

IMediaCasterNetConnection: Internal use only

Method Summary

com.wowza.wms.netconnection.NetConnection	getNetConnection() Receives the INetConnection interface for a live repeater connection
---	--

Methods

getNetConnection

public com.wowza.wms.netconnection.NetConnection **getNetConnection()**

Receives the INetConnection interface for a live repeater connection

Returns:

INetConnection interface

com.wowza.wms.mediacaster Interface IMediaCasterNotify

All Subinterfaces:

[IMediaCasterNotify2](#)

public interface **IMediaCasterNotify**
extends

IMediaCasterNotify: listener interface to MediaCaster system. See IApplicationInstance.addMediaCasterListener().

Method Summary

void	onMediaCasterCreate (IMediaCaster mediaCaster) Invoked when mediaCaster created
void	onMediaCasterDestroy (IMediaCaster mediaCaster) Invoked when MediaCaster destroyed
void	onRegisterPlayer (IMediaCaster mediaCaster, IMediaStreamPlay player) Invoked when a player is added to this mediaCaster
void	onSetSourceStream (IMediaCaster mediaCaster, IMediaStream stream) Invoked when soure stream is set (can be called with stream of null)
void	onUnRegisterPlayer (IMediaCaster mediaCaster, IMediaStreamPlay player) Invoked when a player is removed from this mediaCaster

Methods

onMediaCasterCreate

```
public void onMediaCasterCreate(IMediaCaster mediaCaster)
```

Invoked when mediaCaster created

Parameters:

mediaCaster

onMediaCasterDestroy

```
public void onMediaCasterDestroy(IMediaCaster mediaCaster)
```

Invoked when MediaCaster destroyed

Parameters:

mediaCaster

onRegisterPlayer

```
public void onRegisterPlayer(IMediaCaster mediaCaster,  
    IMediaStreamPlay player)
```

(continued from last page)

Invoked when a player is added to this mediaCaster

Parameters:

mediaCaster
player

onUnRegisterPlayer

```
public void onUnRegisterPlayer(IMediaCaster mediaCaster,  
    IMediaStreamPlay player)
```

Invoked when a player is removed from this mediaCaster

Parameters:

mediaCaster
player

onSetSourceStream

```
public void onSetSourceStream(IMediaCaster mediaCaster,  
    IMediaStream stream)
```

Invoked when source stream is set (can be called with stream of null)

Parameters:

mediaCaster
stream

com.wowza.wms.mediacaster Interface IMediaCasterNotify2

All Superinterfaces:

[IMediaCasterNotify](#)

public interface **IMediaCasterNotify2**

extends [IMediaCasterNotify](#)

Method Summary

void	onConnectFailure (IMediaCaster mediaCaster) Invoked when a connection or reconnection attempt fails
void	onConnectStart (IMediaCaster mediaCaster) Invoked when a connection or reconnection attempt is invoked
void	onConnectSuccess (IMediaCaster mediaCaster) Invoked when a connection or reconnection attempt is successful
void	onStreamStart (IMediaCaster mediaCaster) Invoked when the stream starts receiving media data from the media source.
void	onStreamStop (IMediaCaster mediaCaster) Invoked when the stream stops receiving media data from the media source after the streamTimeout value has passed.

Methods inherited from interface [com.wowza.wms.mediacaster.IMediaCasterNotify](#)

[onMediaCasterCreate](#), [onMediaCasterDestroy](#), [onRegisterPlayer](#), [onSetSourceStream](#), [onUnRegisterPlayer](#)

Methods

onConnectStart

public void **onConnectStart**([IMediaCaster](#) mediaCaster)

Invoked when a connection or reconnection attempt is invoked

Parameters:

mediaCaster

onConnectSuccess

public void **onConnectSuccess**([IMediaCaster](#) mediaCaster)

Invoked when a connection or reconnection attempt is successful

Parameters:

mediaCaster

onConnectFailure

```
public void onConnectFailure(IMediaCaster mediaCaster)
```

Invoked when a connection or reconnection attempt fails

Parameters:

mediaCaster

onStreamStart

```
public void onStreamStart(IMediaCaster mediaCaster)
```

Invoked when the stream starts receiving media data from the media source. This event will only be thrown if the MediaCaster property streamTimeout is set to a non-zero value.

Parameters:

mediaCaster

onStreamStop

```
public void onStreamStop(IMediaCaster mediaCaster)
```

Invoked when the stream stops receiving media data from the media source after the streamTimeout value has passed. This event will only be thrown if the MediaCaster property streamTimeout is set to a non-zero value.

Parameters:

mediaCaster

com.wowza.wms.mediacaster Interface IMediaCasterValidateMediaCaster

All Known Implementing Classes:

[ModuleMediaCasterStreamMonitorAdvanced](#)

public interface **IMediaCasterValidateMediaCaster**
extends

IMediaCasterValidateMediaCaster: interface for implementing stream validators. See
IApplicationInstance.setMediaCasterValidator(IMediaCasterValidateMediaCaster mediaCasterValidator)

Method Summary

boolean	onResetMediaCaster (IApplicationInstance appInstance, IMediaCaster mediaCaster) Called when media caster is reset
boolean	onValidateMediaCaster (IApplicationInstance appInstance, IMediaCaster mediaCaster) Called for each media caster to validate the media caster.
void	onValidateMediaCastersStart (IApplicationInstance appInstance) Called when validation for all streams of an application instance is starting
void	onValidateMediaCastersStop (IApplicationInstance appInstance) Called when validation for all streams of an application instance is done

Methods

onValidateMediaCastersStart

public void **onValidateMediaCastersStart**([IApplicationInstance](#) appInstance)

Called when validation for all streams of an application instance is starting

Parameters:

appInstance - application instance

onValidateMediaCastersStop

public void **onValidateMediaCastersStop**([IApplicationInstance](#) appInstance)

Called when validation for all streams of an application instance is done

Parameters:

appInstance

onValidateMediaCaster

public boolean **onValidateMediaCaster**([IApplicationInstance](#) appInstance, [IMediaCaster](#) mediaCaster)

(continued from last page)

Called for each media caster to validate the media caster. Return true if valid.

Parameters:

appInstance - application instance
mediaCaster - media caster

Returns:

true, if valid

onResetMediaCaster

```
public boolean onResetMediaCaster(IApplicationInstance appInstance,  
    IMediaCaster mediaCaster)
```

Called when media caster is reset

Parameters:

appInstance - application instance
mediaCaster - media caster

Returns:

return true

com.wowza.wms.mediacaster

Class MediaCasterItem

java.lang.Object

└─com.wowza.wms.mediacaster.MediaCasterItem

public class **MediaCasterItem**
extends Object

Constructor Summary

public	MediaCasterItem (String name, String streamType, String baseClass) Media caster item constructor
--------	---

Method Summary

void	clearProperty (String name) Clear property
String	getBaseClass () Get base class
int	getConnectionTimeout () Get connection timeout (milliseconds)
String	getDescription () Get description
static String	getIdString (String name, String liveStreamPacketizer, String liveStreamRepeater) Get id string for this media caster item (not used - returns name unchanged)
int	getKeepAliveTime () Get keep alive time (milliseconds)
String	getName () Get name
WMSProperties	getProperties () Get properties collection
String	getProperty (String name) Get property value
String	getStreamType () Get the stream type
String	idStringToName (String idString) Convert name to id string (not used - returns id string unchanged)
static MediaCasterStreamId	parseIdString (String idString)

void	<code>setBaseClass</code> (String baseClass) Set base class
void	<code>setConnectionTimeout</code> (int connectionTimeout) Set connection timeout (milliseconds)
void	<code>setDescription</code> (String description) Set description
void	<code>setKeepAliveTime</code> (int keepAliveTime) Set keep alive time (milliseconds)
void	<code>setName</code> (String name) Set name
void	<code>setProperty</code> (String name, String value) Set a property
void	<code>setStreamType</code> (String streamType) Set stream type
String	<code>toString</code> () toString

Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Constructors

MediaCasterItem

```
public MediaCasterItem(String name,
                       String streamType,
                       String baseClass)
```

Media caster item constructor

Parameters:

name - media caster name
streamType - stream type
baseClass - base class

Methods

getStreamType

```
public String getStreamType()
```

Get the stream type

Returns:

stream type

(continued from last page)

setStreamType

```
public void setStreamType(String streamType)
```

Set stream type

Parameters:

streamType - stream type

getName

```
public String getName()
```

Get name

Returns:

name

setName

```
public void setName(String name)
```

Set name

Parameters:

name - name

setProperty

```
public void setProperty(String name,  
                        String value)
```

Set a property

Parameters:

name - name

value - value

clearProperty

```
public void clearProperty(String name)
```

Clear property

Parameters:

name - name

getProperty

```
public String getProperty(String name)
```

Get property value

Parameters:

name - name

Returns:

property value

getProperties

```
public WMSProperties getProperties()
```

Get properties collection

Returns:
properties collection

getDescription

```
public String getDescription()
```

Get description

Returns:
description

setDescription

```
public void setDescription(String description)
```

Set description

Parameters:
description - description

getBaseClass

```
public String getBaseClass()
```

Get base class

Returns:
base class

setBaseClass

```
public void setBaseClass(String baseClass)
```

Set base class

Parameters:
baseClass - base class

getConnectionTimeout

```
public int getConnectionTimeout()
```

Get connection timeout (milliseconds)

Returns:
connection timeout (milliseconds)

setConnectionTimeout

```
public void setConnectionTimeout(int connectionTimeout)
```

(continued from last page)

Set connection timeout (milliseconds)

Parameters:

connectionTimeout - connection timeout (milliseconds)

getKeepAliveTime

```
public int getKeepAliveTime()
```

Get keep alive time (milliseconds)

Returns:

keep alive time (milliseconds)

setKeepAliveTime

```
public void setKeepAliveTime(int keepAliveTime)
```

Set keep alive time (milliseconds)

Parameters:

keepAliveTime - keep alive time (milliseconds)

getIdString

```
public static String getIdString(String name,  
    String liveStreamPacketizer,  
    String liveStreamRepeater)
```

Get id string for this media caster item (not used - returns name unchanged)

Parameters:

name - name

Returns:

id string for this media caster item

parseIdString

```
public static MediaCasterStreamId parseIdString(String idString)
```

idStringToName

```
public String idStringToName(String idString)
```

Convert name to id string (not used - returns id string unchanged)

Parameters:

idString - id string

Returns:

name

toString

```
public String toString()
```

(continued from last page)

toString

com.wowza.wms.mediacaster Class MediaCasterList

java.lang.Object

└─com.wowza.wms.mediacaster.MediaCasterList

public class **MediaCasterList**
extends Object

Constructor Summary

public	<u>MediaCasterList()</u> Constructor
--------	---

Method Summary

<u>MediaCasterItem</u>	<u>getMediaCasterDef(String name)</u> Get media caster definition by name
java.util.Map	<u>getMediaCasterDefs()</u> Get map of media caster items
java.util.List	<u>getMediaCasterNames()</u> Get list of media caster names

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

MediaCasterList

public **MediaCasterList()**

Constructor

Methods

getMediaCasterDefs

public java.util.Map **getMediaCasterDefs()**

Get map of media caster items

Returns:

map of media caster items

(continued from last page)

getMediaCasterNames

```
public java.util.List getMediaCasterNames()
```

Get list of media caster names

Returns:

list of media caster names

getMediaCasterDef

```
public MediaCasterItem getMediaCasterDef(String name)
```

Get media caster definition by name

Parameters:

name - name

Returns:

media caster definition

com.wowza.wms.mediacaster

Class MediaCasterSettings



public class **MediaCasterSettings**
extends Object

Constructor Summary

public	MediaCasterSettings()
--------	---------------------------------------

Method Summary

HostPortConfig	getMediaCasterHostPortConfig() Get media caster host port config
int	getMediaCasterProcessorCount() Get the thread count use for this host port
void	setMediaCasterHostPortConfig (HostPortConfig mediaCasterHostPortConfig) Set media caster host port config
void	setMediaCasterProcessorCount (int mediaCasterProcessorCount) Set the thread count for this processor

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

MediaCasterSettings
public **MediaCasterSettings()**

Methods

getMediaCasterHostPortConfig
public HostPortConfig **getMediaCasterHostPortConfig()**

Get media caster host port config

Returns:
host port config

setMediaCasterHostPortConfig

```
public void setMediaCasterHostPortConfig(HostPortConfig mediaCasterHostPortConfig)
```

Set media caster host port config

Parameters:

mediaCasterHostPortConfig

getMediaCasterProcessorCount

```
public int getMediaCasterProcessorCount()
```

Get the thread count use for this host port

Returns:

thread count use for this host port

setMediaCasterProcessorCount

```
public void setMediaCasterProcessorCount(int mediaCasterProcessorCount)
```

Set the thread count for this processor

Parameters:

mediaCasterProcessorCount - thread count use for this host port

com.wowza.wms.mediacaster Class MediaCasterStreamItem

java.lang.Object

└─com.wowza.wms.mediacaster.MediaCasterStreamItem

public class **MediaCasterStreamItem**
extends Object

Constructor Summary

public	MediaCasterStreamItem (long uniqueId) Media caster item constructor
--------	--

Method Summary

void	acquire () Increment acquire lock count for this media caster item
void	acquireAndRelease () Increment then decrement acquire lock count for this media caster item
boolean	doWatchdog () Do idle processing
String	getLiveStreamPacketizer () Get the live stream packetizer for this media caster stream item
String	getLiveStreamRepeater () Get the live stream repeater for this media caster stream item
Object	getLock ()
int	getLockCount () Get the current number of acquire locks on this media caster item
IMediaCaster	getMediaCaster () Get the underlying IMediaCaster interface for this MediaCaster
String	getMediaCasterId () Get this media caster item id
int	getPlayerCount () Get the current number of players associated with this media caster item
String	getStreamExt ()
long	getUniqueId ()

void	<code>init</code> (String mediaCasterId, String streamExt, <code>MediaCasterItem</code> mediaCasterDef, <code>MediaCasterStreamMap</code> parent, String liveStreamPacketizer, String liveStreamRepeater) Initialize the media caster item (internal use)
boolean	<code>isShutdownOnRelease</code> () On last release shutdown the stream even if clients are connected
boolean	<code>isValid</code> ()
void	<code>registerPlayer</code> (<code>IMediaStreamPlay</code> player) Register a player with a media caster item (internal use)
void	<code>release</code> () Decrement acquire lock count for this media caster item
void	<code>reset</code> () Force a reconnect or reset for this media caster item
void	<code>setLiveStreamPacketizer</code> (String liveStreamPacketizer) Set the live stream packetizer for this media caster stream item
void	<code>setLiveStreamRepeater</code> (String liveStreamRepeater) Set the live stream repeater for this media caster stream item
void	<code>setShutdownOnRelease</code> (boolean shutdownOnRelease) On last release shutdown the stream even if clients are connected
void	<code>setStreamExt</code> (String streamExt)
void	<code>setValid</code> (boolean isValid)
void	<code>shutdown</code> (boolean isAppInstanceShutdown) Shutdown this media caster item
void	<code>unregisterPlayer</code> (<code>IMediaStreamPlay</code> player) Unregister a player with a media caster item (internal use)

Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Constructors

MediaCasterStreamItem

```
public MediaCasterStreamItem(long uniqueId)
```

Media caster item constructor

Methods

(continued from last page)

getLock

```
public Object getLock()
```

getUniqueId

```
public long getUniqueId()
```

isValid

```
public boolean isValid()
```

setValid

```
public void setValid(boolean isValid)
```

init

```
public void init(String mediaCasterId,  
                String streamExt,  
                MediaCasterItem mediaCasterDef,  
                MediaCasterStreamMap parent,  
                String liveStreamPacketizer,  
                String liveStreamRepeater)
```

Initialize the media caster item (internal use)

Parameters:

mediaCasterId - media caster id
streamExt - stream extension or prefix
mediaCasterDef - media caster definition
parent - parent map

getMediaCasterId

```
public String getMediaCasterId()
```

Get this media caster item id

Returns:

media caster item id

getMediaCaster

```
public IMediaCaster getMediaCaster()
```

Get the underlying IMediaCaster interface for this MediaCaster

Returns:

underlying IMediaCaster interface

registerPlayer

```
public void registerPlayer(IMediaStreamPlay player)
```

Register a player with a media caster item (internal use)

Parameters:

player - player to register

unregisterPlayer

```
public void unregisterPlayer(IMediaStreamPlay player)
```

Unregister a player with a media caster item (internal use)

Parameters:

player - player to unregister

doWatchdog

```
public boolean doWatchdog()
```

Do idle processing

Returns:

return true if caused shutdown of item

reset

```
public void reset()
```

Force a reconnect or reset for this media caster item

shutdown

```
public void shutdown(boolean isAppInstanceShutdown)
```

Shutdown this media caster item

Parameters:

isAppInstanceShutdown - is this due to application shutdown

getPlayerCount

```
public int getPlayerCount()
```

Get the current number of players associated with this media caster item

Returns:

current number of players associated with this media caster item

getLockCount

```
public int getLockCount()
```

Get the current number of acquire locks on this media caster item

(continued from last page)

Returns:

current number of acquire locks on this media caster item

acquireAndRelease

```
public void acquireAndRelease()
```

Increment then decrement acquire lock count for this media caster item

acquire

```
public void acquire()
```

Increment acquire lock count for this media caster item

release

```
public void release()
```

Decrement acquire lock count for this media caster item

getStreamExt

```
public String getStreamExt()
```

setStreamExt

```
public void setStreamExt(String streamExt)
```

isShutdownOnRelease

```
public boolean isShutdownOnRelease()
```

On last release shutdown the stream even if clients are connected

Returns:

true if shutting down on release

setShutdownOnRelease

```
public void setShutdownOnRelease(boolean shutdownOnRelease)
```

On last release shutdown the stream even if clients are connected

Parameters:

shutdownOnRelease - true if shutting down on release

getLiveStreamPacketizer

```
public String getLiveStreamPacketizer()
```

Get the live stream packetizer for this media caster stream item

Returns:

(continued from last page)

live stream packetizer

setLiveStreamPacketizer

```
public void setLiveStreamPacketizer(String liveStreamPacketizer)
```

Set the live stream packetizer for this media caster stream item

Parameters:

liveStreamPacketizer - live stream packetizer

getLiveStreamRepeater

```
public String getLiveStreamRepeater()
```

Get the live stream repeater for this media caster stream item

Returns:

live stream repeater

setLiveStreamRepeater

```
public void setLiveStreamRepeater(String liveStreamRepeater)
```

Set the live stream repeater for this media caster stream item

Parameters:

liveStreamRepeater - live stream repeater

com.wowza.wms.mediacaster Class MediaCasterStreamManager

java.lang.Object

└-com.wowza.wms.mediacaster.MediaCasterStreamManager

public class **MediaCasterStreamManager**
extends Object

Constructor Summary

public	MediaCasterStreamManager (MediaCasterStreamMap mediaCasterStreamMap)
--------	---

Method Summary

String[]	getStreamArray () Get a list of active streams
java.util.List	getStreamList () Get a list of active streams
boolean	startStream (String streamName, String mediaCasterType) Start a stream by name
boolean	stopStream (String streamName) Stop a stream by name
boolean	streamExists (String streamName) Returns true if stream exists

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

MediaCasterStreamManager

public **MediaCasterStreamManager**([MediaCasterStreamMap](#) mediaCasterStreamMap)

Methods

getStreamArray

public String[] **getStreamArray**()

Get a list of active streams

(continued from last page)

Returns:

list of active streams

getStreamList

```
public java.util.List getStreamList()
```

Get a list of active streams

Returns:

list of active streams

streamExists

```
public boolean streamExists(String streamName)
```

Returns true if stream exists

Parameters:

streamName - stream name

Returns:

true if stream exists

stopStream

```
public boolean stopStream(String streamName)
```

Stop a stream by name

Parameters:

streamName - stream name

Returns:

true is successful

startStream

```
public boolean startStream(String streamName,  
    String mediaCasterType)
```

Start a stream by name

Parameters:

streamName - stream name

mediaCasterType - MediaCaster type as defined in the name field of conf/MediaCasters.xml

Returns:

true is successful

com.wowza.wms.mediacaster

Class MediaCasterStreamMap

java.lang.Object

└─com.wowza.wms.mediacaster.MediaCasterStreamMap

public class **MediaCasterStreamMap**
extends Object

Constructor Summary

public	MediaCasterStreamMap (IApplicationInstance appInstance) Create a new mediacaster map
--------	--

Method Summary

MediaCasterStreamItem	acquire (String streamName) Increment the lock count for a media caster item (so that it remains loaded even if all listeners go away).
MediaCasterStreamItem	acquire (String inStreamName, MediaCasterItem mediaCasterDef) Increment the lock count for a media caster item (so that it remains loaded even if all listeners go away).
MediaCasterStreamItem	acquire (String inStreamName, String streamType) Increment the lock count for a media caster item (so that it remains loaded even if all listeners go away).
MediaCasterStreamItem	acquire (String streamName, String liveStreamPacketizer, String liveStreamRepeater) Increment the lock count for a media caster item (so that it remains loaded even if all listeners go away).
MediaCasterStreamItem	acquire (String inStreamName, String liveStreamPacketizer, String liveStreamRepeater, MediaCasterItem mediaCasterDef) Increment the lock count for a media caster item (so that it remains loaded even if all listeners go away).
MediaCasterStreamItem	acquire (String inStreamName, String liveStreamPacketizer, String liveStreamRepeater, String streamType) Increment the lock count for a media caster item (so that it remains loaded even if all listeners go away).
MediaCasterStreamItem	acquireAndRelease (String inStreamName, String liveStreamPacketizer, String liveStreamRepeater, MediaCasterItem mediaCasterDef) Increment and then decrement the lock count for a media caster item (so that it is loaded and will stay loaded for at least the KeepAliveTime).
void	addStreamSrcToMediaCaster (long streamSrc, String mediaCasterId)
void	clearStreamSrcToMediaCaster (long streamSrc)
void	doWatchdog () Do periodic idle time processing

IApplicationInstance	getApplicationInstance() Get the parent application instance for this map
edu.emory.mathcs.backport.java.util.concurrent.locks.WMSReadWriteLock	getLock() Get the read/write lock for this interface
MediaCasterStreamItem	getMediaCaster(String streamName) Get media caster item based on given stream name
MediaCasterStreamItem	getMediaCaster(String streamName, String liveStreamPacketizer, String liveStreamRepeater) Get media caster item based on given stream name
int	getMediaCasterCount() Get the number of mediacasters current running
java.util.List	getMediaCasterNames() Get a list of all the currently running media caster names
java.util.List	getMediaCasterStreamItems()
MediaCasterStreamManager	getStreamManager() Get the stream manager interface for managing the starting and stopping of streams
void	registerPlayer(IMediaStreamPlay player, MediaCasterItem mediaCasterDef) Register a player to a media caster item (internal use)
void	release(MediaCasterStreamItem mediaCasterStreamItem) Decrement lock count on media caster item
void	release(MediaCasterStreamItem mediaCasterStreamItem, boolean removeIfZero)
void	remove(MediaCasterStreamItem mediaCasterStreamItem)
void	shutdown(boolean isAppInstanceShutdown) Shutdown this media caster and close all running media casters
String	streamSrcToMediaCaster(long streamSrc)
void	unregisterPlayer(IMediaStreamPlay player, MediaCasterItem mediaCasterDef) Unregister a player to a media caster item (internal use)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

MediaCasterStreamMap

```
public MediaCasterStreamMap(IApplicationInstance appInstance)
```

(continued from last page)

Create a new mediacaster map

Parameters:

appInstance - application instance

Methods

getLock

```
public edu.emory.mathcs.backport.java.util.concurrent.locks.WMSReadWriteLock getLock()
```

Get the read/write lock for this interface

Returns:

read/write lock

getStreamManager

```
public MediaCasterStreamManager getStreamManager()
```

Get the stream manager interface for managing the starting and stopping of streams

Returns:

stream manager interface

getApplicationInstance

```
public IApplicationInstance getApplicationInstance()
```

Get the parent application instance for this map

Returns:

parent application instance for this map

getMediaCasterCount

```
public int getMediaCasterCount()
```

Get the number of mediacasters current running

Returns:

number of mediacasters current running

getMediaCasterStreamItems

```
public java.util.List getMediaCasterStreamItems()
```

doWatchdog

```
public void doWatchdog()
```

Do periodic idle time processing

(continued from last page)

shutdown

```
public void shutdown(boolean isAppInstanceShutdown)
```

Shutdown this media caster and close all running media casters

Parameters:

isAppInstanceShutdown - is this due to application shutdown

getMediaCasterNames

```
public java.util.List getMediaCasterNames()
```

Get a list of all the currently running media caster names

Returns:

list of all the currently running media caster names

getMediaCaster

```
public MediaCasterStreamItem getMediaCaster(String streamName)
```

Get media caster item based on given stream name

Parameters:

streamName - stream name

Returns:

media caster item

getMediaCaster

```
public MediaCasterStreamItem getMediaCaster(String streamName,  
String liveStreamPacketizer,  
String liveStreamRepeater)
```

Get media caster item based on given stream name

Parameters:

streamName - stream name

liveStreamPacketizer - live stream packetizer name

liveStreamRepeater - live stream repeater name

Returns:

media caster item

acquire

```
public MediaCasterStreamItem acquire(String streamName,  
String liveStreamPacketizer,  
String liveStreamRepeater)
```

Increment the lock count for a media caster item (so that it remains loaded even if all listeners go away). Will load the media caster if not already loaded. Uses default application instance stream type.

Parameters:

streamName - stream name

liveStreamPacketizer - live stream packetizer name

liveStreamRepeater - live stream repeater name

(continued from last page)

Returns:

media caster item

acquire

```
public MediaCasterStreamItem acquire(String streamName)
```

Increment the lock count for a media caster item (so that it remains loaded even if all listeners go away). Will load the media caster if not already loaded. Uses default application instance stream type.

Parameters:

streamName - stream name

Returns:

media caster item

acquire

```
public MediaCasterStreamItem acquire(String inStreamName,  
    MediaCasterItem mediaCasterDef)
```

Increment the lock count for a media caster item (so that it remains loaded even if all listeners go away). Will load the media caster if not already loaded. This API allow loading of a MediaCaster into a non-MediaCaster stream type such as the "live" or "liverepeater-origin" stream type.

Parameters:

inStreamName - stream name

mediaCasterDef - MediaCaster definition

Returns:

media caster item

acquire

```
public MediaCasterStreamItem acquire(String inStreamName,  
    String liveStreamPacketizer,  
    String liveStreamRepeater,  
    MediaCasterItem mediaCasterDef)
```

Increment the lock count for a media caster item (so that it remains loaded even if all listeners go away). Will load the media caster if not already loaded. This API allow loading of a MediaCaster into a non-MediaCaster stream type such as the "live" or "liverepeater-origin" stream type.

Parameters:

inStreamName - stream name

liveStreamPacketizer - live stream packetizer name

liveStreamRepeater - live stream repeater name

mediaCasterDef - MediaCaster definition

Returns:

media caster item

acquireAndRelease

```
public MediaCasterStreamItem acquireAndRelease(String inStreamName,  
    String liveStreamPacketizer,  
    String liveStreamRepeater,  
    MediaCasterItem mediaCasterDef)
```

(continued from last page)

Increment and then decrement the lock count for a media caster item (so that it is loaded and will stay loaded for at least the KeepAliveTime). Will load the media caster if not already loaded. This API allow loading of a MediaCaster into a non-MediaCaster stream type such as the "live" or "liverepeater-origin" stream type.

Parameters:

inStreamName - stream name
 liveStreamPacketizer - live stream packetizer name
 liveStreamRepeater - live stream repeater name
 mediaCasterDef - MediaCaster definition

Returns:

media caster item

acquire

```
public MediaCasterStreamItem acquire(String inStreamName,  
    String streamType)
```

Increment the lock count for a media caster item (so that it remains loaded even if all listeners go away). Will load the media caster if not already loaded.

Parameters:

inStreamName - stream name
 streamType - stream type to use (null will use default application streamType)

Returns:

media caster item

acquire

```
public MediaCasterStreamItem acquire(String inStreamName,  
    String liveStreamPacketizer,  
    String liveStreamRepeater,  
    String streamType)
```

Increment the lock count for a media caster item (so that it remains loaded even if all listeners go away). Will load the media caster if not already loaded.

Parameters:

inStreamName - stream name
 liveStreamPacketizer - live stream packetizer name
 liveStreamRepeater - live stream repeater name
 streamType - stream type to use (null will use default application streamType)

Returns:

media caster item

release

```
public void release(MediaCasterStreamItem mediaCasterStreamItem)
```

Decrement lock count on media caster item

Parameters:

mediaCasterStreamItem - media caster item to decrement

release

```
public void release(MediaCasterStreamItem mediaCasterStreamItem,  
    boolean removeIfZero)
```


(continued from last page)

remove

```
public void remove(MediaCasterStreamItem mediaCasterStreamItem)
```

streamSrcToMediaCaster

```
public String streamSrcToMediaCaster(long streamSrc)
```

addStreamSrcToMediaCaster

```
public void addStreamSrcToMediaCaster(long streamSrc,  
    String mediaCasterId)
```

clearStreamSrcToMediaCaster

```
public void clearStreamSrcToMediaCaster(long streamSrc)
```

registerPlayer

```
public void registerPlayer(IMediaStreamPlay player,  
    MediaCasterItem mediaCasterDef)
```

Register a player to a media caster item (internal use)

Parameters:

player - player to register
mediaCasterDef - media caster definition

unregisterPlayer

```
public void unregisterPlayer(IMediaStreamPlay player,  
    MediaCasterItem mediaCasterDef)
```

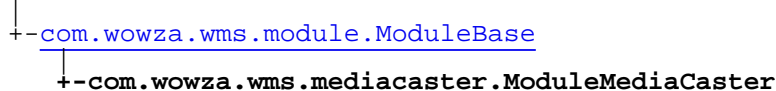
Unregister a player to a media caster item (internal use)

Parameters:

player - player to unregister
mediaCasterDef - media caster definition

com.wowza.wms.mediacaster Class ModuleMediaCaster

java.lang.Object



All Implemented Interfaces:

[IModuleOnApp](#)

public class **ModuleMediaCaster**
 extends [ModuleBase](#)
 implements [IModuleOnApp](#)

ModuleMediaCaster: Module for manipulating media casters through a Flash UI.

Fields inherited from class [com.wowza.wms.module.ModuleBase](#)

[CALLBACK_PARAM1](#), [CALLBACK_PARAM10](#), [CALLBACK_PARAM2](#), [CALLBACK_PARAM3](#), [CALLBACK_PARAM4](#),
[CALLBACK_PARAM5](#), [CALLBACK_PARAM6](#), [CALLBACK_PARAM7](#), [CALLBACK_PARAM8](#), [CALLBACK_PARAM9](#), [PARAM1](#),
[PARAM10](#), [PARAM2](#), [PARAM3](#), [PARAM4](#), [PARAM5](#), [PARAM6](#), [PARAM7](#), [PARAM8](#), [PARAM9](#), [PARAMMETHODNAME](#),
[PLAYTRANSITION_APPEND](#), [PLAYTRANSITION_APPEND_IMMEDIATE](#), [PLAYTRANSITION_RESET](#),
[PLAYTRANSITION_RESET_IMMEDIATE](#), [PLAYTRANSITION_STOP](#), [PLAYTRANSITION_SWAP](#),
[PLAYTRANSITION_SWITCH](#), [PLAYTRANSITION_UNKNOWN](#), [PLAYTRANSITIONSTR_APPEND](#),
[PLAYTRANSITIONSTR_RESET](#), [PLAYTRANSITIONSTR_STOP](#), [PLAYTRANSITIONSTR_SWAP](#),
[PLAYTRANSITIONSTR_SWITCH](#), [PLAYTRANSITIONSTR_UNKNOWN](#)

Constructor Summary

public	ModuleMediaCaster ()
--------	--------------------------------------

Method Summary

void	acquireMediaCaster (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) Increment the lock count of a media caster stream.
void	getLockCount (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) Get the current lock count for a stream
void	getPlayerCount (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) Get the numbers of players associated with a particular media caster
void	getStreamNames (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) Get an array of stream names that are media casters associate with this application instance (returned as AMFDataArray)
void	onAppStart (IApplicationInstance appInstance) onAppStart

void	onAppStop (IApplicationInstance appInstance) onAppStop
void	releaseMediaCaster (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) Decrement the lock count of a media caster stream
void	resetStream (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) Reset a media caster stream
void	shutdownStream (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) Force shutdown a media caster stream

Methods inherited from class [com.wowza.wms.module.ModuleBase](#)

[getAppInstance](#), [getApplication](#), [getCallbackParamCount](#), [getLogger](#), [getParam](#), [getParamBoolean](#), [getParamBoolean](#), [getParamCount](#), [getParamDate](#), [getParamDouble](#), [getParamDouble](#), [getParamInt](#), [getParamInt](#), [getParamLong](#), [getParamLong](#), [getParamMixedArray](#), [getParamObj](#), [getParamString](#), [getParamString](#), [getParamType](#), [getStream](#), [getVHost](#), [invokePrevious](#), [invokePrevious](#), [isSendResult](#), [sendClientOnStatusError](#), [sendResult](#), [sendResult](#), [sendResult](#), [sendResult](#), [sendResult](#), [sendStreamOnStatusError](#)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface [com.wowza.wms.module.IModuleOnApp](#)

[onAppStart](#), [onAppStop](#)

Constructors

ModuleMediaCaster

```
public ModuleMediaCaster()
```

Methods

onAppStart

```
public void onAppStart(IApplicationInstance appInstance)
```

onAppStart

onAppStop

```
public void onAppStop(IApplicationInstance appInstance)
```

onAppStop

getLockCount

```
public void getLockCount(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Get the current lock count for a stream

Parameters:

client - client
function - function
params - {streamName}

getPlayerCount

```
public void getPlayerCount(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Get the numbers of players associated with a particular media caster

Parameters:

client - client
function - function
params - {streamName}

getStreamNames

```
public void getStreamNames(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Get an array of stream names that are media casters associate with this application instance (returned as AMFDataArray)

Parameters:

client - client
function - function
params - (no params)

resetStream

```
public void resetStream(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Reset a media caster stream

Parameters:

client - client
function - function
params - {streamName}

shutdownStream

```
public void shutdownStream(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Force shutdown a media caster stream

(continued from last page)

Parameters:

client - client
function - function
params - {streamName}

acquireMediaCaster

```
public void acquireMediaCaster(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Increment the lock count of a media caster stream. If the media caster does not exist create it and connect.

Parameters:

client - client
function - function
params - {streamName, streamType [optional]}

releaseMediaCaster

```
public void releaseMediaCaster(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Decrement the lock count of a media caster stream

Parameters:

client - client
function - function
params - {streamName}

Package

com.wowza.wms.mediacaster.rtp

com.wowza.wms.mediacaster.rtp Interface IRTPSessionDescriptionDataProvider

public interface **IRTPSessionDescriptionDataProvider**
extends

IRTPSessionDescriptionDataProvider: Internal use.

Method Summary

boolean	<code>doIdle</code> (MediaCaster mediaCaster)
RTPSessionDescription Data	<code>getSessionDescriptionData</code> (IApplicationInstance appInstance, String streamName, int retryCount, IRTPSessionDescriptionSessionHandler handler, IMediaCaster mediaCaster)
void	<code>init</code> (MediaCaster mediaCaster)
void	<code>sessionStart</code> (RTPSession rtpSession)
void	<code>sessionStop</code> (RTPSession rtpSession)

Methods

init

```
public void init(MediaCaster mediaCaster)
```

getSessionDescriptionData

```
public RTPSessionDescriptionData getSessionDescriptionData(IApplicationInstance appInstance,  
    String streamName,  
    int retryCount,  
    IRTPSessionDescriptionSessionHandler handler,  
    IMediaCaster mediaCaster)
```

sessionStart

```
public void sessionStart(RTPSession rtpSession)
```

sessionStop

```
public void sessionStop(RTPSession rtpSession)
```

(continued from last page)

doIdle

```
public boolean doIdle(MediaCaster mediaCaster)
```

com.wowza.wms.mediacaster.rtp Interface IRTPSessionDescriptionSessionHandler

public interface **IRTPSessionDescriptionSessionHandler**
extends

IRTPSessionDescriptionSessionHandler: Internal use.

Method Summary

void	onDisconnect()
------	--------------------------------

Methods

onDisconnect

public void **onDisconnect()**

Package

com.wowza.wms.mediacaster.shoutcast

com.wowza.wms.mediacaster.shoutcast Interface IShoutCastFrameReceiver

public interface **IShoutCastFrameReceiver**
extends

IShoutCastFrameReceiver: Internal use.

Method Summary

void	<code>onCodecConfigAAC</code> (com.wowza.wms.media.aac.AACFrame frame, byte[] buffer, long offset)
void	<code>onFrameAAC</code> (com.wowza.wms.media.aac.AACFrame frame, byte[] buffer, long offset)
void	<code>onFrameMP3</code> (int frequency, int samplesPerFrame, int channels, byte[] syncHeader, byte[] packetHeader, byte[] frameData)
void	<code>onHeaderData</code> (java.util.Map headerMap)
void	<code>onMetaData</code> (java.util.Map metaMap)
void	<code>onTrim</code> ()

Methods

onFrameMP3

```
public void onFrameMP3(int frequency,
    int samplesPerFrame,
    int channels,
    byte[] syncHeader,
    byte[] packetHeader,
    byte[] frameData)
```

onCodecConfigAAC

```
public void onCodecConfigAAC(com.wowza.wms.media.aac.AACFrame frame,
    byte[] buffer,
    long offset)
```

onFrameAAC

```
public void onFrameAAC(com.wowza.wms.media.aac.AACFrame frame,
    byte[] buffer,
    long offset)
```

(continued from last page)

onTrim

```
public void onTrim()
```

onHeaderData

```
public void onHeaderData(java.util.Map headerMap)
```

onMetaData

```
public void onMetaData(java.util.Map metaMap)
```

com.wowza.wms.mediacaster.shoutcast

Interface IShoutCastMetaDataListener

public interface IShoutCastMetaDataListener

extends

IShoutCastMetaDataListener: Internal use.

Method Summary	
void	addMetaDataListener (IShoutCastMetaDataNotify listener)
boolean	removeMetaDataListener (IShoutCastMetaDataNotify listener)

Methods

addMetaDataListener

public void **addMetaDataListener**([IShoutCastMetaDataNotify](#) listener)

removeMetaDataListener

public boolean **removeMetaDataListener**([IShoutCastMetaDataNotify](#) listener)

com.wowza.wms.mediacaster.shoutcast Interface IShoutCastMetaDataNotify

public interface **IShoutCastMetaDataNotify**
extends

IShoutCastMetaDataNotify: Internal use.

Method Summary

void	onAACEncodeInfo (IMediaCaster mediaCaster, int frequency, int channels, int samplesPerFrame)
void	onHeaderData (IMediaCaster mediaCaster, java.util.Map headerMap)
void	onMetaData (IMediaCaster mediaCaster, java.util.Map metaMap)
void	onMP3EncodeInfo (IMediaCaster mediaCaster, int frequency, int channels, int samplesPerFrame)

Methods

onHeaderData

```
public void onHeaderData(IMediaCaster mediaCaster,  
    java.util.Map headerMap)
```

onMetaData

```
public void onMetaData(IMediaCaster mediaCaster,  
    java.util.Map metaMap)
```

onAACEncodeInfo

```
public void onAACEncodeInfo(IMediaCaster mediaCaster,  
    int frequency,  
    int channels,  
    int samplesPerFrame)
```

onMP3EncodeInfo

```
public void onMP3EncodeInfo(IMediaCaster mediaCaster,  
    int frequency,  
    int channels,  
    int samplesPerFrame)
```

(continued from last page)

Package

com.wowza.wms.medialist

com.wowza.wms.medialist Class MediaList

java.lang.Object

└─com.wowza.wms.medialist.MediaList

public class **MediaList**
extends Object

Field Summary

protected	lock
protected	name
protected	properties
protected	segments
protected	title

Constructor Summary

public	MediaList()
--------	-----------------------------

Method Summary

void	addSegment (int index, MediaListSegment mediaListSegment)
void	addSegment (MediaListSegment mediaListSegment)
void	clearSegments ()
MediaListSegment	getFirstSegment ()
Object	getLock ()
String	getName ()
WMSProperties	getProperties ()
WMSProperties	getProperties (boolean write)
java.util.List	getSegment ()

MediaListSegment	getSegment (int index)
String	getTitle ()
MediaListSegment	removeSegment (int index)
void	removeSegment (MediaListSegment mediaListSegment)
void	reset ()
void	setName (String name)
void	setTitle (String title)
String	toSMILString ()
String	toString ()

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

name

protected java.lang.String **name**

title

protected java.lang.String **title**

segments

protected java.util.List **segments**

properties

protected com.wowza.wms.application.WMSProperties **properties**

(continued from last page)

lock

protected java.lang.Object **lock**

Constructors

MediaList

public **MediaList**()

Methods

toString

public String **toString**()

toSMILString

public String **toSMILString**()

reset

public void **reset**()

getSegment

public java.util.List **getSegment**()

addSegment

public void **addSegment**([MediaListSegment](#) mediaListSegment)

addSegment

public void **addSegment**(int index,
[MediaListSegment](#) mediaListSegment)

removeSegment

public void **removeSegment**([MediaListSegment](#) mediaListSegment)

removeSegment

```
public MediaListSegment removeSegment(int index)
```

clearSegments

```
public void clearSegments()
```

getFirstSegment

```
public MediaListSegment getFirstSegment()
```

getSegment

```
public MediaListSegment getSegment(int index)
```

getLock

```
public Object getLock()
```

getProperties

```
public WMSProperties getProperties()
```

getProperties

```
public WMSProperties getProperties(boolean write)
```

getName

```
public String getName()
```

setName

```
public void setName(String name)
```

getTitle

```
public String getTitle()
```

(continued from last page)

setTitle

```
public void setTitle(String title)
```

com.wowza.wms.medialist Class MediaListRendition

java.lang.Object

└─com.wowza.wms.medialist.MediaListRendition

public class **MediaListRendition**
extends Object

Field Summary

protected	audioCodecId
protected	bitrateAudio
protected	bitrateTotal
protected	bitrateVideo
protected	bitrateVideoPeak
protected	height
protected	lock
protected	mediaListSegment
protected	name
protected	playDuration
protected	playSeek
protected	playStart
protected	properties
protected	type
protected	videoCodecId
protected	vodTranscodeNGRP
protected	width

protected	wowzaAudioOnly
protected	wowzaKeyFrameOnly
protected	wowzaVideoOnly

Constructor Summary

public	MediaListRendition()
--------	--------------------------------------

Method Summary

String	getAudioCodecId()
int	getBitrateAudio()
int	getBitrateTotal()
int	getBitrateTotalPeak()
int	getBitrateVideo()
int	getBitrateVideoPeak()
String	getCodecId()
int	getHeight()
String	getLanguage()
Object	getLock()
MediaListSegment	getMediaListSegment()
String	getName()
long	getPlayDuration()
long	getPlaySeek()
long	getPlayStart()
WMSProperties	getProperties()
WMSProperties	getProperties(boolean write)
int[]	getSize()

String	getTitle()
int	getType()
String	getTypeAsString()
String	getVideoCodecId()
String	getVODTranscodeNGRP()
int	getWidth()
boolean	isWowzaAudioOnly()
boolean	isWowzaKeyFrameOnly()
boolean	isWowzaVideoOnly()
static java.util.ArrayList	parseCommaSeparatedLanguages (String langStr)
void	setAudioCodecId (String audioCodecId)
void	setBitrateAudio (int bitrateAudio)
void	setBitrateTotal (int bitrateTotal)
void	setBitrateVideo (int bitrateVideo)
void	setBitrateVideoPeak (int bitrateVideoPeak)
void	setHeight (int height)
void	setLanguage (String language)
void	setLock (Object lock)
void	setMediaListSegment (MediaListSegment mediaListSegment)
void	setName (String name)
void	setPlayDuration (long playDuration)
void	setPlaySeek (long playSeek)
void	setPlayStart (long playStart)
void	setSize (int width, int height)

void	<code>setTitle(String title)</code>
void	<code>setType(int type)</code>
void	<code>setVideoCodecId(String videoCodecId)</code>
void	<code>setVODTranscodeNGRP(String vodTranscodeNGRP)</code>
void	<code>setWidth(int width)</code>
void	<code>setWowzaAudioOnly(boolean wowzaAudioOnly)</code>
void	<code>setWowzaKeyFrameOnly(boolean wowzaKeyFrameOnly)</code>
void	<code>setWowzaVideoOnly(boolean wowzaVideoOnly)</code>
String	<code>toSMILString()</code>
String	<code>toString()</code>

Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Fields

lock

protected `java.lang.Object` **lock**

mediaListSegment

protected `com.wowza.wms.medialist.MediaListSegment` **mediaListSegment**

properties

protected `com.wowza.wms.application.WMSProperties` **properties**

name

protected `java.lang.String` **name**

type

protected int **type**

bitrateTotal

protected int **bitrateTotal**

bitrateAudio

protected int **bitrateAudio**

bitrateVideo

protected int **bitrateVideo**

bitrateVideoPeak

protected int **bitrateVideoPeak**

videoCodecId

protected java.lang.String **videoCodecId**

audioCodecId

protected java.lang.String **audioCodecId**

wowzaAudioOnly

protected boolean **wowzaAudioOnly**

wowzaVideoOnly

protected boolean **wowzaVideoOnly**

wowzaKeyFrameOnly

protected boolean **wowzaKeyFrameOnly**

(continued from last page)

width

protected int **width**

height

protected int **height**

playStart

protected long **playStart**

playDuration

protected long **playDuration**

playSeek

protected long **playSeek**

vodTranscodeNGRP

protected java.lang.String **vodTranscodeNGRP**

Constructors

MediaListRendition

public **MediaListRendition**()

Methods

toString

public String **toString**()

toSMILString

public String **toSMILString**()

(continued from last page)

getTypeAsString

```
public String getTypeAsString()
```

getLock

```
public Object getLock()
```

setLock

```
public void setLock(Object lock)
```

getMediaListSegment

```
public MediaListSegment getMediaListSegment()
```

setMediaListSegment

```
public void setMediaListSegment(MediaListSegment mediaListSegment)
```

getProperties

```
public WMSProperties getProperties()
```

getProperties

```
public WMSProperties getProperties(boolean write)
```

getBitrateTotal

```
public int getBitrateTotal()
```

getBitrateTotalPeak

```
public int getBitrateTotalPeak()
```

(continued from last page)

setBitrateTotal

```
public void setBitrateTotal(int bitrateTotal)
```

getBitrateAudio

```
public int getBitrateAudio()
```

setBitrateAudio

```
public void setBitrateAudio(int bitrateAudio)
```

getBitrateVideo

```
public int getBitrateVideo()
```

setBitrateVideo

```
public void setBitrateVideo(int bitrateVideo)
```

getBitrateVideoPeak

```
public int getBitrateVideoPeak()
```

setBitrateVideoPeak

```
public void setBitrateVideoPeak(int bitrateVideoPeak)
```

getVideoCodecId

```
public String getVideoCodecId()
```

setVideoCodecId

```
public void setVideoCodecId(String videoCodecId)
```

getCodecId

```
public String getCodecId()
```

(continued from last page)

getAudioCodecId

```
public String getAudioCodecId()
```

setAudioCodecId

```
public void setAudioCodecId(String audioCodecId)
```

isWowzaAudioOnly

```
public boolean isWowzaAudioOnly()
```

setWowzaAudioOnly

```
public void setWowzaAudioOnly(boolean wowzaAudioOnly)
```

getName

```
public String getName()
```

setName

```
public void setName(String name)
```

getType

```
public int getType()
```

setType

```
public void setType(int type)
```

getWidth

```
public int getWidth()
```

(continued from last page)

setWidth

```
public void setWidth(int width)
```

getHeight

```
public int getHeight()
```

setHeight

```
public void setHeight(int height)
```

setSize

```
public void setSize(int width,  
                    int height)
```

getSize

```
public int[] getSize()
```

getPlayStart

```
public long getPlayStart()
```

setPlayStart

```
public void setPlayStart(long playStart)
```

getPlayDuration

```
public long getPlayDuration()
```

setPlayDuration

```
public void setPlayDuration(long playDuration)
```

getPlaySeek

```
public long getPlaySeek()
```

(continued from last page)

setPlaySeek

```
public void setPlaySeek(long playSeek)
```

getVODTranscodeNGRP

```
public String getVODTranscodeNGRP()
```

setVODTranscodeNGRP

```
public void setVODTranscodeNGRP(String vodTranscodeNGRP)
```

setLanguage

```
public void setLanguage(String language)
```

getLanguage

```
public String getLanguage()
```

parseCommaSeparatedLanguages

```
public static java.util.ArrayList parseCommaSeparatedLanguages(String langStr)
```

isWowzaVideoOnly

```
public boolean isWowzaVideoOnly()
```

setWowzaVideoOnly

```
public void setWowzaVideoOnly(boolean wowzaVideoOnly)
```

isWowzaKeyFrameOnly

```
public boolean isWowzaKeyFrameOnly()
```

(continued from last page)

setWowzaKeyFrameOnly

```
public void setWowzaKeyFrameOnly(boolean wowzaKeyFrameOnly)
```

getTitle

```
public String getTitle()
```

setTitle

```
public void setTitle(String title)
```

com.wowza.wms.medialist

Class MediaListSegment

java.lang.Object

└─com.wowza.wms.medialist.MediaListSegment

public class **MediaListSegment**
extends Object

Field Summary

protected	lock
protected	mediaList
protected	properties
protected	renditions

Constructor Summary

public	MediaListSegment()
--------	------------------------------------

Method Summary

void	addRendition (int index, MediaListRendition mediaListRendition)
void	addRendition (MediaListRendition mediaListRendition)
void	clearSegments ()
MediaListRendition	getFirstRendition ()
Object	getLock ()
MediaList	getMediaList ()
WMSProperties	getProperties ()
WMSProperties	getProperties (boolean write)
MediaListRendition	getRendition (int index)
java.util.List	getRenditions ()

MediaListRendition	removeRendition (int index)
void	removeRendition (MediaListRendition mediaListRendition)
void	setLock (Object lock)
void	setMediaList (MediaList mediaList)
String	toSMILString ()
String	toString ()

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

lock

protected java.lang.Object **lock**

mediaList

protected com.wowza.wms.medialist.MediaList **mediaList**

renditions

protected java.util.List **renditions**

properties

protected com.wowza.wms.application.WMSProperties **properties**

Constructors

MediaListSegment

public **MediaListSegment**()

Methods

(continued from last page)

toString

```
public String toString()
```

toSMILString

```
public String toSMILString()
```

getRenditions

```
public java.util.List getRenditions()
```

addRendition

```
public void addRendition(MediaListRendition mediaListRendition)
```

addRendition

```
public void addRendition(int index,  
    MediaListRendition mediaListRendition)
```

removeRendition

```
public void removeRendition(MediaListRendition mediaListRendition)
```

removeRendition

```
public MediaListRendition removeRendition(int index)
```

clearSegments

```
public void clearSegments()
```

getFirstRendition

```
public MediaListRendition getFirstRendition()
```

(continued from last page)

getRendition

```
public MediaListRendition getRendition(int index)
```

getLock

```
public Object getLock()
```

setLock

```
public void setLock(Object lock)
```

getMediaList

```
public MediaList getMediaList()
```

setMediaList

```
public void setMediaList(MediaList mediaList)
```

getProperties

```
public WMSProperties getProperties()
```

getProperties

```
public WMSProperties getProperties(boolean write)
```

Package

com.wowza.wms.module

com.wowza.wms.module Interface IModuleCallResult

public interface **IModuleCallResult**
extends

IModuleCallResult: callback interface used by IClient call.

Method Summary

void	<code>onResult(IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)</code> Triggered on client side result from call to IClient.call
------	--

Methods

onResult

```
public void onResult(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Triggered on client side result from call to IClient.call

Parameters:

client - client
function - function object
params - result parameters

com.wowza.wms.module Interface IModuleNotify

public interface **IModuleNotify**
extends

IModuleNotify: listener interface for listening to module loading and unloading. See
IApplicationInstance.addModuleListener(IModuleNotify moduleListener)

Method Summary

void	<u>onModuleLoad</u> (ModuleItem item) Called when module loaded
void	<u>onModuleUnload</u> (ModuleItem item) Called when module unloaded

Methods

onModuleLoad

public void **onModuleLoad**(ModuleItem item)

Called when module loaded

Parameters:

item - module

onModuleUnload

public void **onModuleUnload**(ModuleItem item)

Called when module unloaded

Parameters:

item - module

com.wowza.wms.module Interface IModuleOnApp

All Subinterfaces:

[IModuleOnApp2](#)

All Known Implementing Classes:

[ModuleMediaCaster](#)

public interface **IModuleOnApp**
extends

IModuleCallResult: method interface examples for application level methods in a module.

Since module method must be implemented as static method a module cannot directly implements this interface. This interface only serves as an example of the method names and call signature needed to implement these application methods.

Method Summary

void	onAppStart (IApplicationInstance appInstance) Invoked when an application instance is started.
void	onAppStop (IApplicationInstance appInstance) Invoked when an application instance is stopped (destroyed).

Methods

onAppStart

public void **onAppStart**([IApplicationInstance](#) appInstance)

Invoked when an application instance is started.

Parameters:

appInstance - application instance

onAppStop

public void **onAppStop**([IApplicationInstance](#) appInstance)

Invoked when an application instance is stopped (destroyed).

Parameters:

appInstance - application instance

com.wowza.wms.module Interface IModuleOnApp2

All Superinterfaces:

[IModuleOnApp](#)

public interface **IModuleOnApp2**

extends [IModuleOnApp](#)

Method Summary

void	onAppCreate (IApplicationInstance appInstance) Invoked when an application instance is created.
void	onAppDestroy (IApplicationInstance appInstance) Invoked when an application instance is destroyed

Methods inherited from interface [com.wowza.wms.module.IModuleOnApp](#)

[onAppStart](#), [onAppStop](#)

Methods

onAppCreate

public void **onAppCreate**([IApplicationInstance](#) appInstance)

Invoked when an application instance is created.

Parameters:

appInstance - application instance

onAppDestroy

public void **onAppDestroy**([IApplicationInstance](#) appInstance)

Invoked when an application instance is destroyed

Parameters:

appInstance - application instance

com.wowza.wms.module Interface IModuleOnCall

public interface **IModuleOnCall**
extends

IModuleOnCall: method interface example for the catch-all method handler onCall.

Since module method must be implemented as static method a module cannot directly implements this interface. This interface only serves as an example of the method name and call signature needed to implement this method. The onCall method, when defined in a module, is invoked for all handlers that are undefined in a given module. The onCall handler can also be used to catch calls to server side component calls.

Method Summary

void

[onCall](#)(String handlerName, [IClient](#) client,
com.wowza.wms.request.RequestFunction function, [AMFDataList](#) params)
Catch-all method handler.

Methods

onCall

```
public void onCall(String handlerName,  
    IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Catch-all method handler. The onCall method, when defined in a module, is invoked for all handlers that are undefined in a given module.

Parameters:

handlerName - handler name
client - client
function - function object
params - function parameters

com.wowza.wms.module Interface IModuleOnConnect

public interface **IModuleOnConnect**
extends

IModuleOnConnect: method interface examples for client level methods in a module.

Since module method must be implemented as static method a module cannot directly implements this interface. This interface only serves as an example of the method names and call signature needed to implement these client methods.

Method Summary

void	onConnect (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) Invoked when a client connection is initiated.
void	onConnectAccept (IClient client) Invoked when a client connection is accepted.
void	onConnectReject (IClient client) Invoked when a client connection is rejected.
void	onDisconnect (IClient client) Invoked when a client disconnects.

Methods

onConnect

```
public void onConnect(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Invoked when a client connection is initiated. It is within this method that module can call to client.acceptConnection or client.rejectConnection.

Parameters:

client - client
function - function object
params - function parameters

onDisconnect

```
public void onDisconnect(IClient client)
```

Invoked when a client disconnects.

Parameters:

client - client

(continued from last page)

onConnectAccept

```
public void onConnectAccept(IClient client)
```

Invoked when a client connection is accepted.

Parameters:

client - client

onConnectReject

```
public void onConnectReject(IClient client)
```

Invoked when a client connection is rejected.

Parameters:

client - client

com.wowza.wms.module Interface IModuleOnHTTPCupertinoEncryption

public interface **IModuleOnHTTPCupertinoEncryption**
extends

IModuleOnHTTPCupertinoEncryption: listener interface for listening to AES-128 encryption events.

Field Summary

public static final	<u>KEYDATA_MODE_ENCRYPT</u> Value: 3
public static final	<u>KEYDATA_MODE_INIT</u> Value: 1
public static final	<u>KEYDATA_MODE_KEYFRAME_ENCRYPT</u> Value: 6
public static final	<u>KEYDATA_MODE_KEYFRAME_INIT</u> Value: 4
public static final	<u>KEYDATA_MODE_KEYFRAME_PLAYLIST</u> Value: 5
public static final	<u>KEYDATA_MODE_PLAYLIST</u> Value: 2

Method Summary

void	<u>onHTTPCupertinoEncryptionKeyCreateLive</u> (<u>IApplicationInstance</u> appInstance, <u>String</u> streamName, <u>byte[]</u> encKey) Called when live stream key is requested (per-published stream)
void	<u>onHTTPCupertinoEncryptionKeyCreateVOD</u> (<u>HTTPStreamerSessionCupertino</u> httpSession, <u>byte[]</u> encKey) Called when video on demand key is requested (per-session)
void	<u>onHTTPCupertinoEncryptionKeyData</u> (<u>HTTPStreamerSessionCupertino</u> httpSession, <u>IHTTPRequest</u> req, <u>IHTTPResponse</u> resp, <u>byte[]</u> encKeyData) Called when a key data is requested.
void	<u>onHTTPCupertinoEncryptionKeyLiveChunk</u> (<u>ILiveStreamPacketizer</u> liveStreamPacketizer, <u>String</u> streamName, <u>CupertinoEncInfo</u> encInfo, long chunkId, int mode) Called when live stream key is requested (per-published stream, per-chunk - for rotating keys)
void	<u>onHTTPCupertinoEncryptionKeyRequest</u> (<u>HTTPStreamerSessionCupertino</u> httpSession, <u>IHTTPRequest</u> req, <u>IHTTPResponse</u> resp) Called when a key is requested.

void	<div><div><div>onHTTPCupertinoEncryptionKeyVODChunk(HTTPStreamerSessionCupertino httpSession, IHTTPStreamerCupertinoIndex index, CupertinoEncInfo encInfo, long chunkId, int mode)</div></div><div>Called when video on demand key is requested (per-session).</div></div>
------	--

Fields

KEYDATA_MODE_INIT

public static final int **KEYDATA_MODE_INIT**

Constant value: **1**

KEYDATA_MODE_PLAYLIST

public static final int **KEYDATA_MODE_PLAYLIST**

Constant value: **2**

KEYDATA_MODE_ENCRYPT

public static final int **KEYDATA_MODE_ENCRYPT**

Constant value: **3**

KEYDATA_MODE_KEYFRAME_INIT

public static final int **KEYDATA_MODE_KEYFRAME_INIT**

Constant value: **4**

KEYDATA_MODE_KEYFRAME_PLAYLIST

public static final int **KEYDATA_MODE_KEYFRAME_PLAYLIST**

Constant value: **5**

KEYDATA_MODE_KEYFRAME_ENCRYPT

public static final int **KEYDATA_MODE_KEYFRAME_ENCRYPT**

Constant value: **6**

Methods

onHTTPCupertinoEncryptionKeyRequest

```
public void onHTTPCupertinoEncryptionKeyRequest(HTTPStreamerSessionCupertino
httpSession,
IHTTPRequest req,
IHTTPResponse resp)
```

(continued from last page)

Called when a key is requested. Call `httpSession.rejectSession` to reject the streaming session and stop delivery of the encryption key.

Parameters:

`httpSession` - HTTP session
`req` - HTTP request
`resp` - HTTP response

onHTTPCupertinoEncryptionKeyData

```
public void onHTTPCupertinoEncryptionKeyData(HTTPStreamerSessionCupertino httpSession,
IHTTPRequest req,
IHTTPResponse resp,
byte[] encKeyData)
```

Called when a key data is requested. Set `encKeyData` to the key data to be sent to the client.

Parameters:

`httpSession` - HTTP session
`req` - request
`resp` - response
`encKeyData` - key data

onHTTPCupertinoEncryptionKeyCreateVOD

```
public void onHTTPCupertinoEncryptionKeyCreateVOD(HTTPStreamerSessionCupertino
httpSession,
byte[] encKey)
```

Called when video on demand key is requested (per-session)

Parameters:

`httpSession` - HTTP session
`encKey` - encryption key

onHTTPCupertinoEncryptionKeyVODChunk

```
public void onHTTPCupertinoEncryptionKeyVODChunk(HTTPStreamerSessionCupertino
httpSession,
IHTTPStreamerCupertinoIndex index,
CupertinoEncInfo encInfo,
long chunkId,
int mode)
```

Called when video on demand key is requested (per-session). Allows setting of encryption key and URL.

Parameters:

`httpSession` - HTTP session
`index` - file index
`encInfo` - encryption key
`mode` - `KEYDATA_MODE_*`

onHTTPCupertinoEncryptionKeyCreateLive

```
public void onHTTPCupertinoEncryptionKeyCreateLive(IApplicationInstance appInstance,
String streamName,
byte[] encKey)
```

Called when live stream key is requested (per-published stream)

(continued from last page)

Parameters:

appInstance - application instance
streamName - stream name
encKey - encryption key

onHTTPCupertinoEncryptionKeyLiveChunk

```
public void onHTTPCupertinoEncryptionKeyLiveChunk(ILiveStreamPacketizer  
liveStreamPacketizer,  
    String streamName,  
    CupertinoEncInfo encInfo,  
    long chunkId,  
    int mode)
```

Called when live stream key is requested (per-published stream, per-chunk - for rotating keys)

Parameters:

liveStreamPacketizer - live stream packetizer
streamName - stream name
encInfo - encryption info
chunkId - chunk ID, -1 for stream creation
mode - KEYDATA_MODE_*

com.wowza.wms.module Interface IModuleOnHTTPCupertinoStreamingSession

public interface **IModuleOnHTTPCupertinoStreamingSession**
extends

IModuleOnHTTPSession: method interface for HTTP Cupertino Streaming session create/destroy.

Method Summary

void	<code>onHTTPCupertinoStreamingSessionCreate</code> (<code>HTTPStreamerSessionCupertino</code> <code>httpCupertinoStreamingSession</code>) Invoked when an HTTP Cupertino Streaming session is created.
void	<code>onHTTPCupertinoStreamingSessionDestroy</code> (<code>HTTPStreamerSessionCupertino</code> <code>httpCupertinoStreamingSession</code>) Invoked when an HTTP Cupertino Streaming session is destroyed.

Methods

onHTTPCupertinoStreamingSessionCreate

public void **onHTTPCupertinoStreamingSessionCreate**([`HTTPStreamerSessionCupertino`](#)
[`httpCupertinoStreamingSession`](#))

Invoked when an HTTP Cupertino Streaming session is created.

Parameters:

`httpCupertinoStreamingSession` - [`httpCupertinoStreamingSession`](#)

onHTTPCupertinoStreamingSessionDestroy

public void **onHTTPCupertinoStreamingSessionDestroy**([`HTTPStreamerSessionCupertino`](#)
[`httpCupertinoStreamingSession`](#))

Invoked when an HTTP Cupertino Streaming session is destroyed.

Parameters:

`httpCupertinoStreamingSession` - [`httpCupertinoStreamingSession`](#)

com.wowza.wms.module Interface IModuleOnHTTPMPEGDashEncryption

public interface **IModuleOnHTTPMPEGDashEncryption**
extends

IModuleOnHTTPMPEGDashEncryption: listener interface for listening to MPEG-DASH encryption events.

Method Summary

void	onHTTPMPEGDashEncryptionKeyLiveChunk (ILiveStreamPacketizer liveStreamPacketizer, String streamName, CencInfo cencInfo, long chunkId) Called when live stream key is requested (per-published stream)
void	onHTTPMPEGDashEncryptionKeyVODChunk (HTTPStreamerSessionMPEGDash httpSession, IHTTPStreamerMPEGDashIndex index, CencInfo cencInfo, long chunkId) Called when video on demand key is requested (per-session).

Methods

onHTTPMPEGDashEncryptionKeyVODChunk

```
public void onHTTPMPEGDashEncryptionKeyVODChunk(HTTPStreamerSessionMPEGDash
httpSession,
    IHTTPStreamerMPEGDashIndex index,
    CencInfo cencInfo,
    long chunkId)
```

Called when video on demand key is requested (per-session). Allows setting of encryption key and URL.

Parameters:

httpSession - HTTP session
index - file index
cencInfo - Common Encryption information
chunkId - chunk identifier

onHTTPMPEGDashEncryptionKeyLiveChunk

```
public void onHTTPMPEGDashEncryptionKeyLiveChunk(ILiveStreamPacketizer
liveStreamPacketizer,
    String streamName,
    CencInfo cencInfo,
    long chunkId)
```

Called when live stream key is requested (per-published stream)

Parameters:

liveStreamPacketizer - live stream packetizer
streamName - stream name
cencInfo - Common Encryption information
chunkId - chunk identifier

com.wowza.wms.module Interface IModuleOnHTTPMPEGDashStreamingSession

public interface **IModuleOnHTTPMPEGDashStreamingSession**
extends

IModuleOnHTTPMPEGDashStreamingSession: method interface for HTTP MPEGDash Streaming session create/destroy.

Method Summary

void	onHTTPMPEGDashStreamingSessionCreate (HTTPStreamerSessionMPEGDash httpMPEGDashStreamingSession) Invoked when an HTTP MPEGDash Streaming session is created.
void	onHTTPMPEGDashStreamingSessionDestroy (HTTPStreamerSessionMPEGDash httpMPEGDashStreamingSession) Invoked when an HTTP MPEGDash Streaming session is destroyed.

Methods

onHTTPMPEGDashStreamingSessionCreate

public void **onHTTPMPEGDashStreamingSessionCreate**([HTTPStreamerSessionMPEGDash](#)
[httpMPEGDashStreamingSession](#))

Invoked when an HTTP MPEGDash Streaming session is created.

Parameters:

[httpMPEGDashStreamingSession](#) - [httpMPEGDashStreamingSession](#)

onHTTPMPEGDashStreamingSessionDestroy

public void **onHTTPMPEGDashStreamingSessionDestroy**([HTTPStreamerSessionMPEGDash](#)
[httpMPEGDashStreamingSession](#))

Invoked when an HTTP MPEGDash Streaming session is destroyed.

Parameters:

[httpMPEGDashStreamingSession](#) - [httpMPEGDashStreamingSession](#)

com.wowza.wms.module

Interface IModuleOnHTTPSanJoseStreamingSession

public interface IModuleOnHTTPSanJoseStreamingSession
extends

IModuleOnHTTPSanJoseStreamingSession: method interface for HTTP SanJose Streaming session create/destroy.

Method Summary	
void	onHTTPSanJoseStreamingSessionCreate(HTTPStreamerSessionSanJose httpSanJoseStreamingSession) Invoked when an HTTP SanJose Streaming session is created.
void	onHTTPSanJoseStreamingSessionDestroy(HTTPStreamerSessionSanJose httpSanJoseStreamingSession) Invoked when an HTTP SanJose Streaming session is destroyed.

Methods

onHTTPSanJoseStreamingSessionCreate

public void **onHTTPSanJoseStreamingSessionCreate**([HTTPStreamerSessionSanJose httpSanJoseStreamingSession](#))

Invoked when an HTTP SanJose Streaming session is created.

Parameters:

`httpSanJoseStreamingSession` - httpSanJoseStreamingSession

onHTTPSanJoseStreamingSessionDestroy

public void **onHTTPSanJoseStreamingSessionDestroy**([HTTPStreamerSessionSanJose httpSanJoseStreamingSession](#))

Invoked when an HTTP SanJose Streaming session is destroyed.

Parameters:

`httpSanJoseStreamingSession` - httpSanJoseStreamingSession

com.wowza.wms.module Interface IModuleOnHTTPSession

public interface **IModuleOnHTTPSession**
extends

IModuleOnHTTPSession: method interface for HTTP Streaming session create/destroy.

Method Summary

void	onHTTPSessionCreate (IHTTPStreamerSession httpSession) Invoked when an HTTP Session is created (both Smooth and Cupertino sessions).
void	onHTTPSessionDestroy (IHTTPStreamerSession httpSession) Invoked when an HTTP Session is destroyed (both Smooth and Cupertino sessions).

Methods

onHTTPSessionCreate

public void **onHTTPSessionCreate**([IHTTPStreamerSession](#) httpSession)

Invoked when an HTTP Session is created (both Smooth and Cupertino sessions).

Parameters:

httpSession - httpSession

onHTTPSessionDestroy

public void **onHTTPSessionDestroy**([IHTTPStreamerSession](#) httpSession)

Invoked when an HTTP Session is destroyed (both Smooth and Cupertino sessions).

Parameters:

httpSession - httpSession

com.wowza.wms.module Interface IModuleOnHTTPSmoothStreamingPlayReady

public interface **IModuleOnHTTPSmoothStreamingPlayReady**
extends

IModuleOnHTTPSmoothStreamingPlayReady: Still working on this...

Method Summary

void	onHTTPSmoothStreamingPlayReadyCreateLive (IApplicationInstance appInstance, String streamName, com.wowza.wms.drm.playready.PlayReadyKeyInfo playReadyKeyInfo)
void	onHTTPSmoothStreamingPlayReadyCreateVOD (HTTPStreamerSessionSmoothStreamer httpSession, com.wowza.wms.drm.playready.PlayReadyKeyInfo playReadyKeyInfo)

Methods

onHTTPSmoothStreamingPlayReadyCreateVOD

```
public void onHTTPSmoothStreamingPlayReadyCreateVOD(HTTPStreamerSessionSmoothStreamer
httpSession,
    com.wowza.wms.drm.playready.PlayReadyKeyInfo playReadyKeyInfo)
```

onHTTPSmoothStreamingPlayReadyCreateLive

```
public void onHTTPSmoothStreamingPlayReadyCreateLive(IApplicationInstance appInstance,
    String streamName,
    com.wowza.wms.drm.playready.PlayReadyKeyInfo playReadyKeyInfo)
```

com.wowza.wms.module

Interface IModuleOnHTTPSmoothStreamingSession

public interface IModuleOnHTTPSmoothStreamingSession
extends

IModuleOnHTTPSession: method interface for HTTP Smooth Streaming session create/destroy.

Method Summary	
void	onHTTPSmoothStreamingSessionCreate(HTTPStreamerSessionSmoothStreamer httpSmoothStreamingSession) Invoked when an HTTP Smooth Streaming session is created.
void	onHTTPSmoothStreamingSessionDestroy(HTTPStreamerSessionSmoothStreamer httpSmoothStreamingSession) Invoked when an HTTP Smooth Streaming session is destroyed.

Methods

onHTTPSmoothStreamingSessionCreate

public void **onHTTPSmoothStreamingSessionCreate**([HTTPStreamerSessionSmoothStreamer httpSmoothStreamingSession](#))

Invoked when an HTTP Smooth Streaming session is created.

Parameters:

httpSmoothStreamingSession - httpSmoothStreamingSession

onHTTPSmoothStreamingSessionDestroy

public void **onHTTPSmoothStreamingSessionDestroy**([HTTPStreamerSessionSmoothStreamer httpSmoothStreamingSession](#))

Invoked when an HTTP Smooth Streaming session is destroyed.

Parameters:

httpSmoothStreamingSession - httpSmoothStreamingSession

com.wowza.wms.module Interface IModuleOnPushPublish

public interface **IModuleOnPushPublish**
extends

Method Summary

void	onPushPublishSessionCreate (String streamName, IMediaStream stream, IPushPublishSession pushPublishSession) Invoked when push publish session is created
void	onPushPublishSessionDestroy (String streamName, IMediaStream stream, IPushPublishSession pushPublishSession) Invoked when push publish session is destroyed

Methods

onPushPublishSessionCreate

```
public void onPushPublishSessionCreate(String streamName,  
    IMediaStream stream,  
    IPushPublishSession pushPublishSession)
```

Invoked when push publish session is created

Parameters:

streamName - stream name
stream - stream
pushPublishSession - push publish session

onPushPublishSessionDestroy

```
public void onPushPublishSessionDestroy(String streamName,  
    IMediaStream stream,  
    IPushPublishSession pushPublishSession)
```

Invoked when push publish session is destroyed

Parameters:

streamName - stream name
stream - stream
pushPublishSession - push publish session

com.wowza.wms.module Interface IModuleOnRTPSession

public interface **IModuleOnRTPSession**
extends

IModuleOnRTPSession: method interface for RTP session create/destroy.

Method Summary

void	onRTPSessionCreate(RTPSession rtpSession) Invoked when an RTP Session is created.
void	onRTPSessionDestroy(RTPSession rtpSession) Invoked when an RTP Session is destroyed.

Methods

onRTPSessionCreate

public void **onRTPSessionCreate**([RTPSession](#) rtpSession)

Invoked when an RTP Session is created.

Parameters:

rtpSession - rtpSession

onRTPSessionDestroy

public void **onRTPSessionDestroy**([RTPSession](#) rtpSession)

Invoked when an RTP Session is destroyed.

Parameters:

rtpSession - rtpSession

com.wowza.wms.module Interface IModuleOnStream

public interface **IModuleOnStream**
extends

IModuleOnStream: method interface examples for stream level methods in a module.

Method Summary

void	onStreamCreate (IMediaStream stream) Invoked when a stream is created.
void	onStreamDestroy (IMediaStream stream) Invoked when a stream is destroyed.

Methods

onStreamCreate

public void **onStreamCreate**([IMediaStream](#) stream)

Invoked when a stream is created.

Parameters:

stream - stream object

onStreamDestroy

public void **onStreamDestroy**([IMediaStream](#) stream)

Invoked when a stream is destroyed.

Parameters:

stream - stream object

com.wowza.wms.module Interface IModulePingResult

public interface **IModulePingResult**
extends

IModulePingResult: callback interface used by IClient ping.

Method Summary

void	<code>onResult(IClient client, long pingTime, int pingId, boolean result)</code> Triggered on client side result from call to IClient.ping.
------	--

Methods

onResult

```
public void onResult(IClient client,  
                    long pingTime,  
                    int pingId,  
                    boolean result)
```

Triggered on client side result from call to IClient.ping. This method will be called on a success or failure (ping timeout) of a ping.

Parameters:

`client` - client
`pingTime` - time in server time (milliseconds) when the ping was initiated
`pingId` - internal server id for the ping request
`result` - true is ping was successful false if failure (timeout)

com.wowza.wms.module Class ModuleBase

java.lang.Object

└-com.wowza.wms.module.ModuleBase

Direct Known Subclasses:

[ModuleProperties](#), [ModuleMediaCasterStreamMonitorAdvanced](#), [ModuleFastPlay](#), [ModuleCore](#), [ModuleClientLogging](#), [ModuleMediaCaster](#)

public abstract class **ModuleBase**
extends Object

ModuleBase: Base class that all server side modules should extend. Provides basic utility functionality for handling function parameters and return data. Also provides a simplified API for logging.

Field Summary

public static final	CALLBACK_PARAM1 Callback param: param1 Value: 1
public static final	CALLBACK_PARAM10 Callback param: param10 Value: 10
public static final	CALLBACK_PARAM2 Callback param: param2 Value: 2
public static final	CALLBACK_PARAM3 Callback param: param3 Value: 3
public static final	CALLBACK_PARAM4 Callback param: param4 Value: 4
public static final	CALLBACK_PARAM5 Callback param: param5 Value: 5
public static final	CALLBACK_PARAM6 Callback param: param6 Value: 6
public static final	CALLBACK_PARAM7 Callback param: param7 Value: 7

public static final	CALLBACK_PARAM8 Callback param: param8 Value: 8
public static final	CALLBACK_PARAM9 Callback param: param9 Value: 9
public static final	PARAM1 Method param: param1 Value: 3
public static final	PARAM10 Method param: param10 Value: 12
public static final	PARAM2 Method param: param2 Value: 4
public static final	PARAM3 Method param: param3 Value: 5
public static final	PARAM4 Method param: param4 Value: 6
public static final	PARAM5 Method param: param5 Value: 7
public static final	PARAM6 Method param: param6 Value: 8
public static final	PARAM7 Method param: param7 Value: 9
public static final	PARAM8 Method param: param8 Value: 10
public static final	PARAM9 Method param: param9 Value: 11
public static final	PARAMMETHODNAME Method: method name Value: 0
public static final	PLAYTRANSITION_APPEND Value: 0

public static final	<u>PLAYTRANSITION_APPEND_IMMEDIATE</u> Value: 2
public static final	<u>PLAYTRANSITION_RESET</u> Value: 1
public static final	<u>PLAYTRANSITION_RESET_IMMEDIATE</u> Value: 3
public static final	<u>PLAYTRANSITION_STOP</u> Value: 10
public static final	<u>PLAYTRANSITION_SWAP</u> Value: 12
public static final	<u>PLAYTRANSITION_SWITCH</u> Value: 13
public static final	<u>PLAYTRANSITION_UNKNOWN</u> Value: 14
public static final	<u>PLAYTRANSITIONSTR_APPEND</u> Play2 transition: APPEND Value: append
public static final	<u>PLAYTRANSITIONSTR_RESET</u> Play2 transition: RESET Value: reset
public static final	<u>PLAYTRANSITIONSTR_STOP</u> Play2 transition: STOP Value: stop
public static final	<u>PLAYTRANSITIONSTR_SWAP</u> Play2 transition: SWAP Value: swap
public static final	<u>PLAYTRANSITIONSTR_SWITCH</u> Play2 transition: SWITCH Value: switch
public static final	<u>PLAYTRANSITIONSTR_UNKNOWN</u> Play2 transition: UNKNOWN Value: unknown

Constructor Summary

public	<u>ModuleBase()</u>
--------	-------------------------------------

Method Summary

static IApplicationInstance	getAppInstance (IClient client) Get applicationInstace of a client.
static IApplication	getApplication (IClient client) Get application of a client.
static int	getCallbackParamCount (AMFDataList params) Get the total number of parameters passed to callback.
static WMSLogger	getLogger () Get the logging interface.
static AMFData	getParam (AMFDataList params, int index) Get parameter by index.
static boolean	getParamBoolean (AMFDataList params, int index) Get parameter by index, Return as boolean.
static boolean	getParamBoolean (AMFDataList params, int index, boolean defaultVal) Get parameter by index, Return as boolean.
static int	getParamCount (AMFDataList params) Get the total number of parameters passed to method.
static java.util.Date	getParamDate (AMFDataList params, int index) Get parameter by index, Return as Date.
static double	getParamDouble (AMFDataList params, int index) Get parameter by index, Return as double.
static double	getParamDouble (AMFDataList params, int index, double defaultVal) Get parameter by index, Return as double.
static int	getParamInt (AMFDataList params, int index) Get parameter by index, Return as int.
static int	getParamInt (AMFDataList params, int index, int defaultVal) Get parameter by index, Return as int.
static long	getParamLong (AMFDataList params, int index) Get parameter by index, Return as long.
static long	getParamLong (AMFDataList params, int index, long defaultVal) Get parameter by index, Return as long.
static AMFDataMixedArray	getParamMixedArray (AMFDataList params, int index) Get parameter by index, Return as AMFDataMixedArray.
static AMFDataObj	getParamObj (AMFDataList params, int index) Get parameter by index, Return as Object.
static String	getParamString (AMFDataList params, int index) Get parameter by index, Return as String.
static String	getParamString (AMFDataList params, int index, String defaultVal) Get parameter by index, Return as String.
static int	getParamType (AMFDataList params, int index) Get parameter type.

static IMediaStream	getStream (IClient client, com.wowza.wms.request.RequestFunction function) For methods called from the client side object NetStream (publish, play, deleteStream) get the IMediaStream object associated with the call.
static IVHost	getVHost (IClient client) Get vHost of a client.
void	invokePrevious (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) Wowza Pro will determine command priority based on the order of the modules in the module list.
static void	invokePrevious (Object instance, IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) Wowza Pro will determine command priority based on the order of the modules in the module list.
static boolean	isSendResult (AMFDataList params) Is this method call expecting sendResult to be called.
static void	sendClientOnStatusError (IClient client, String code, String description) Send an error message to the client-side client.onStatus handler
static boolean	sendResult (IClient client, AMFDataList params, AMFData data) Send a result to client method call as a AMFData object.
static boolean	sendResult (IClient client, AMFDataList params, boolean value) Send a result to client method call as a single boolean value.
static boolean	sendResult (IClient client, AMFDataList params, double value) Send a result to client method call as a single double value.
static boolean	sendResult (IClient client, AMFDataList params, int value) Send a result to client method call as a single int value.
static boolean	sendResult (IClient client, AMFDataList params, String value) Send a result to client method call as a single String value.
static void	sendStreamOnStatusError (IMediaStream stream, String code, String description) Send an error to the client-side NetStream.onStatus handler

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

PARAMMETHODNAME

public static final int **PARAMMETHODNAME**

Method: method name
Constant value: 0

(continued from last page)

PARAM1

```
public static final int PARAM1
```

Method param: param1
Constant value: **3**

PARAM2

```
public static final int PARAM2
```

Method param: param2
Constant value: **4**

PARAM3

```
public static final int PARAM3
```

Method param: param3
Constant value: **5**

PARAM4

```
public static final int PARAM4
```

Method param: param4
Constant value: **6**

PARAM5

```
public static final int PARAM5
```

Method param: param5
Constant value: **7**

PARAM6

```
public static final int PARAM6
```

Method param: param6
Constant value: **8**

PARAM7

```
public static final int PARAM7
```

Method param: param7
Constant value: **9**

PARAM8

```
public static final int PARAM8
```

Method param: param8
Constant value: **10**

PARAM9

```
public static final int PARAM9
```

(continued from last page)

Method param: param9
Constant value: **11**

PARAM10

public static final int **PARAM10**

Method param: param10
Constant value: **12**

CALLBACK_PARAM1

public static final int **CALLBACK_PARAM1**

Callback param: param1
Constant value: **1**

CALLBACK_PARAM2

public static final int **CALLBACK_PARAM2**

Callback param: param2
Constant value: **2**

CALLBACK_PARAM3

public static final int **CALLBACK_PARAM3**

Callback param: param3
Constant value: **3**

CALLBACK_PARAM4

public static final int **CALLBACK_PARAM4**

Callback param: param4
Constant value: **4**

CALLBACK_PARAM5

public static final int **CALLBACK_PARAM5**

Callback param: param5
Constant value: **5**

CALLBACK_PARAM6

public static final int **CALLBACK_PARAM6**

Callback param: param6
Constant value: **6**

CALLBACK_PARAM7

public static final int **CALLBACK_PARAM7**

Callback param: param7
Constant value: **7**

CALLBACK_PARAM8

```
public static final int CALLBACK_PARAM8
```

Callback param: param8
Constant value: **8**

CALLBACK_PARAM9

```
public static final int CALLBACK_PARAM9
```

Callback param: param9
Constant value: **9**

CALLBACK_PARAM10

```
public static final int CALLBACK_PARAM10
```

Callback param: param10
Constant value: **10**

PLAYTRANSITIONSTR_APPEND

```
public static final java.lang.String PLAYTRANSITIONSTR_APPEND
```

Play2 transition: APPEND
Constant value: **append**

PLAYTRANSITIONSTR_RESET

```
public static final java.lang.String PLAYTRANSITIONSTR_RESET
```

Play2 transition: RESET
Constant value: **reset**

PLAYTRANSITIONSTR_STOP

```
public static final java.lang.String PLAYTRANSITIONSTR_STOP
```

Play2 transition: STOP
Constant value: **stop**

PLAYTRANSITIONSTR_SWAP

```
public static final java.lang.String PLAYTRANSITIONSTR_SWAP
```

Play2 transition: SWAP
Constant value: **swap**

PLAYTRANSITIONSTR_SWITCH

```
public static final java.lang.String PLAYTRANSITIONSTR_SWITCH
```

Play2 transition: SWITCH
Constant value: **switch**

(continued from last page)

PLAYTRANSITIONSTR_UNKNOWN

```
public static final java.lang.String PLAYTRANSITIONSTR_UNKNOWN
```

Play2 transition: UNKNOWN
Constant value: **unknown**

PLAYTRANSITION_APPEND

```
public static final int PLAYTRANSITION_APPEND
```

Constant value: **0**

PLAYTRANSITION_RESET

```
public static final int PLAYTRANSITION_RESET
```

Constant value: **1**

PLAYTRANSITION_APPEND_IMMEDIATE

```
public static final int PLAYTRANSITION_APPEND_IMMEDIATE
```

Constant value: **2**

PLAYTRANSITION_RESET_IMMEDIATE

```
public static final int PLAYTRANSITION_RESET_IMMEDIATE
```

Constant value: **3**

PLAYTRANSITION_STOP

```
public static final int PLAYTRANSITION_STOP
```

Constant value: **10**

PLAYTRANSITION_SWAP

```
public static final int PLAYTRANSITION_SWAP
```

Constant value: **12**

PLAYTRANSITION_SWITCH

```
public static final int PLAYTRANSITION_SWITCH
```

Constant value: **13**

PLAYTRANSITION_UNKNOWN

```
public static final int PLAYTRANSITION_UNKNOWN
```

(continued from last page)

Constant value: **14**

Constructors

ModuleBase

```
public ModuleBase( )
```

Methods

getParamCount

```
protected static int getParamCount(AMFDataList params)
```

Get the total number of parameters passed to method.

Parameters:

params - parameters

Returns:

total number of parameters

getCallbackParamCount

```
protected static int getCallbackParamCount(AMFDataList params)
```

Get the total number of parameters passed to callback.

Parameters:

params - parameters

Returns:

total number of parameters

getParamType

```
protected static int getParamType(AMFDataList params,  
int index)
```

Get parameter type.

Parameters:

params - parameters

index - parameter index

Returns:

parameter type (AMFData.DATA_TYPE_*)

getParam

```
protected static AMFData getParam(AMFDataList params,  
int index)
```

Get parameter by index. Return as AMFData object.

(continued from last page)

Parameters:

params - parameters
index - parameter index

Returns:

parameter value as AMFData object, null if out of bounds

getParamMixedArray

```
protected static AMFDataMixedArray getParamMixedArray(AMFDataList params,  
int index)
```

Get parameter by index, Return as AMFDataMixedArray.

Parameters:

params - parameters
index - parameter index

Returns:

parameter value as AMFDataMixedArray object, null if out of bounds

getParamObj

```
protected static AMFDataObj getParamObj(AMFDataList params,  
int index)
```

Get parameter by index, Return as Object.

Parameters:

params - parameters
index - parameter index

Returns:

parameter value as Object object, null if out of bounds

getParamString

```
protected static String getParamString(AMFDataList params,  
int index)
```

Get parameter by index, Return as String.

Parameters:

params - parameters
index - parameter index

Returns:

parameter value as String object, null if out of bounds

getParamString

```
protected static String getParamString(AMFDataList params,  
int index,  
String defaultVal)
```

Get parameter by index, Return as String.

Parameters:

params - parameters

(continued from last page)

index - parameter index
defaultVal - default value

Returns:

parameter value as String object, defaultVal if out of bounds

getParamDate

```
protected static java.util.Date getParamDate(AMFDataList params,  
int index)
```

Get parameter by index, Return as Date.

Parameters:

params - parameters
index - parameter index

Returns:

parameter value as Date object, null if out of bounds

getParamInt

```
protected static int getParamInt(AMFDataList params,  
int index)
```

Get parameter by index, Return as int.

Parameters:

params - parameters
index - parameter index

Returns:

parameter value as int, 0 if out of bounds

getParamInt

```
protected static int getParamInt(AMFDataList params,  
int index,  
int defaultVal)
```

Get parameter by index, Return as int.

Parameters:

params - parameters
index - parameter index
defaultVal - default value

Returns:

parameter value as int, defaultVal if out of bounds

getParamDouble

```
protected static double getParamDouble(AMFDataList params,  
int index)
```

Get parameter by index, Return as double.

Parameters:

params - parameters

(continued from last page)

index - parameter index

Returns:

parameter value as double, 0 if out of bounds

getParamDouble

```
protected static double getParamDouble(AMFDataList params,  
    int index,  
    double defaultVal)
```

Get parameter by index, Return as double.

Parameters:

params - parameters
index - parameter index
defaultVal - default value

Returns:

parameter value as double, 0 if out of bounds

getParamLong

```
protected static long getParamLong(AMFDataList params,  
    int index)
```

Get parameter by index, Return as long.

Parameters:

params - parameters
index - parameter index

Returns:

parameter value as long, 0 if out of bounds

getParamLong

```
protected static long getParamLong(AMFDataList params,  
    int index,  
    long defaultVal)
```

Get parameter by index, Return as long.

Parameters:

params - parameters
index - parameter index
defaultVal - default value

Returns:

parameter value as long, defaultVal if out of bounds

getParamBoolean

```
protected static boolean getParamBoolean(AMFDataList params,  
    int index)
```

Get parameter by index, Return as boolean.

Parameters:

params - parameters

(continued from last page)

index - parameter index

Returns:

parameter value as boolean, false if out of bounds

getParamBoolean

```
protected static boolean getParamBoolean(AMFDataList params,  
int index,  
boolean defaultVal)
```

Get parameter by index, Return as boolean.

Parameters:

params - parameters
index - parameter index
defaultVal - default value

Returns:

parameter value as boolean, defaultVal if out of bounds

getAppInstance

```
protected static IApplicationInstance getAppInstance(IClient client)
```

Get applicationInstance of a client.

Parameters:

client - client

Returns:

applicationInstance

getApplication

```
protected static IApplication getApplication(IClient client)
```

Get application of a client.

Parameters:

client - client

Returns:

application

isSendResult

```
protected static boolean isSendResult(AMFDataList params)
```

Is this method call expecting sendResult to be called. If on the client side the call to `NetConnection.call("handlerName", resultObj, param1...);` had a value for resultObj (non-null), then the method is expecting some type of result or return data. Calling a variant of sendResult will provide this callback.

Parameters:

params - parameters

Returns:

true if client side call is expecting call to sendResult

getVHost

protected static [IVHost](#) **getVHost**([IClient](#) client)

Get vHost of a client.

Parameters:

client - client

Returns:

vHost

sendResult

protected static boolean **sendResult**([IClient](#) client,
[AMFDataList](#) params,
String value)

Send a result to client method call as a single String value.

Parameters:

client - client

params - parameters

value - return value

Returns:

true if client side call is expecting call to sendResult

sendResult

protected static boolean **sendResult**([IClient](#) client,
[AMFDataList](#) params,
boolean value)

Send a result to client method call as a single boolean value.

Parameters:

client - client

params - parameters

value - return value

Returns:

true if client side call is expecting call to sendResult

sendResult

protected static boolean **sendResult**([IClient](#) client,
[AMFDataList](#) params,
int value)

Send a result to client method call as a single int value.

Parameters:

client - client

params - parameters

value - return value

Returns:

true if client side call is expecting call to sendResult

sendResult

```
protected static boolean sendResult(IClient client,  
    AMFDataList params,  
    double value)
```

Send a result to client method call as a single double value.

Parameters:

client - client
params - parameters
value - return value

Returns:

true if client side call is expecting call to sendResult

sendResult

```
protected static boolean sendResult(IClient client,  
    AMFDataList params,  
    AMFData data)
```

Send a result to client method call as a AMFData object. This can be a single AMFData value like new AMFDataItem((double)1.234) or a complex type like AMFDataMixedArray, AMFDataArray or AMFDataObj.

Parameters:

client - client
params - parameters
data - return value

Returns:

true if client side call is expecting call to sendResult

getLogger

```
protected static WMSLogger getLogger()
```

Get the logging interface.

See Also:

[WMSLogger](#)

getStream

```
protected static IMediaStream getStream(IClient client,  
    com.wowza.wms.request.RequestFunction function)
```

For methods called from the client side object NetStream (publish, play, deleteStream) get the IMediaStream object associated with the call.

Parameters:

client - client
function - functions

Returns:

media stream

(continued from last page)

invokePrevious

```
protected static void invokePrevious(Object instance,
    IClient client,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

Wowza Pro will determine command priority based on the order of the modules in the module list. Wowza Pro by default will only invoke the last module that defines a given command. This method enables you to call the method by the same name in the previous module that defines that command. For example if you have three modules defined in the module list; ModuleCore, ModuleSecureURLParams and ModuleMyModule and all three modules implement the "play" command. When "play" is invoked it will invoke the "play" command in ModuleMyModule. If in ModuleMyModule you would like to call the "play" command in ModuleSecureURLParams, you can call: invokePrevious(this, client, function, params); and it will invoke the "play" command defined in ModuleSecureURLParams.

Parameters:

instance - instance of the current module
 client - client object passed into command
 function - function passed into command
 params - parameters passed into command

invokePrevious

```
protected void invokePrevious(IClient client,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

Wowza Pro will determine command priority based on the order of the modules in the module list. Wowza Pro by default will only invoke the last module that defines a given command. This method enables you to call the method by the same name in the previous module that defines that command. For example if you have three modules defined in the module list; ModuleCore, ModuleSecureURLParams and ModuleMyModule and all three modules implement the "play" command. When "play" is invoked it will invoke the "play" command in ModuleMyModule. If in ModuleMyModule you would like to call the "play" command in ModuleSecureURLParams, you can call: this.invokePrevious(client, function, params); and it will invoke the "play" command defined in ModuleSecureURLParams.

Parameters:

client - client object passed into command
 function - function passed into command
 params - parameters passed into command

sendClientOnStatusError

```
protected static void sendClientOnStatusError(IClient client,
    String code,
    String description)
```

Send an error message to the client-side client.onStatus handler

Parameters:

client - destination client
 code - code
 description - description

sendStreamOnStatusError

```
protected static void sendStreamOnStatusError(IMediaStream stream,
    String code,
    String description)
```

Send an error to the client-side NetStream.onStatus handler

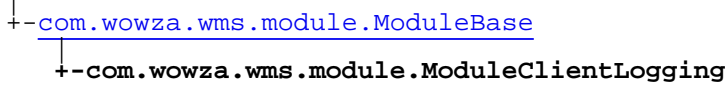
Parameters:

(continued from last page)

stream - destination stream
code - code
description - description

com.wowza.wms.module Class ModuleClientLogging

java.lang.Object



public class **ModuleClientLogging**
extends [ModuleBase](#)

Fields inherited from class [com.wowza.wms.module.ModuleBase](#)

[CALLBACK_PARAM1](#), [CALLBACK_PARAM10](#), [CALLBACK_PARAM2](#), [CALLBACK_PARAM3](#), [CALLBACK_PARAM4](#), [CALLBACK_PARAM5](#), [CALLBACK_PARAM6](#), [CALLBACK_PARAM7](#), [CALLBACK_PARAM8](#), [CALLBACK_PARAM9](#), [PARAM1](#), [PARAM10](#), [PARAM2](#), [PARAM3](#), [PARAM4](#), [PARAM5](#), [PARAM6](#), [PARAM7](#), [PARAM8](#), [PARAM9](#), [PARAMMETHODNAME](#), [PLAYTRANSITION_APPEND](#), [PLAYTRANSITION_APPEND_IMMEDIATE](#), [PLAYTRANSITION_RESET](#), [PLAYTRANSITION_RESET_IMMEDIATE](#), [PLAYTRANSITION_STOP](#), [PLAYTRANSITION_SWAP](#), [PLAYTRANSITION_SWITCH](#), [PLAYTRANSITION_UNKNOWN](#), [PLAYTRANSITIONSTR_APPEND](#), [PLAYTRANSITIONSTR_RESET](#), [PLAYTRANSITIONSTR_STOP](#), [PLAYTRANSITIONSTR_SWAP](#), [PLAYTRANSITIONSTR_SWITCH](#), [PLAYTRANSITIONSTR_UNKNOWN](#)

Constructor Summary

public	ModuleClientLogging()
--------	---------------------------------------

Method Summary

static void	logDebug (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) Send a debug message to the logging system NetConnection.call("logDebug", null, message);
static void	logError (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) Send a error message to the logging system NetConnection.call("logError", null, message);
static void	logInfo (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) Send a info message to the logging system NetConnection.call("logInfo", null, message);
static void	logWarn (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) Send a warning message to the logging system NetConnection.call("logWarn", null, message);

Methods inherited from class [com.wowza.wms.module.ModuleBase](#)

[getAppInstance](#), [getApplication](#), [getCallbackParamCount](#), [getLogger](#), [getParam](#), [getParamBoolean](#), [getParamBoolean](#), [getParamCount](#), [getParamDate](#), [getParamDouble](#), [getParamDouble](#), [getParamInt](#), [getParamInt](#), [getParamLong](#), [getParamLong](#), [getParamMixedArray](#), [getParamObj](#), [getParamString](#), [getParamString](#), [getParamType](#), [getStream](#), [getVHost](#), [invokePrevious](#), [invokePrevious](#), [isSendResult](#), [sendClientOnStatusError](#), [sendResult](#), [sendResult](#), [sendResult](#), [sendResult](#), [sendResult](#), [sendStreamOnStatusError](#)

Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Constructors

ModuleClientLogging

```
public ModuleClientLogging()
```

Methods

logDebug

```
public static void logDebug(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Send a debug message to the logging system `NetConnection.call("logDebug", null, message);`

Parameters:

`client` - client
`function` - function
`params` - params (message)

logInfo

```
public static void logInfo(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Send a info message to the logging system `NetConnection.call("logInfo", null, message);`

Parameters:

`client` - client
`function` - function
`params` - params (message)

logWarn

```
public static void logWarn(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Send a warning message to the logging system `NetConnection.call("logWarn", null, message);`

Parameters:

`client` - client
`function` - function
`params` - params (message)

(continued from last page)

logError

```
public static void logError(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Send a error message to the logging system `NetConnection.call("logError", null, message);`

Parameters:

client - client

function - function

params - params (message)

com.wowza.wms.module Class ModuleCore

```
java.lang.Object
|
+-com.wowza.wms.module.ModuleBase
|
+-com.wowza.wms.module.ModuleCore
```

```
public class ModuleCore
extends ModuleBase
```

ModuleCore includes all the basic commands support by the NetStream and NetConnection client side objects. Any of these methods can be overwritten in a custom module. Below is an example of custom module that overrides the play method to modify the name of the file to be played:

```
package com.wowza.wms.plugin.overrideexample;

import com.wowza.wms.amf.*;
import com.wowza.wms.client.*;
import com.wowza.wms.module.*;
import com.wowza.wms.request.*;

public class ModuleOverrideExample extends ModuleBase
{
    public void play(IClient client, RequestFunction function, AMFDataList params)
    {
        if (params.get(PARAM1).getType() == AMFData.DATA_TYPE_STRING)
        {
            String playName = params.getString(PARAM1);
            params.set(PARAM1, new AMFDataItem(playName+"_newname"));
        }
        this.invokePrevious(client, function, params);
    }
}
```

Fields inherited from class [com.wowza.wms.module.ModuleBase](#)

[CALLBACK_PARAM1](#), [CALLBACK_PARAM10](#), [CALLBACK_PARAM2](#), [CALLBACK_PARAM3](#), [CALLBACK_PARAM4](#), [CALLBACK_PARAM5](#), [CALLBACK_PARAM6](#), [CALLBACK_PARAM7](#), [CALLBACK_PARAM8](#), [CALLBACK_PARAM9](#), [PARAM1](#), [PARAM10](#), [PARAM2](#), [PARAM3](#), [PARAM4](#), [PARAM5](#), [PARAM6](#), [PARAM7](#), [PARAM8](#), [PARAM9](#), [PARAMMETHODNAME](#), [PLAYTRANSITION_APPEND](#), [PLAYTRANSITION_APPEND_IMMEDIATE](#), [PLAYTRANSITION_RESET](#), [PLAYTRANSITION_RESET_IMMEDIATE](#), [PLAYTRANSITION_STOP](#), [PLAYTRANSITION_SWAP](#), [PLAYTRANSITION_SWITCH](#), [PLAYTRANSITION_UNKNOWN](#), [PLAYTRANSITIONSTR_APPEND](#), [PLAYTRANSITIONSTR_RESET](#), [PLAYTRANSITIONSTR_STOP](#), [PLAYTRANSITIONSTR_SWAP](#), [PLAYTRANSITIONSTR_SWITCH](#), [PLAYTRANSITIONSTR_UNKNOWN](#)

Constructor Summary

public	ModuleCore()
--------	------------------------------

Method Summary

static void	closeStream (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) Server side implementation of NetStream.close();
static void	createStream (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) Create new server side NetStream object (internal to Flash workings).
static void	deleteStream (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) Delete server side stream object (internal to Flash workings).
static void	FCPublish (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) FCPublish method called by FME 2.5
static void	FCSubscribe (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) FCSubscribe subscribes to a live stream (if origin edge will start the stream from the origin to the edge)
static void	FCUnpublish (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) FCUnpublish method called by FME 2.5
static void	FCUnpublish (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) FCUnpublish method called by FME 2.5
static void	FCUnsubscribe (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) FCUnSubscribe to a live stream
static void	FCUnsubscribe (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) FCUnSubscribe to a live stream
static void	getClientID (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) Get the clientId for a client connection NetConnection.call("getClientID", resultObj);
static void	getLastStreamId (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) Get the id for the last created stream NetConnection.call("getLastStreamId", resultObj);
static void	getLiveStreamPacketizer (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) Get the live stream packetizer list for a client connection NetConnection.call("getLiveStreamPacketizer", resultObj);
static void	getPageUrl (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) getPageUrl returns the pageUrl from the onConnect metadata
static void	getReferrer (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) getReferrer returns the referrer from the onConnect metadata

static void	getRepeaterOriginUrl (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) Get the Repeater Origin URL for this client
void	getStreamBitrate (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) Get stream bitrate in bits-per-second for a video on demand stream (will not work for live).
static void	getStreamLength (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) Get stream length in seconds NetConnection.call("getStreamLength", resultObj, streamName); If you pass in an array of streamNames it will return an array of durations.
static void	getStreamType (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) Get the default stream type for a client connection NetConnection.call("getStreamType", resultObj);
static void	getVersion (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) Get the Wowza Pro server version and build number NetConnection.call("getVersion", resultObj);
static void	initLiveStreamRepeating (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) Initialize a stream for live stream repeating
static void	initStream (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) Create new server side NetStream object (internal to Flash workings).
static void	pause (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) Server side implementation of NetStream.pause([flag : Boolean]);
static void	pauseRaw (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) pauseRaw method introduced in Flash player 10
static void	play (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) Server side implementation of NetStream.play(name : Object [,start : Number[, len : Number[, reset : Object]]]);
static void	play2 (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) Server side implementation of NetStream.play(playOptions : NetStreamPlayOptions);
static void	publish (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) Server side implementation of NetStream.publish(name : String [, howToPublish : String]);
static void	receiveAudio (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) Server side implementation of NetStream.receiveAudio(receive : Boolean);
static void	receiveVideo (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) Server side implementation of NetStream.receiveVideo(receive : Boolean FPS : Number); FPS does not work the same as FMS.
static void	releaseStream (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)

static void	seek (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) Server side implementation of NetStream.seek(offset : Number);
static void	setBandwidthLimit (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)
static void	setBufferTime (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) Server side implementation of NetStream.setBufferTime(bufferTime : Number);
static void	setLiveStreamPacketizer (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) Set the live stream packetizer for a stream
static void	setRepeaterOriginUrl (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) Set the Repeater Origin URL for this client
static void	setStreamType (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) Set the default stream type for a client connection NetConnection.call("setStreamType", null, streamType);
static void	startTransmit (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)
static void	stopTransmit (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)

Methods inherited from class [com.wowza.wms.module.ModuleBase](#)

[getAppInstance](#), [getApplication](#), [getCallbackParamCount](#), [getLogger](#), [getParam](#), [getParamBoolean](#), [getParamBoolean](#), [getParamCount](#), [getParamDate](#), [getParamDouble](#), [getParamDouble](#), [getParamInt](#), [getParamInt](#), [getParamLong](#), [getParamLong](#), [getParamMixedArray](#), [getParamObj](#), [getParamString](#), [getParamString](#), [getParamType](#), [getStream](#), [getVHost](#), [invokePrevious](#), [invokePrevious](#), [isSendResult](#), [sendClientOnStatusError](#), [sendResult](#), [sendResult](#), [sendResult](#), [sendResult](#), [sendResult](#), [sendStreamOnStatusError](#)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

ModuleCore

```
public ModuleCore()
```

Methods

(continued from last page)

createStream

```
public static void createStream(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Create new server side NetStream object (internal to Flash workings).

Parameters:

client - client
function - function
params - parameters (no params)

initStream

```
public static void initStream(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Create new server side NetStream object (internal to Flash workings).

Parameters:

client - client
function - function
params - parameters (streamIndex, boolean)

releaseStream

```
public static void releaseStream(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

getLastStreamId

```
public static void getLastStreamId(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Get the id for the last created stream `NetConnection.call("getLastStreamId", resultObj);`

Parameters:

client - client
function - client
params - params (no params)

deleteStream

```
public static void deleteStream(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Delete server side stream object (internal to Flash workings).

Parameters:

client - client
function - function
params - params

publish

```
public static void publish(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Server side implementation of NetStream.publish(name : String [, howToPublish : String]);

Parameters:

client - client

function - function

params - params (name [if name = "false" or "null" stop publishing], howToPublish ["record", "live", "append"])

setBandwidthLimit

```
public static void setBandwidthLimit(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

play2

```
public static void play2(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Server side implementation of NetStream.play(playOptions : NetStreamPlayOptions);

Parameters:

client

function

params

play

```
public static void play(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Server side implementation of NetStream.play(name : Object [,start : Number[, len : Number[, reset : Object]]]);

Parameters:

client - client

function - function

params - params (name, start, len, reset)

closeStream

```
public static void closeStream(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Server side implementation of NetStream.close();

Parameters:

client - client

function - function

params - params (no params)

seek

```
public static void seek(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Server side implementation of NetStream.seek(offset : Number);

Parameters:

client - client
function - function
params - params (offset)

pause

```
public static void pause(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Server side implementation of NetStream.pause([flag : Boolean]);

Parameters:

client - client
function - function
params - params (flag)

setBufferTime

```
public static void setBufferTime(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Server side implementation of NetStream.setBufferTime(bufferTime : Number);

Parameters:

client - client
function - function
params - params (bufferTime)

getClientID

```
public static void getClientID(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Get the clientId for a client connection NetConnection.call("getClientID", resultObj);

Parameters:

client - client
function - function
params - params (no params)

getVersion

```
public static void getVersion(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Get the Wowza Pro server version and build number NetConnection.call("getVersion", resultObj);

(continued from last page)

Parameters:

client - client
function - function
params - params (no params)

setLiveStreamPacketizer

```
public static void setLiveStreamPacketizer(IClient client,  
com.wowza.wms.request.RequestFunction function,  
AMFDataList params)
```

Set the live stream packetizer for a stream

Parameters:

client - client
function - function
params - params (liveStreamPacketizer)

initLiveStreamRepeating

```
public static void initLiveStreamRepeating(IClient client,  
com.wowza.wms.request.RequestFunction function,  
AMFDataList params)
```

Initialize a stream for live stream repeating

Parameters:

client - client
function - function
params - params (liveStreamPacketizer, liveStreamRepeater)

getLiveStreamPacketizer

```
public static void getLiveStreamPacketizer(IClient client,  
com.wowza.wms.request.RequestFunction function,  
AMFDataList params)
```

Get the live stream packetizer list for a client connection `NetConnection.call("getLiveStreamPacketizer", resultObj);`

Parameters:

client - client
function - function
params - params (no params)

setStreamType

```
public static void setStreamType(IClient client,  
com.wowza.wms.request.RequestFunction function,  
AMFDataList params)
```

Set the default stream type for a client connection `NetConnection.call("setStreamType", null, streamType);`

Parameters:

client - client
function - function
params - params (streamType)

(continued from last page)

getStreamType

```
public static void getStreamType(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Get the default stream type for a client connection `NetConnection.call("getStreamType", resultObj);`

Parameters:

client - client
function - function
params - params (no params)

receiveAudio

```
public static void receiveAudio(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Server side implementation of `NetStream.receiveAudio(receive : Boolean);`

Parameters:

client - client
function - function
params - params (receive)

receiveVideo

```
public static void receiveVideo(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Server side implementation of `NetStream.receiveVideo(receive : Boolean | FPS : Number);` FPS does not work the same as FMS. Wowza Pro accepts the following values:

- true: Send all video frames
- false: Send no video
- -1: Send all video frames
- -2: Send half the frames (remove B frames) (Note: only works with Sorenson Spark video with B-frames)
- -3: Send only key frames

Parameters:

client - client
function - function
params - params (receive | FPS)

getStreamBitrate

```
public void getStreamBitrate(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Get stream bitrate in bits-per-second for a video on demand stream (will not work for live). To call, `NetConnection.call("getStreamBitrate", resultObj, streamName);`

Parameters:

client - client
function - function
params - params (streamName:String)

getStreamLength

```
public static void getStreamLength(IClient client,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

Get stream length in seconds NetConnection.call("getStreamLength", resultObj, streamName); If you pass in an array of streamNames it will return an array of durations.

Parameters:

client - client
function - function
params - params (streamName:String or streamNames:Array)

getRepeaterOriginUrl

```
public static void getRepeaterOriginUrl(IClient client,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

Get the Repeater Origin URL for this client

Parameters:

client - client
function - function
params - params

setRepeaterOriginUrl

```
public static void setRepeaterOriginUrl(IClient client,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

Set the Repeater Origin URL for this client

Parameters:

client - client
function - function
params - params (repeaterOriginUrl)

FCPublish

```
public static void FCPublish(IClient client,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

FCPublish method called by FME 2.5

Parameters:

client - client
function - function
params - params

FCUnPublish

```
public static void FCUnPublish(IClient client,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

(continued from last page)

FCUnpublish method called by FME 2.5

Parameters:

client - client
function - function
params - params

FCUnpublish

```
public static void FCUnpublish(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

pauseRaw

```
public static void pauseRaw(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

pauseRaw method introduced in Flash player 10

Parameters:

client - client
function - function
params - params

getReferrer

```
public static void getReferrer(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

getReferrer returns the referrer from the onConnect metadata

Parameters:

client - client
function - function
params - params

getPageUrl

```
public static void getPageUrl(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

getPageUrl returns the pageUrl from the onConnect metadata

Parameters:

client - client
function - function
params - params

FCSubscribe

```
public static void FCSubscribe(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

(continued from last page)

FCSubscribe subscribes to a live stream (if origin edge will start the stream from the origin to the edge)

Parameters:

client - client
function - function
params - params (streamName:String, [mediaCasterType:String])

FCUnsubscribe

```
public static void FCUnsubscribe(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

FCUnSubscribe to a live stream

Parameters:

client - client
function - function
params - params (streamName:String)

FCUnSubscribe

```
public static void FCUnSubscribe(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

startTransmit

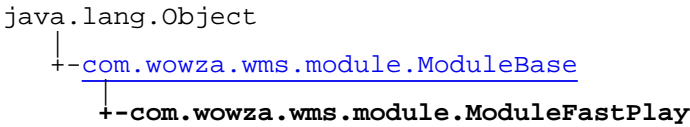
```
public static void startTransmit(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

stopTransmit

```
public static void stopTransmit(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

com.wowza.wms.module

Class ModuleFastPlay



```
public class ModuleFastPlay
extends ModuleBase
```

Fields inherited from class com.wowza.wms.module.ModuleBase
CALLBACK_PARAM1 , CALLBACK_PARAM10 , CALLBACK_PARAM2 , CALLBACK_PARAM3 , CALLBACK_PARAM4 , CALLBACK_PARAM5 , CALLBACK_PARAM6 , CALLBACK_PARAM7 , CALLBACK_PARAM8 , CALLBACK_PARAM9 , PARAM1 , PARAM10 , PARAM2 , PARAM3 , PARAM4 , PARAM5 , PARAM6 , PARAM7 , PARAM8 , PARAM9 , PARAMMETHODNAME , PLAYTRANSITION_APPEND , PLAYTRANSITION_APPEND_IMMEDIATE , PLAYTRANSITION_RESET , PLAYTRANSITION_RESET_IMMEDIATE , PLAYTRANSITION_STOP , PLAYTRANSITION_SWAP , PLAYTRANSITION_SWITCH , PLAYTRANSITION_UNKNOWN , PLAYTRANSITIONSTR_APPEND , PLAYTRANSITIONSTR_RESET , PLAYTRANSITIONSTR_STOP , PLAYTRANSITIONSTR_SWAP , PLAYTRANSITIONSTR_SWITCH , PLAYTRANSITIONSTR_UNKNOWN

Constructor Summary	
public	ModuleFastPlay()

Method Summary	
static void	setFastPlay (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) Turn on fast play NetStream.call("setFastPlay", null, multiplier, frames-per-second, direction);

Methods inherited from class com.wowza.wms.module.ModuleBase
getAppInstance , getApplication , getCallbackParamCount , getLogger , getParam , getParamBoolean , getParamBoolean , getParamCount , getParamDate , getParamDouble , getParamDouble , getParamInt , getParamInt , getParamLong , getParamLong , getParamMixedArray , getParamObj , getParamString , getParamString , getParamType , getStream , getVHost , invokePrevious , invokePrevious , isSendResult , sendClientOnStatusError , sendResult , sendResult , sendResult , sendResult , sendResult , sendStreamOnStatusError

Methods inherited from class java.lang.Object
clone , equals , finalize , getClass , hashCode , notify , notifyAll , toString , wait , wait , wait

Constructors

(continued from last page)

ModuleFastPlay

```
public ModuleFastPlay()
```

Methods

setFastPlay

```
public static void setFastPlay(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Turn on fast play NetStream.call("setFastPlay", null, multiplier, frames-per-second, direction);

Parameters:

client - client

function - function

params - params (multiplier, frames-per-second, direction)

com.wowza.wms.module Class ModuleMediaCasterStreamMonitorAdvanced

```
java.lang.Object
|
+-com.wowza.wms.module.ModuleBase
|
+-com.wowza.wms.module.ModuleMediaCasterStreamMonitorAdvanced
```

All Implemented Interfaces:

[IMediaCasterValidateMediaCaster](#)

```
public class ModuleMediaCasterStreamMonitorAdvanced
extends ModuleBase
implements IMediaCasterValidateMediaCaster
```

ModuleMediaCasterStreamMonitorAdvanced: Advanced stream monitoring module.

Add this <Module> definition to the end of the <Modules> list in [install-dir]/conf/[application]/Application.xml:

```
<Module>
  <Name>ModuleMediaCasterStreamMonitorAdvanced</Name>
  <Description>ModuleMediaCasterStreamMonitorAdvanced</Description>
  <Class>com.wowza.wms.module.ModuleMediaCasterStreamMonitorAdvanced</Class>
</Module>
```

Add these properties to the application level <Properties> container at the bottom of [install-dir]/conf/[application]/Application.xml (be sure to get the correct <Properties> container - there are several in the Application.xml file). Carefully read the instructions above each set of properties and set accordingly:


```

<!--
Monitor incoming packets (stream, audio, video) to be sure packets continue to flow from
encoder to stream. The
streamMonitor[type]StartTimeout controls the timeout (milliseconds) for the first packet. The
streamMonitor[type]Timeout
controls the timeout (milliseconds) for packets after the first packet. The stream type
refers to a catch all of any packet of
any type (audio, video, data). If any of these values are set to zero, the test is turned
off.
-->
<Property>
    <Name>streamMonitorStreamStartTimeout</Name>
    <Value>20000</Value>
    <Type>Integer</Type>
</Property>
<Property>
    <Name>streamMonitorStreamTimeout</Name>
    <Value>12000</Value>
    <Type>Integer</Type>
</Property>
<Property>
    <Name>streamMonitorVideoStartTimeout</Name>
    <Value>0</Value>
    <Type>Integer</Type>
</Property>
<Property>
    <Name>streamMonitorVideoTimeout</Name>
    <Value>0</Value>
    <Type>Integer</Type>
</Property>
<Property>
    <Name>streamMonitorAudioStartTimeout</Name>
    <Value>0</Value>
    <Type>Integer</Type>
</Property>
<Property>
    <Name>streamMonitorAudioTimeout</Name>
    <Value>0</Value>
    <Type>Integer</Type>
</Property>

<!--
Monitor the incoming packet timecodes (audio, video or data) to be sure packets do not arrive
out of order or late. The
streamMonitor[type]TCPosTolerance and streamMonitor[type]TCNegTolerance (milliseconds) values
define a sliding window
based on the timecode of the previous packet. For example if these values are set to -500 and
3000 respectively then the
timecode difference between the current arriving packet and the previous packet of that type
must fall within
-500 and 3000 milliseconds. If not and streamMonitor[type]TCToleranceEnable is set to true

```

then stream will be considered
unhealthy and will be reset.

-->

```
<Property>
    <Name>streamMonitorVideoTCToleranceEnable</Name>
    <Value>false</Value>
    <Type>Boolean</Type>
</Property>
<Property>
    <Name>streamMonitorVideoTCPosTolerance</Name>
    <Value>3000</Value>
    <Type>Integer</Type>
</Property>
<Property>
    <Name>streamMonitorVideoTCNegTolerance</Name>
    <Value>-500</Value>
    <Type>Integer</Type>
</Property>
<Property>
    <Name>streamMonitorAudioTCToleranceEnable</Name>
    <Value>false</Value>
    <Type>Boolean</Type>
</Property>
<Property>
    <Name>streamMonitorAudioTCPosTolerance</Name>
    <Value>3000</Value>
    <Type>Integer</Type>
</Property>
<Property>
    <Name>streamMonitorAudioTCNegTolerance</Name>
    <Value>-500</Value>
    <Type>Integer</Type>
</Property>
<Property>
    <Name>streamMonitorDataTCToleranceEnable</Name>
    <Value>false</Value>
    <Type>Boolean</Type>
</Property>
<Property>
    <Name>streamMonitorDataTCPosTolerance</Name>
    <Value>3000</Value>
    <Type>Integer</Type>
</Property>
<Property>
    <Name>streamMonitorDataTCNegTolerance</Name>
    <Value>-500</Value>
    <Type>Integer</Type>
</Property>

<!--
```

Monitors the time difference between the audio and video channel of a stream. If the difference between the currently arriving video packet and the previous audio packet (or vice-versa) is greater than streamMonitorAVSyncTolerance and streamMonitorAVSyncToleranceEnable is set to true, then the stream will be considered unhealthy and will be reset.

-->

<Property>

<Name>streamMonitorAVSyncToleranceEnable</Name>

<Value>>false</Value>

<Type>Boolean</Type>

</Property>

<Property>

<Name>streamMonitorAVSyncTolerance</Name>

<Value>1500</Value>

<Type>Integer</Type>

</Property>

<!--

If set to true, then when a stream is reset and it belong to a MediaStreamNameGroup all streams in the group will be reset. If false only the unhealthy stream will be reset.

-->

<Property>

<Name>streamMonitorResetNameGroups</Name>

<Value>>true</Value>

<Type>Boolean</Type>

</Property>

<!--

Turns on debug logging of the monitoring.

-->

<Property>

<Name>streamMonitorDebug</Name>

<Value>>false</Value>

<Type>Boolean</Type>

</Property>

Field Summary

protected	appInstance
protected	badStreams
protected	monitors
protected	streamMonitorAudioStartTimeout

protected	streamMonitorAudioTCNegTolerance
protected	streamMonitorAudioTCPosTolerance
protected	streamMonitorAudioTCToleranceEnable
protected	streamMonitorAudioTimeout
protected	streamMonitorAVSyncTolerance
protected	streamMonitorAVSyncToleranceEnable
protected	streamMonitorDataTCNegTolerance
protected	streamMonitorDataTCPosTolerance
protected	streamMonitorDataTCToleranceEnable
protected	streamMonitorDebug
protected	streamMonitorResetNameGroups
protected	streamMonitorStreamStartTimeout
protected	streamMonitorStreamTimeout
protected	streamMonitorVideoStartTimeout
protected	streamMonitorVideoTCNegTolerance
protected	streamMonitorVideoTCPosTolerance
protected	streamMonitorVideoTCToleranceEnable
protected	streamMonitorVideoTimeout

Fields inherited from class [com.wowza.wms.module.ModuleBase](#)

[CALLBACK_PARAM1](#), [CALLBACK_PARAM10](#), [CALLBACK_PARAM2](#), [CALLBACK_PARAM3](#), [CALLBACK_PARAM4](#), [CALLBACK_PARAM5](#), [CALLBACK_PARAM6](#), [CALLBACK_PARAM7](#), [CALLBACK_PARAM8](#), [CALLBACK_PARAM9](#), [PARAM1](#), [PARAM10](#), [PARAM2](#), [PARAM3](#), [PARAM4](#), [PARAM5](#), [PARAM6](#), [PARAM7](#), [PARAM8](#), [PARAM9](#), [PARAMMETHODNAME](#), [PLAYTRANSITION_APPEND](#), [PLAYTRANSITION_APPEND_IMMEDIATE](#), [PLAYTRANSITION_RESET](#), [PLAYTRANSITION_RESET_IMMEDIATE](#), [PLAYTRANSITION_STOP](#), [PLAYTRANSITION_SWAP](#), [PLAYTRANSITION_SWITCH](#), [PLAYTRANSITION_UNKNOWN](#), [PLAYTRANSITIONSTR_APPEND](#), [PLAYTRANSITIONSTR_RESET](#), [PLAYTRANSITIONSTR_STOP](#), [PLAYTRANSITIONSTR_SWAP](#), [PLAYTRANSITIONSTR_SWITCH](#), [PLAYTRANSITIONSTR_UNKNOWN](#)

Constructor Summary

public	ModuleMediaCasterStreamMonitorAdvanced()
--------	--

Method Summary

void	onAppStart (IApplicationInstance appInstance)
void	onAppStop (IApplicationInstance appInstance)
boolean	onResetMediaCaster (IApplicationInstance appInstance, IMediaCaster mediaCaster)
void	onStreamCreate (IMediaStream stream)
void	onStreamDestroy (IMediaStream stream)
boolean	onValidateMediaCaster (IApplicationInstance appInstance, IMediaCaster mediaCaster)
void	onValidateMediaCastersStart (IApplicationInstance appInstance)
void	onValidateMediaCastersStop (IApplicationInstance appInstance)

Methods inherited from class [com.wowza.wms.module.ModuleBase](#)

[getAppInstance](#), [getApplication](#), [getCallbackParamCount](#), [getLogger](#), [getParam](#), [getParamBoolean](#), [getParamBoolean](#), [getParamCount](#), [getParamDate](#), [getParamDouble](#), [getParamDouble](#), [getParamInt](#), [getParamInt](#), [getParamLong](#), [getParamLong](#), [getParamMixedArray](#), [getParamObj](#), [getParamString](#), [getParamString](#), [getParamType](#), [getStream](#), [getVHost](#), [invokePrevious](#), [invokePrevious](#), [isSendResult](#), [sendClientOnStatusError](#), [sendResult](#), [sendResult](#), [sendResult](#), [sendResult](#), [sendResult](#), [sendStreamOnStatusError](#)

Methods inherited from class [java.lang.Object](#)

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

Methods inherited from interface [com.wowza.wms.mediacaster.IMediaCasterValidateMediaCaster](#)

[onResetMediaCaster](#), [onValidateMediaCaster](#), [onValidateMediaCastersStart](#), [onValidateMediaCastersStop](#)

Fields

streamMonitorStreamStartTimeout

protected int **streamMonitorStreamStartTimeout**

(continued from last page)

streamMonitorStreamTimeout

protected int **streamMonitorStreamTimeout**

streamMonitorVideoTCToleranceEnable

protected boolean **streamMonitorVideoTCToleranceEnable**

streamMonitorVideoTCPosTolerance

protected int **streamMonitorVideoTCPosTolerance**

streamMonitorVideoTCNegTolerance

protected int **streamMonitorVideoTCNegTolerance**

streamMonitorAudioTCToleranceEnable

protected boolean **streamMonitorAudioTCToleranceEnable**

streamMonitorAudioTCPosTolerance

protected int **streamMonitorAudioTCPosTolerance**

streamMonitorAudioTCNegTolerance

protected int **streamMonitorAudioTCNegTolerance**

streamMonitorDataTCToleranceEnable

protected boolean **streamMonitorDataTCToleranceEnable**

streamMonitorDataTCPosTolerance

protected int **streamMonitorDataTCPosTolerance**

streamMonitorDataTCNegTolerance

protected int **streamMonitorDataTCNegTolerance**

(continued from last page)

streamMonitorAVSyncToleranceEnableprotected boolean **streamMonitorAVSyncToleranceEnable**

streamMonitorAVSyncToleranceprotected int **streamMonitorAVSyncTolerance**

streamMonitorVideoStartTimeoutprotected int **streamMonitorVideoStartTimeout**

streamMonitorVideoTimeoutprotected int **streamMonitorVideoTimeout**

streamMonitorAudioStartTimeoutprotected int **streamMonitorAudioStartTimeout**

streamMonitorAudioTimeoutprotected int **streamMonitorAudioTimeout**

streamMonitorResetNameGroupsprotected boolean **streamMonitorResetNameGroups**

streamMonitorDebugprotected boolean **streamMonitorDebug**

appInstanceprotected com.wowza.wms.application.IApplicationInstance **appInstance**

(continued from last page)

monitors

```
protected java.util.Map monitors
```

badStreams

```
protected java.util.Set badStreams
```

Constructors

ModuleMediaCasterStreamMonitorAdvanced

```
public ModuleMediaCasterStreamMonitorAdvanced()
```

Methods

onAppStart

```
public void onAppStart(IApplicationInstance appInstance)
```

onAppStop

```
public void onAppStop(IApplicationInstance appInstance)
```

onStreamCreate

```
public void onStreamCreate(IMediaStream stream)
```

onStreamDestroy

```
public void onStreamDestroy(IMediaStream stream)
```

onValidateMediaCastersStart

```
public void onValidateMediaCastersStart(IApplicationInstance appInstance)
```

onValidateMediaCaster

```
public boolean onValidateMediaCaster(IApplicationInstance appInstance,  
    IMediaCaster mediaCaster)
```

onValidateMediaCastersStop

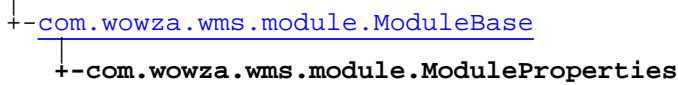
```
public void onValidateMediaCastersStop(IApplicationInstance appInstance)
```

onResetMediaCaster

```
public boolean onResetMediaCaster(IApplicationInstance appInstance,  
    IMediaCaster mediaCaster)
```

com.wowza.wms.module Class ModuleProperties

java.lang.Object



public class **ModuleProperties**
extends [ModuleBase](#)

Fields inherited from class [com.wowza.wms.module.ModuleBase](#)

[CALLBACK_PARAM1](#), [CALLBACK_PARAM10](#), [CALLBACK_PARAM2](#), [CALLBACK_PARAM3](#), [CALLBACK_PARAM4](#), [CALLBACK_PARAM5](#), [CALLBACK_PARAM6](#), [CALLBACK_PARAM7](#), [CALLBACK_PARAM8](#), [CALLBACK_PARAM9](#), [PARAM1](#), [PARAM10](#), [PARAM2](#), [PARAM3](#), [PARAM4](#), [PARAM5](#), [PARAM6](#), [PARAM7](#), [PARAM8](#), [PARAM9](#), [PARAMMETHODNAME](#), [PLAYTRANSITION_APPEND](#), [PLAYTRANSITION_APPEND_IMMEDIATE](#), [PLAYTRANSITION_RESET](#), [PLAYTRANSITION_RESET_IMMEDIATE](#), [PLAYTRANSITION_STOP](#), [PLAYTRANSITION_SWAP](#), [PLAYTRANSITION_SWITCH](#), [PLAYTRANSITION_UNKNOWN](#), [PLAYTRANSITIONSTR_APPEND](#), [PLAYTRANSITIONSTR_RESET](#), [PLAYTRANSITIONSTR_STOP](#), [PLAYTRANSITIONSTR_SWAP](#), [PLAYTRANSITIONSTR_SWITCH](#), [PLAYTRANSITIONSTR_UNKNOWN](#)

Constructor Summary

public	ModuleProperties ()
--------	--------------------------------------

Method Summary

static void	getAppInstanceProperty (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) Get application instance level property value NetConnection.call("getAppInstanceProperty", resultObj, name);
static void	getApplicationProperty (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) Get application level property value NetConnection.call("getApplicationProperty", resultObj, name);
static void	getClientProperty (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) Get client level property value NetConnection.call("getClientProperty", resultObj, name);
static void	getStreamProperty (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) Get stream level property value NetConnection.call("getStreamProperty", resultObj, streamId, name);
static void	setAppInstanceProperty (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) Set application instance level property NetConnection.call("setAppInstanceProperty", null, name, value);
static void	setApplicationProperty (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) Set application level property NetConnection.call("setApplicationProperty", null, name, value);

static void	setClientProperty (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) Set client level property NetConnection.call("setClientProperty", null, name, value);
static void	setStreamProperty (IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params) Set stream level property NetConnection.call("setStreamProperty", null, streamId, name, value);

Methods inherited from class [com.wowza.wms.module.ModuleBase](#)

[getAppInstance](#), [getApplication](#), [getCallbackParamCount](#), [getLogger](#), [getParam](#),
[getParamBoolean](#), [getParamBoolean](#), [getParamCount](#), [getParamDate](#), [getParamDouble](#),
[getParamDouble](#), [getParamInt](#), [getParamInt](#), [getParamLong](#), [getParamLong](#),
[getParamMixedArray](#), [getParamObj](#), [getParamString](#), [getParamString](#), [getParamType](#),
[getStream](#), [getVHost](#), [invokePrevious](#), [invokePrevious](#), [isSendResult](#),
[sendClientOnStatusError](#), [sendResult](#), [sendResult](#), [sendResult](#), [sendResult](#), [sendResult](#),
[sendStreamOnStatusError](#)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait,
wait

Constructors

ModuleProperties

```
public ModuleProperties()
```

Methods

setApplicationProperty

```
public static void setApplicationProperty(IClient client,  
com.wowza.wms.request.RequestFunction function,  
AMFDataList params)
```

Set application level property NetConnection.call("setApplicationProperty", null, name, value);

Parameters:

client - client
function - function
params - params (name, value)

getApplicationProperty

```
public static void getApplicationProperty(IClient client,  
com.wowza.wms.request.RequestFunction function,  
AMFDataList params)
```

Get application level property value NetConnection.call("getApplicationProperty", resultObj, name);

Parameters:

client - client
function - function

(continued from last page)

params - params (name)

setAppInstanceProperty

```
public static void setAppInstanceProperty(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Set application instance level property NetConnection.call("setAppInstanceProperty", null, name, value);

Parameters:

client - client
function - function
params - params (name, value)

getAppInstanceProperty

```
public static void getAppInstanceProperty(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Get application instance level property value NetConnection.call("getAppInstanceProperty", resultObj, name);

Parameters:

client - client
function - function
params - params (name)

setClientProperty

```
public static void setClientProperty(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Set client level property NetConnection.call("setClientProperty", null, name, value);

Parameters:

client - client
function - function
params - params (name, value)

getClientProperty

```
public static void getClientProperty(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Get client level property value NetConnection.call("getClientProperty", resultObj, name);

Parameters:

client - client
function - function
params - params (name)

setStreamProperty

```
public static void setStreamProperty(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

(continued from last page)

Set stream level property `NetConnection.call("setStreamProperty", null, streamId, name, value);`

Parameters:

`client` - client

`function` - function

`params` - params (streamId, name, value)

getStreamProperty

```
public static void getStreamProperty(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Get stream level property value `NetConnection.call("getStreamProperty", resultObj, streamId, name);`

Parameters:

`client` - client

`function` - function

`params` - params (streamId, name)

Package

com.wowza.wms.pushpublish.model

com.wowza.wms.pushpublish.model Interface IPushPublish

All Known Implementing Classes:
[PushPublishBase](#)

public interface **IPushPublish**
 extends

The Interface IPushPublisher.

Field Summary

public static final	PUSHPUBLISH_TYPE_RTMP Value: 1
public static final	PUSHPUBLISH_TYPE_RTP Value: 2
public static final	PUSHPUBLISH_TYPE_UNKNOWN Value: 0

Method Summary

void	connect()
IPushPublishSession	createPushPublishSession()
void	disconnect()
String	getContextStr()
String	getDstStreamName()
String	getHostname()
int	getMapFileIndex()
long	getMessagesAttemptedCount()
long	getMessagesFailedCount()
long	getMessagesRetriedCount()
int	getPort()
WMSProperties	getProperties()

int	<u>getPushPublishType()</u>
int	<u>getReconnectWaitTime()</u>
<u>IMediaStream</u>	<u>getSrcStream()</u>
String	<u>getSrcStreamName()</u>
void	<u>init()</u> (<u>IApplicationInstance</u> appInstance2, String streamName, <u>IMediaStream</u> stream, java.util.HashMap profileData, java.util.HashMap maps, boolean debug)
Boolean	<u>isConnected()</u>
void	<u>resetConnection()</u>
void	<u>setDebugLog()</u> (boolean debugLog)
void	<u>setDstStreamName()</u> (String dstStreamNameIn)
void	<u>setHost()</u> (String hostname)
void	<u>setHostname()</u> (String hostname)
void	<u>setMapFileIndex()</u> (String index)
void	<u>setPort()</u> (int port)
void	<u>setPort()</u> (String port)
void	<u>setReconnectWaitTime()</u> (int reconnectWaitTime)
void	<u>setSrcStream()</u> (<u>IMediaStream</u> srcStream)
void	<u>setSrcStreamName()</u> (String srcStreamName)
void	<u>setStreamName()</u> (String streamName)

Fields

PUSHPUBLISH_TYPE_UNKNOWN

```
public static final int PUSHPUBLISH_TYPE_UNKNOWN
```

Constant value: **0**

(continued from last page)

PUSHPUBLISH_TYPE_RTMP

```
public static final int PUSHPUBLISH_TYPE_RTMP
```

Constant value: **1**

PUSHPUBLISH_TYPE_RTP

```
public static final int PUSHPUBLISH_TYPE_RTP
```

Constant value: **2**

Methods

connect

```
public void connect()
```

disconnect

```
public void disconnect()
```

resetConnection

```
public void resetConnection()
```

getProperties

```
public WMSProperties getProperties()
```

getMapFileIndex

```
public int getMapFileIndex()
```

setMapFileIndex

```
public void setMapFileIndex(String index)
```

getSrcStreamName

```
public String getSrcStreamName()
```

setSrcStreamName

```
public void setSrcStreamName(String srcStreamName)
```

getDstStreamName

```
public String getDstStreamName()
```

setDstStreamName

```
public void setDstStreamName(String dstStreamNameIn)
```

setStreamName

```
public void setStreamName(String streamName)
```

getHostname

```
public String getHostname()
```

setHost

```
public void setHost(String hostname)
```

setHostname

```
public void setHostname(String hostname)
```

setDebugLog

```
public void setDebugLog(boolean debugLog)
```

getPort

```
public int getPort()
```

setPort

```
public void setPort(int port)
```

(continued from last page)

setPort

```
public void setPort(String port)
```

getReconnectWaitTime

```
public int getReconnectWaitTime()
```

setReconnectWaitTime

```
public void setReconnectWaitTime(int reconnectWaitTime)
```

init

```
public void init(IApplicationInstance appInstance2,  
                String streamName,  
                IMediaStream stream,  
                java.util.HashMap profileData,  
                java.util.HashMap maps,  
                boolean debug)
```

getPushPublishType

```
public int getPushPublishType()
```

createPushPublishSession

```
public IPushPublishSession createPushPublishSession()
```

getSrcStream

```
public IMediaStream getSrcStream()
```

setSrcStream

```
public void setSrcStream(IMediaStream srcStream)
```

getContextStr

```
public String getContextStr()
```

(continued from last page)

isConnected

```
public Boolean isConnected()
```

getMessagesAttemptedCount

```
public long getMessagesAttemptedCount()
```

getMessagesRetriedCount

```
public long getMessagesRetriedCount()
```

getMessagesFailedCount

```
public long getMessagesFailedCount()
```

com.wowza.wms.pushpublish.model Interface IPushPublishCrossSession

All Known Implementing Classes:

[PushPublishHTTPPlaylistMap](#)

public interface **IPushPublishCrossSession**
extends

Method Summary

boolean	isSessionValid()
---------	----------------------------------

Methods

isSessionValid

public boolean **isSessionValid()**

com.wowza.wms.pushpublish.model Interface IPushPublishHTTPAdaptiveGroup

public interface **IPushPublishHTTPAdaptiveGroup**
extends

Method Summary

void	<u>addAdaptiveGroupItem</u> (String adaptiveGroupName)
String	<u>createGroupPlaylist</u> (String id, java.util.List groupingsByRedundantPairs, java.util.List groupingsByAbrGroups)
java.util.List	<u>getAdaptiveGroupItems</u> ()
String	<u>getDstStreamName</u> ()
String	<u>getGroupId</u> ()
String	<u>getGroupMemberName</u> ()
String	<u>getPlaylistRenditionDescription</u> ()
String	<u>getRedundancyPairId</u> ()
String	<u>getSrcStreamName</u> ()
Boolean	<u>groupPlaylistNeedsUpdating</u> (long lastUpdateTime, java.util.List groupingsByRedundantPairs, java.util.List groupingsByAbrGroups)
boolean	<u>isDebugLog</u> ()
boolean	<u>isInAdaptiveGroup</u> ()
boolean	<u>isInRedundancyPair</u> ()
boolean	<u>isSendToBackupServer</u> ()
String	<u>sendAdaptiveGroupPlaylist</u> (String groupName, String playlistStr)

Methods

(continued from last page)

groupPlaylistNeedsUpdating

```
public Boolean groupPlaylistNeedsUpdating(long lastUpdateTime,  
    java.util.List groupingsByRedundantPairs,  
    java.util.List groupingsByAbrGroups)
```

createGroupPlaylist

```
public String createGroupPlaylist(String id,  
    java.util.List groupingsByRedundantPairs,  
    java.util.List groupingsByAbrGroups)
```

getPlaylistRenditionDescription

```
public String getPlaylistRenditionDescription()
```

sendAdaptiveGroupPlaylist

```
public String sendAdaptiveGroupPlaylist(String groupName,  
    String playlistStr)
```

isDebugLog

```
public boolean isDebugLog()
```

addAdaptiveGroupItem

```
public void addAdaptiveGroupItem(String adaptiveGroupName)
```

getAdaptiveGroupItems

```
public java.util.List getAdaptiveGroupItems()
```

isInAdaptiveGroup

```
public boolean isInAdaptiveGroup()
```

isInRedundancyPair

```
public boolean isInRedundancyPair()
```

isSendToBackupServer

```
public boolean isSendToBackupServer()
```

getGroupId

```
public String getGroupId()
```

getGroupMemberName

```
public String getGroupMemberName()
```

getRedundancyPairId

```
public String getRedundancyPairId()
```

getSrcStreamName

```
public String getSrcStreamName()
```

getDstStreamName

```
public String getDstStreamName()
```


com.wowza.wms.pushpublish.model Interface IPushPublishHTTPOutputBuilder

public interface **IPushPublishHTTPOutputBuilder**
extends

Method Summary

PushPublishHTTPOutputItem	createOutputItem (ILiveStreamPacketizer liveStreamPacketizer, int lastChunkIndex)
IApplicationInstance	getAppInstance ()
int	getChunkDurationTarget (ILiveStreamPacketizer liveStreamPacketizer)
int	getConnectionTimeout ()
String	getDstStreamName ()
String	getIngressDomain ()
String	getIngressPath ()
int	getIngressPort ()
int	getLastChunkId (ILiveStreamPacketizer liveStreamPacketizer)
String	getLiveStreamPacketizerName ()
int	getMaxRetries ()
Object	getPacketizerLock (ILiveStreamPacketizer liveStreamPacketizer)
int	getReadWriteTimeout ()
String	getSrcStreamName ()
boolean	isDebugLog ()
boolean	isHttpWriterDebug ()
void	resetNewPacketizer (ILiveStreamPacketizer liveStreamPacketizer)
void	sendOutput (PushPublishHTTPOutputItem outputItem, HTTPByteWriter byteWriter, int maxItemsInDir)

Methods

getAppInstance

```
public IApplicationInstance getAppInstance()
```

isDebugLog

```
public boolean isDebugLog()
```

isHttpWriterDebug

```
public boolean isHttpWriterDebug()
```

getSrcStreamName

```
public String getSrcStreamName()
```

getDstStreamName

```
public String getDstStreamName()
```

getLiveStreamPacketizerName

```
public String getLiveStreamPacketizerName()
```

getPacketizerLock

```
public Object getPacketizerLock(ILiveStreamPacketizer liveStreamPacketizer)
```

getLastChunkId

```
public int getLastChunkId(ILiveStreamPacketizer liveStreamPacketizer)
```

getChunkDurationTarget

```
public int getChunkDurationTarget(ILiveStreamPacketizer liveStreamPacketizer)
```

getIngressDomain

```
public String getIngressDomain()
```

getIngressPort

```
public int getIngressPort()
```

getIngressPath

```
public String getIngressPath()
```

getConnectionTimeout

```
public int getConnectionTimeout()
```

getReadWriteTimeout

```
public int getReadWriteTimeout()
```

getMaxRetries

```
public int getMaxRetries()
```

resetNewPacketizer

```
public void resetNewPacketizer(ILiveStreamPacketizer liveStreamPacketizer)
```

createOutputItem

```
public PushPublishHTTPOutputItem createOutputItem(ILiveStreamPacketizer  
liveStreamPacketizer,  
int lastChunkIndex)
```

sendOutput

```
public void sendOutput(PushPublishHTTPOutputItem outputItem,  
HTTPByteWriter byteWriter,  
int maxItemsInDir)
```

com.wowza.wms.pushpublish.model Interface IPushPublishSession

All Known Implementing Classes:
[PushPublishSessionBase](#)

public interface **IPushPublishSession**
extends

The Interface IPushPublishSession.

Field Summary

public static final	PUSHPUBLISHSESSION_TYPE_RTMP Value: 1
public static final	PUSHPUBLISHSESSION_TYPE_RTP Value: 2
public static final	PUSHPUBLISHSESSION_TYPE_UNKNOWN Value: 0

Method Summary

String	getContextStr()
String	getProfile()
IPushPublish	getPushPublishImpl()
IMediaStream	getStream()
int	getType()
void	setProfile (String profile)
void	setPushPublishImpl (IPushPublish pushPublishImpl)
void	setStream (IMediaStream stream)

Fields

PUSHPUBLISHSESSION_TYPE_UNKNOWN

public static final int **PUSHPUBLISHSESSION_TYPE_UNKNOWN**

(continued from last page)

Constant value: **0**

PUSHPUBLISHSESSION_TYPE_RTMP

```
public static final int PUSHPUBLISHSESSION_TYPE_RTMP
```

Constant value: **1**

PUSHPUBLISHSESSION_TYPE_RTP

```
public static final int PUSHPUBLISHSESSION_TYPE_RTP
```

Constant value: **2**

Methods

getType

```
public int getType()
```

getPushPublishImpl

```
public IPushPublish getPushPublishImpl()
```

setPushPublishImpl

```
public void setPushPublishImpl(IPushPublish pushPublishImpl)
```

getStream

```
public IMediaStream getStream()
```

setStream

```
public void setStream(IMediaStream stream)
```

getProfile

```
public String getProfile()
```

setProfile

```
public void setProfile(String profile)
```

(continued from last page)

getContextStr

```
public String getContextStr()
```

com.wowza.wms.pushpublish.model Interface IPushPublishSessionNotify

All Known Implementing Classes:

[PushPublishSessionNotifyBase](#)

public interface **IPushPublishSessionNotify**
extends

Method Summary

void	onPushPublishSessionCreate (String streamName, IMediaStream stream, IPushPublishSession pushPublishSession) Invoked when push publish session is created
void	onPushPublishSessionDestroy (String streamName, IMediaStream stream, IPushPublishSession pushPublishSession) Invoked when push publish session is destroyed

Methods

onPushPublishSessionCreate

```
public void onPushPublishSessionCreate(String streamName,  
    IMediaStream stream,  
    IPushPublishSession pushPublishSession)
```

Invoked when push publish session is created

Parameters:

streamName - stream name
stream - stream
pushPublishSession - push publish session

onPushPublishSessionDestroy

```
public void onPushPublishSessionDestroy(String streamName,  
    IMediaStream stream,  
    IPushPublishSession pushPublishSession)
```

Invoked when push publish session is destroyed

Parameters:

streamName - stream name
stream - stream
pushPublishSession - push publish session

com.wowza.wms.pushpublish.model Class PushPublishBase

java.lang.Object

└-com.wowza.wms.pushpublish.model.PushPublishBase

All Implemented Interfaces:

[IPushPublish](#)

Direct Known Subclasses:

[PushPublishRTP](#), [PushPublishRTMP](#), [PushPublishHTTPBase](#)

public abstract class **PushPublishBase**

extends Object

implements [IPushPublish](#)

The Class PushPublisherBase.

Field Summary

protected	appInstance The app instance.
public static final	CLASS
public static final	CLASSNAME Value: PushPublishBase
protected	dstStreamName The dst stream name.
protected	hostname The m host.
protected	inetAddr The m inet address.
protected	mapFileIndex
protected	mImplementation The m implementation.
protected	mType The m type.
protected	originalTimecodeThreshold
protected	password The password.
protected	port The m port.

protected	properties The properties.
protected	pushPublishStreamDebug
protected	pushPublishType
protected	reconnectWaitTime
protected	sendingPerfomance
protected	sendOriginalTimecodes The my params.
protected	srcStream
protected	srcStreamName The src stream name.
protected	userName The user name.

Fields inherited from interface [com.wowza.wms.pushpublish.model.IPushPublish](#)

[PUSHPUBLISH_TYPE_RTMP](#), [PUSHPUBLISH_TYPE_RTP](#), [PUSHPUBLISH_TYPE_UNKNOWN](#)

Constructor Summary

public	PushPublishBase () Instantiates a new push publisher base.
--------	--

Method Summary

IPushPublishSession	createPushPublishSession ()
IApplicationInstance	getAppInstance () Get appInstance interface.
double	getBandwidth ()
boolean	getConfigDataOrDefault (org.w3c.dom.Element config, String tag, boolean deflt) Gets the config data or default.
int	getConfigDataOrDefault (org.w3c.dom.Element config, String tag, int deflt) Gets the config data or default.
String	getConfigDataOrDefault (org.w3c.dom.Element config, String tag, String deflt) Gets the config data or default.
String	getContextStr ()

String	<u>getDstStreamName()</u> Get destination stream name.
String	<u>getHostname()</u> Get destination server host name.
String	<u>getImplementation()</u> Gets the implementation.
int	<u>getMapFileIndex()</u>
static int	<u>getNumberFromString()</u> (String in)
long	<u>getOriginalTimecodeThreshold()</u> Get the original timecode threshold.
String	<u>getPassword()</u> Get Akamai password.
int	<u>getPort()</u> Get destination server port.
<u>WMSProperties</u>	<u>getProperties()</u>
int	<u>getPushPublishType()</u>
static String	<u>getPushPublishVersionStr()</u> Gets the push publish version str.
int	<u>getReconnectWaitTime()</u> Get the reconnect wait time (milliseconds).
<u>IMediaStream</u>	<u>getSrcStream()</u>
String	<u>getSrcStreamName()</u> Get source stream name.
String	<u>getUserName()</u> Gets the user name.
void	<u>init()</u> (<u>IApplicationInstance</u> appInstance2, String streamName, <u>IMediaStream</u> stream, java.util.HashMap profileData, java.util.HashMap maps, boolean streamDebug) Inits the instance.
boolean	<u>isAdaptiveStreaming()</u> Checks if is adaptive streaming.
boolean	<u>isSendOriginalTimecodes()</u> true if sending the original timecodes of source stream.
void	<u>load()</u> (java.util.HashMap dataMap)
void	<u>setAdaptiveStreaming()</u> (boolean isAdaptiveStreaming) Sets the adaptive streaming.

void	<u>setAppInstance</u> (<u>IApplicationInstance</u> appInstance) Set appInstance interface.
void	<u>setDebugLog</u> (boolean debugLog) Set debug logging.
void	<u>setDstStreamName</u> (String dstStreamNameIn) Set destination stream name.
void	<u>setHost</u> (String hostname) Set destination server host name.
void	<u>setHostname</u> (String hostname)
void	<u>setImplementation</u> (String implementation) Sets the implementation.
void	<u>setMapFileIndex</u> (String index)
void	<u>setOriginalTimecodeThreshold</u> (long originalTimecodeThreshold) Set the original timecode threshold.
void	<u>setOriginalTimecodeThreshold</u> (String i)
void	<u>setPassword</u> (String password) Set Akamai password.
void	<u>setPort</u> (int port) Set destination server port.
void	<u>setPort</u> (String port)
void	<u>setReconnectWaitTime</u> (int reconnectWaitTime) Set the reconnect wait time (milliseconds).
void	<u>setSendOriginalTimecodes</u> (boolean sendOriginalTimecodes) true if sending the original timecodes of source stream.
void	<u>setSrcStream</u> (<u>IMediaStream</u> srcStream)
void	<u>setSrcStreamName</u> (String srcStreamName) Set source stream name.
void	<u>setStreamName</u> (String streamName) Sets the stream name.
void	<u>setUserName</u> (String userName) Set Akamai user name.

Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Methods inherited from interface [com.wowza.wms.pushpublish.model.IPushPublish](#)

```
connect, createPushPublishSession, disconnect, getContextStr, getDstStreamName,  
getHostname, getMapFileIndex, getMessagesAttemptedCount, getMessagesFailedCount,  
getMessagesRetriedCount, getPort, getProperties, getPushPublishType,  
getReconnectWaitTime, getSrcStream, getSrcStreamName, init, isConnected,  
resetConnection, setDebugLog, setDstStreamName, setHost, setHostname,  
setMapFileIndex, setPort, setPort, setReconnectWaitTime, setSrcStream,  
setSrcStreamName, setStreamName
```

Fields

CLASS

```
public static final java.lang.Class CLASS
```

CLASSNAME

```
public static final java.lang.String CLASSNAME
```

Constant value: **PushPublishBase**

pushPublishType

```
protected int pushPublishType
```

sendingPerfomance

```
protected com.wowza.util.IOPerformanceCounter sendingPerfomance
```

pushPublishStreamDebug

```
protected boolean pushPublishStreamDebug
```

reconnectWaitTime

```
protected int reconnectWaitTime
```

mapFileIndex

```
protected int mapFileIndex
```

(continued from last page)

properties

protected com.wowza.wms.application.WMSProperties **properties**

The properties.

userName

protected java.lang.String **userName**

The user name.

password

protected java.lang.String **password**

The password.

appInstance

protected com.wowza.wms.application.IApplicationInstance **appInstance**

The app instance.

srcStreamName

protected java.lang.String **srcStreamName**

The src stream name.

srcStream

protected com.wowza.wms.stream.IMediaStream **srcStream**

dstStreamName

protected java.lang.String **dstStreamName**

The dst stream name.

hostname

protected java.lang.String **hostname**

The m host.

inetAddr

protected java.net.InetAddress **inetAddr**

The m inet address.

port

protected int **port**

(continued from last page)

The m port.

mImplementation

protected java.lang.String **mImplementation**

The m implementation.

mType

protected java.lang.String **mType**

The m type.

sendOriginalTimecodes

protected boolean **sendOriginalTimecodes**

The my params.

originalTimecodeThreshold

protected long **originalTimecodeThreshold**

Constructors

PushPublishBase

public **PushPublishBase**()

Instantiates a new push publisher base.

Methods

init

```
public void init(IApplicationInstance appInstance2,  
                String streamName,  
                IMediaStream stream,  
                java.util.HashMap profileData,  
                java.util.HashMap maps,  
                boolean streamDebug)
```

Init's the instance.

Parameters:

appInstance2 - the app instance2
streamName
maps
profileData

getImplementation

public String **getImplementation**()

Gets the implementation.

(continued from last page)

Returns:
the implementation

setImplementation

```
public void setImplementation(String implementation)
```

Sets the implementation.

Parameters:
implementation - the new implementation

getProperties

```
public WMSProperties getProperties()
```

getConfigDataOrDefault

```
protected String getConfigDataOrDefault(org.w3c.dom.Element config,  
    String tag,  
    String deflt)
```

Gets the config data or default.

Parameters:
config - the config
tag - the tag
deflt - the deflt

Returns:
the config data or default

getConfigDataOrDefault

```
protected int getConfigDataOrDefault(org.w3c.dom.Element config,  
    String tag,  
    int deflt)
```

Gets the config data or default.

Parameters:
config - the config
tag - the tag
deflt - the deflt

Returns:
the config data or default

getConfigDataOrDefault

```
protected boolean getConfigDataOrDefault(org.w3c.dom.Element config,  
    String tag,  
    boolean deflt)
```

Gets the config data or default.

Parameters:

(continued from last page)

config - the config
tag - the tag
deflt - the deflt

Returns:

the config data or default

isAdaptiveStreaming

```
public boolean isAdaptiveStreaming()
```

Checks if is adaptive streaming.

Returns:

true, if is adaptive streaming

setAdaptiveStreaming

```
public void setAdaptiveStreaming(boolean isAdaptiveStreaming)
```

Sets the adaptive streaming.

Parameters:

isAdaptiveStreaming - the new adaptive streaming

isSendOriginalTimecodes

```
public boolean isSendOriginalTimecodes()
```

true if sending the original timecodes of source stream. false if starting at zero.

Returns:

true if sending the original timecodes of source stream

setSendOriginalTimecodes

```
public void setSendOriginalTimecodes(boolean sendOriginalTimecodes)
```

true if sending the original timecodes of source stream. false if starting at zero.

Parameters:

sendOriginalTimecodes - true if sending the original timecodes of source stream

getOriginalTimecodeThreshold

```
public long getOriginalTimecodeThreshold()
```

Get the original timecode threshold. This value will attempt to reduce the magnitude of absolute timecodes while keeping multiple-bitrate alignment. It is probably best to set this value quite high to a value such as 0x100000. The value is in milliseconds. The following formula is used to calculate a timecode offset $\text{timecodeOffset} = \text{timecode} - (\text{timecode} \% \text{originalTimecodeThreshold})$;

Returns:

original timecode threshold

setOriginalTimecodeThreshold

```
public void setOriginalTimecodeThreshold(long originalTimecodeThreshold)
```


(continued from last page)

Set the original timecode threshold. This value will attempt to reduce the magnitude of absolute timecodes while keeping multiple-bitrate alignment. It is probably best to set this value quite high to a value such as 0x100000. The value is in milliseconds. The following formula is used to calculate a timecode offset $\text{timecodeOffset} = \text{timecode} - (\text{timecode} \% \text{originalTimecodeThreshold})$;

Parameters:

`originalTimecodeThreshold` - original timecode threshold

setOriginalTimecodeThreshold

```
public void setOriginalTimecodeThreshold(String i)
```

getUserName

```
public String getUserName()
```

Gets the user name.

Returns:

the user name

setUserName

```
public void setUserName(String userName)
```

Set Akamai user name.

Parameters:

`userName` - Akamai user name

getPassword

```
public String getPassword()
```

Get Akamai password.

Returns:

Akamai password

setPassword

```
public void setPassword(String password)
```

Set Akamai password.

Parameters:

`password` - Akamai password

getAppInstance

```
public IApplicationInstance getAppInstance()
```

Get appInstance interface.

Returns:

appInstance interface

setAppInstance

```
public void setAppInstance(IApplicationInstance appInstance)
```

Set appInstance interface.

Parameters:

appInstance - appInstance interface

getSrcStreamName

```
public String getSrcStreamName()
```

Get source stream name.

Returns:

source stream name

setSrcStreamName

```
public void setSrcStreamName(String srcStreamName)
```

Set source stream name.

Parameters:

srcStreamName - source stream name

getDstStreamName

```
public String getDstStreamName()
```

Get destination stream name.

Returns:

destination stream name

setDstStreamName

```
public void setDstStreamName(String dstStreamNameIn)
```

Set destination stream name.

Parameters:

dstStreamNameIn - destination stream name

setStreamName

```
public void setStreamName(String streamName)
```

Sets the stream name.

Parameters:

streamName - the new stream name

getHostname

```
public String getHostname()
```

(continued from last page)

Get destination server host name.

Returns:

destination server host name

setHost

```
public void setHost(String hostname)
```

Set destination server host name.

Parameters:

hostname - destination server host name

setHostname

```
public void setHostname(String hostname)
```

setDebugLog

```
public void setDebugLog(boolean debugLog)
```

Set debug logging.

Parameters:

debugLog - debug logging

getPort

```
public int getPort()
```

Get destination server port.

Returns:

destination server port

setPort

```
public void setPort(int port)
```

Set destination server port.

Parameters:

port - destination server port

setPort

```
public void setPort(String port)
```

getPushPublishVersionStr

```
public static String getPushPublishVersionStr()
```

Gets the push publish version str.

(continued from last page)

Returns:

the push publish version str

load

```
public void load(java.util.HashMap dataMap)
```

getNumberFromString

```
protected static int getNumberFromString(String in)
```

getMapFileIndex

```
public int getMapFileIndex()
```

setMapFileIndex

```
public void setMapFileIndex(String index)
```

getReconnectWaitTime

```
public int getReconnectWaitTime()
```

Get the reconnect wait time (milliseconds). Shortest time between connection attempts.

Returns:

reconnect wait time (milliseconds)

setReconnectWaitTime

```
public void setReconnectWaitTime(int reconnectWaitTime)
```

Set the reconnect wait time (milliseconds). Shortest time between connection attempts.

Parameters:

reconnectWaitTime - reconnect wait time (milliseconds)

getBandwidth

```
public double getBandwidth()
```

getPushPublishType

```
public int getPushPublishType()
```

createPushPublishSession

```
public IPushPublishSession createPushPublishSession()
```

getSrcStream

```
public IMediaStream getSrcStream()
```

setSrcStream

```
public void setSrcStream(IMediaStream srcStream)
```

getContextStr

```
public String getContextStr()
```

com.wowza.wms.pushpublish.model Class PushPublishContext

java.lang.Object

└─com.wowza.wms.pushpublish.model.PushPublishContext

public class **PushPublishContext**
extends Object

Field Summary

public static final	CLASS
public static final	CLASSNAME Value: PushPublishContext

Constructor Summary

public	PushPublishContext (IVHost vhost)
--------	--

Method Summary

void	doWatchdog ()
IPushPublishCrossSession	getCrossSession (String contextStr)
Object	getCrossSessionLock ()
int	getNextMapIndex ()
java.util.HashMap	getProfileData (String profileName)
void	putCrossSession (String contextStr, IPushPublishCrossSession crossSession)
void	putProfileData (String profileName, java.util.HashMap profileData)
IPushPublishCrossSession	removeCrossSession (String contextStr)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

(continued from last page)

Fields

CLASS

```
public static final java.lang.Class CLASS
```

CLASSNAME

```
public static final java.lang.String CLASSNAME
```

Constant value: **PushPublishContext**

Constructors

PushPublishContext

```
public PushPublishContext(IVHost vhost)
```

Methods

getProfileData

```
public java.util.HashMap getProfileData(String profileName)
```

putProfileData

```
public void putProfileData(String profileName,  
    java.util.HashMap profileData)
```

getNextMapIndex

```
public int getNextMapIndex()
```

putCrossSession

```
public void putCrossSession(String contextStr,  
    IPushPublishCrossSession crossSession)
```

getCrossSession

```
public IPushPublishCrossSession getCrossSession(String contextStr)
```

(continued from last page)

removeCrossSession

```
public IPushPublishCrossSession removeCrossSession(String contextStr)
```

doWatchdog

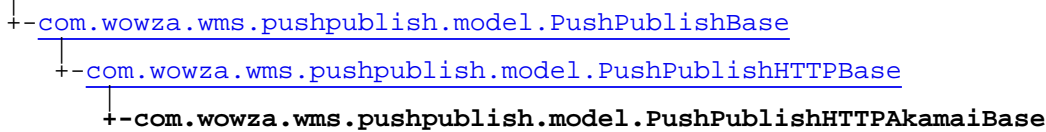
```
public void doWatchdog()
```

getCrossSessionLock

```
public Object getCrossSessionLock()
```


com.wowza.wms.pushpublish.model Class PushPublishHTTPAkamaiBase

java.lang.Object



All Implemented Interfaces:

[IPushPublish](#)

public abstract class **PushPublishHTTPAkamaiBase**

extends [PushPublishHTTTPBase](#)

Field Summary

public static final	AKAMAI_DEFAULT_HOSTID Value: example-i
protected	akamaiConfigName
protected	akamaiEventName
protected	akamaiHostId
protected	akamaiIngressURLPrefix
protected	akamaiPlaybackURLPrefix
protected	akamaiProtocolName
protected	akamaiStreamId
public static final	CLASS
public static final	CLASSNAME Value: PushPublishHTTPAkamaiBase
protected	httpIngressDomainOverride
protected	httpIngressDomainSuffix
protected	httpPlaybackDomainOverride
protected	httpPlaybackDomainSuffix

protected	redundant
protected	sendToBackupServer

Fields inherited from class [com.wowza.wms.pushpublish.model.PushPublishHTTPBase](#)

[CLASS](#), [CLASSNAME](#), [connectionTimeout](#), [fakePosts](#), [httpWriterDebug](#), [maxRetries](#), [MIN_PLAYLIST_COUNT](#), [playlistAcrossSessions](#), [playlistCountFromMap](#), [playlistCountFromPacketizer](#), [playlistCrossName](#), [playlistMap](#), [playlistName](#), [playlistTimeout](#), [pushPublishHTTPPlayListDebug](#), [readWriteTimeout](#), [reuseSockets](#)

Fields inherited from class [com.wowza.wms.pushpublish.model.PushPublishBase](#)

[appInstance](#), [CLASS](#), [CLASSNAME](#), [dstStreamName](#), [hostname](#), [inetAddr](#), [mapFileIndex](#), [mImplementation](#), [mType](#), [originalTimecodeThreshold](#), [password](#), [port](#), [properties](#), [pushPublishStreamDebug](#), [pushPublishType](#), [reconnectWaitTime](#), [sendingPerformance](#), [sendOriginalTimecodes](#), [srcStream](#), [srcStreamName](#), [userName](#)

Fields inherited from interface [com.wowza.wms.pushpublish.model.IPushPublish](#)

[PUSHPUBLISH_TYPE_RTMP](#), [PUSHPUBLISH_TYPE_RTP](#), [PUSHPUBLISH_TYPE_UNKNOWN](#)

Constructor Summary

public	PushPublishHTTPAkamaiBase()
--------	---

Method Summary

String	getAkamaiConfigName()
String	getAkamaiEventName()
String	getAkamaiHostId()
String	getAkamaiIngressAdaptiveGroupPath() (String groupName)
String	getAkamaiIngressAdaptiveGroupURL() (String groupName)
String	getAkamaiIngressURL()
String	getAkamaiPlaybackAdaptiveGroupURL() (String groupName)
String	getAkamaiPlaybackDomain()
String	getAkamaiPlaybackURL()
String	getAkamaiPlaybackURLPrefix()
String	getAkamaiStreamId()

String	<u>getIngressDomain()</u>
String	<u>getIngressPath()</u>
int	<u>getIngressPort()</u>
String	<u>getResolvedAkamaiStreamId()</u>
void	<u>init()</u> (<u>IApplicationInstance</u> appInstance, String streamName, <u>IMediaStream</u> stream, java.util.HashMap profileData, java.util.HashMap maps, boolean streamDebug)
boolean	<u>isInRedundancyPair()</u>
boolean	<u>isRedundant()</u>
boolean	<u>isSendToBackupServer()</u>
void	<u>load()</u> (java.util.HashMap dataMap)
void	<u>resetConnection()</u>
void	<u>setAkamaiConfigName()</u> (String akamaiConfigName)
void	<u>setAkamaiEventName()</u> (String akamaiEventName)
void	<u>setAkamaiHostId()</u> (String akamaiHostId)
void	<u>setAkamaiPlaybackURLPrefix()</u> (String akamaiPlaybackURLPrefix)
void	<u>setAkamaiStreamId()</u> (String akamaiStreamId)
void	<u>setRedundant()</u> (boolean redundant)
void	<u>setSendToBackupServer()</u> (boolean sendToBackupServer)

Methods inherited from class [com.wowza.wms.pushpublish.model.PushPublishHTTPBase](#)

[addAdaptiveGroupItem](#), [getAdaptiveGroupItems](#), [getConnectionTimeout](#), [getMaxRetries](#), [getPlaylistCount](#), [getPlaylistTimeout](#), [getReadWriteTimeout](#), [init](#), [isFakePosts](#), [isHttpWriterDebug](#), [isInAdaptiveGroup](#), [isPlaylistAcrossSessions](#), [isReuseSockets](#), [load](#), [setConnectionTimeout](#), [setFakePosts](#), [setHttpWriterDebug](#), [setMaxRetries](#), [setPlaylistAcrossSessions](#), [setPlaylistCountFromMap](#), [setPlaylistCountFromPacketizer](#), [setPlaylistTimeout](#), [setReadWriteTimeout](#), [setReuseSockets](#)

Methods inherited from class [com.wowza.wms.pushpublish.model.PushPublishBase](#)

[createPushPublishSession](#), [getAppInstance](#), [getBandwidth](#), [getConfigDataOrDefault](#), [getConfigDataOrDefault](#), [getConfigDataOrDefault](#), [getContextStr](#), [getDstStreamName](#), [getHostname](#), [getImplementation](#), [getMapFileIndex](#), [getNumberFromString](#), [getOriginalTimecodeThreshold](#), [getPassword](#), [getPort](#), [getProperties](#), [getPushPublishType](#), [getPushPublishVersionStr](#), [getReconnectWaitTime](#), [getSrcStream](#), [getSrcStreamName](#), [getUserName](#), [init](#), [isAdaptiveStreaming](#), [isSendOriginalTimecodes](#), [load](#), [setAdaptiveStreaming](#), [setAppInstance](#), [setDebugLog](#), [setDstStreamName](#), [setHost](#), [setHostname](#), [setImplementation](#), [setMapFileIndex](#), [setOriginalTimecodeThreshold](#), [setOriginalTimecodeThreshold](#), [setPassword](#), [setPort](#), [setPort](#), [setReconnectWaitTime](#), [setSendOriginalTimecodes](#), [setSrcStream](#), [setSrcStreamName](#), [setStreamName](#), [setUserName](#)

Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Methods inherited from interface [com.wowza.wms.pushpublish.model.IPushPublish](#)

[connect](#), [createPushPublishSession](#), [disconnect](#), [getContextStr](#), [getDstStreamName](#), [getHostname](#), [getMapFileIndex](#), [getMessagesAttemptedCount](#), [getMessagesFailedCount](#), [getMessagesRetriedCount](#), [getPort](#), [getProperties](#), [getPushPublishType](#), [getReconnectWaitTime](#), [getSrcStream](#), [getSrcStreamName](#), [init](#), [isConnected](#), [resetConnection](#), [setDebugLog](#), [setDstStreamName](#), [setHost](#), [setHostname](#), [setMapFileIndex](#), [setPort](#), [setPort](#), [setReconnectWaitTime](#), [setSrcStream](#), [setSrcStreamName](#), [setStreamName](#)

Fields

CLASS

public static final `java.lang.Class` **CLASS**

CLASSNAME

public static final `java.lang.String` **CLASSNAME**

Constant value: **PushPublishHTTPAkamaiBase**

AKAMAI_DEFAULT_HOSTID

public static final `java.lang.String` **AKAMAI_DEFAULT_HOSTID**

Constant value: **example-i**

sendToBackupServer

protected boolean **sendToBackupServer**

(continued from last page)

redundant

protected boolean **redundant**

httpIngressDomainSuffix

protected java.lang.String **httpIngressDomainSuffix**

httpIngressDomainOverride

protected java.lang.String **httpIngressDomainOverride**

httpPlaybackDomainSuffix

protected java.lang.String **httpPlaybackDomainSuffix**

httpPlaybackDomainOverride

protected java.lang.String **httpPlaybackDomainOverride**

akamaiStreamId

protected java.lang.String **akamaiStreamId**

akamaiHostId

protected java.lang.String **akamaiHostId**

akamaiIngressURLPrefix

protected java.lang.String **akamaiIngressURLPrefix**

akamaiPlaybackURLPrefix

protected java.lang.String **akamaiPlaybackURLPrefix**

akamaiEventName

protected java.lang.String **akamaiEventName**

(continued from last page)

akamaiConfigName

```
protected java.lang.String akamaiConfigName
```

akamaiProtocolName

```
protected java.lang.String akamaiProtocolName
```

Constructors

PushPublishHTTPAkamaiBase

```
public PushPublishHTTPAkamaiBase()
```

Methods

init

```
public void init(IApplicationInstance appInstance,  
                String streamName,  
                IMediaStream stream,  
                java.util.HashMap profileData,  
                java.util.HashMap maps,  
                boolean streamDebug)
```

Inits the instance.

getResolvedAkamaiStreamId

```
protected String getResolvedAkamaiStreamId()
```

getIngressDomain

```
public String getIngressDomain()
```

isSendToBackupServer

```
public boolean isSendToBackupServer()
```

setSendToBackupServer

```
public void setSendToBackupServer(boolean sendToBackupServer)
```

isInRedundancyPair

```
public boolean isInRedundancyPair()
```

isRedundant

```
public boolean isRedundant()
```

setRedundant

```
public void setRedundant(boolean redundant)
```

getAkamaiStreamId

```
public String getAkamaiStreamId()
```

setAkamaiStreamId

```
public void setAkamaiStreamId(String akamaiStreamId)
```

getIngressPort

```
public int getIngressPort()
```

getAkamaiPlaybackDomain

```
public String getAkamaiPlaybackDomain()
```

getIngressPath

```
public String getIngressPath()
```

getAkamaiIngressURL

```
public String getAkamaiIngressURL()
```

getAkamaiIngressAdaptiveGroupPath

```
public String getAkamaiIngressAdaptiveGroupPath(String groupName)
```

(continued from last page)

getAkamaiIngressAdaptiveGroupURL

```
public String getAkamaiIngressAdaptiveGroupURL(String groupName)
```

getAkamaiPlaybackAdaptiveGroupURL

```
public String getAkamaiPlaybackAdaptiveGroupURL(String groupName)
```

getAkamaiPlaybackURL

```
public String getAkamaiPlaybackURL()
```

resetConnection

```
public void resetConnection()
```

getAkamaiHostId

```
public String getAkamaiHostId()
```

setAkamaiHostId

```
public void setAkamaiHostId(String akamaiHostId)
```

getAkamaiEventName

```
public String getAkamaiEventName()
```

setAkamaiEventName

```
public void setAkamaiEventName(String akamaiEventName)
```

getAkamaiConfigName

```
public String getAkamaiConfigName()
```

(continued from last page)

setAkamaiConfigName

```
public void setAkamaiConfigName(String akamaiConfigName)
```

getAkamaiPlaybackURLPrefix

```
public String getAkamaiPlaybackURLPrefix()
```

setAkamaiPlaybackURLPrefix

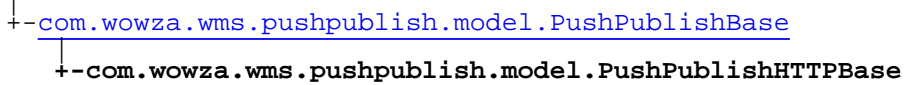
```
public void setAkamaiPlaybackURLPrefix(String akamaiPlaybackURLPrefix)
```

load

```
public void load(java.util.HashMap dataMap)
```

com.wowza.wms.pushpublish.model Class PushPublishHTTPBase

java.lang.Object



All Implemented Interfaces:

[IPushPublish](#)

Direct Known Subclasses:

[PushPublishHTTPAkamaiBase](#)

public abstract class **PushPublishHTTPBase**

extends [PushPublishBase](#)

Field Summary

public static final	CLASS
public static final	CLASSNAME Value: PushPublishHTTPBase
protected	connectionTimeout
protected	fakePosts
protected	httpWriterDebug
protected	maxRetries
protected static final	MIN_PLAYLIST_COUNT Value: 3
protected	playlistAcrossSessions
protected	playlistCountFromMap
protected	playlistCountFromPacketizer
protected	playlistCrossName
protected	playlistMap
protected	playlistName

protected	playlistTimeout
protected	pushPublishHTTPPlayListDebug
protected	readWriteTimeout
protected	reuseSockets

Fields inherited from class [com.wowza.wms.pushpublish.model.PushPublishBase](#)

[appInstance](#), [CLASS](#), [CLASSNAME](#), [dstStreamName](#), [hostname](#), [inetAddr](#), [mapFileIndex](#), [mImplementation](#), [mType](#), [originalTimecodeThreshold](#), [password](#), [port](#), [properties](#), [pushPublishStreamDebug](#), [pushPublishType](#), [reconnectWaitTime](#), [sendingPerformance](#), [sendOriginalTimecodes](#), [srcStream](#), [srcStreamName](#), [userName](#)

Fields inherited from interface [com.wowza.wms.pushpublish.model.IPushPublish](#)

[PUSHPUBLISH_TYPE_RTMP](#), [PUSHPUBLISH_TYPE_RTP](#), [PUSHPUBLISH_TYPE_UNKNOWN](#)

Constructor Summary

public	PushPublishHTTPBase()
--------	---------------------------------------

Method Summary

void	addAdaptiveGroupItem (String adaptiveGroupName)
java.util.List	getAdaptiveGroupItems ()
int	getConnectionTimeout ()
int	getMaxRetries ()
int	getPlaylistCount ()
int	getPlaylistTimeout ()
int	getReadWriteTimeout ()
void	init (IApplicationInstance appInstance, String streamName, IMediaStream stream, java.util.HashMap profileData, java.util.HashMap maps, boolean streamDebug)
boolean	isFakePosts ()
boolean	isHttpWriterDebug ()
boolean	isInAdaptiveGroup ()
boolean	isPlaylistAcrossSessions ()

boolean	<u>isReuseSockets()</u>
void	<u>load</u> (java.util.HashMap dataMap)
void	<u>setConnectionTimeout</u> (int connectionTimeout)
void	<u>setFakePosts</u> (boolean fakePosts)
void	<u>setHttpWriterDebug</u> (boolean httpWriterDebug)
void	<u>setMaxRetries</u> (int maxRetries)
void	<u>setPlaylistAcrossSessions</u> (boolean playlistAcrossSessions)
void	<u>setPlaylistCountFromMap</u> (int playlistCountFromMap)
void	<u>setPlaylistCountFromPacketizer</u> (int playlistCountFromPacketizer)
void	<u>setPlaylistTimeout</u> (int playlistTimeout)
void	<u>setReadWriteTimeout</u> (int readTimeout)
void	<u>setReuseSockets</u> (boolean reuseSockets)

Methods inherited from class [com.wowza.wms.pushpublish.model.PushPublishBase](#)

[createPushPublishSession](#), [getAppInstance](#), [getBandwidth](#), [getConfigDataOrDefault](#), [getConfigDataOrDefault](#), [getConfigDataOrDefault](#), [getContextStr](#), [getDstStreamName](#), [getHostname](#), [getImplementation](#), [getMapFileIndex](#), [getNumberFromString](#), [getOriginalTimecodeThreshold](#), [getPassword](#), [getPort](#), [getProperties](#), [getPushPublishType](#), [getPushPublishVersionStr](#), [getReconnectWaitTime](#), [getSrcStream](#), [getSrcStreamName](#), [getUserName](#), [init](#), [isAdaptiveStreaming](#), [isSendOriginalTimecodes](#), [load](#), [setAdaptiveStreaming](#), [setAppInstance](#), [setDebugLog](#), [setDstStreamName](#), [setHost](#), [setHostname](#), [setImplementation](#), [setMapFileIndex](#), [setOriginalTimecodeThreshold](#), [setOriginalTimecodeThreshold](#), [setPassword](#), [setPort](#), [setPort](#), [setReconnectWaitTime](#), [setSendOriginalTimecodes](#), [setSrcStream](#), [setSrcStreamName](#), [setStreamName](#), [setUserName](#)

Methods inherited from class [java.lang.Object](#)

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

Methods inherited from interface [com.wowza.wms.pushpublish.model.IPushPublish](#)

[connect](#), [createPushPublishSession](#), [disconnect](#), [getContextStr](#), [getDstStreamName](#), [getHostname](#), [getMapFileIndex](#), [getMessagesAttemptedCount](#), [getMessagesFailedCount](#), [getMessagesRetriedCount](#), [getPort](#), [getProperties](#), [getPushPublishType](#), [getReconnectWaitTime](#), [getSrcStream](#), [getSrcStreamName](#), [init](#), [isConnected](#), [resetConnection](#), [setDebugLog](#), [setDstStreamName](#), [setHost](#), [setHostname](#), [setMapFileIndex](#), [setPort](#), [setPort](#), [setReconnectWaitTime](#), [setSrcStream](#), [setSrcStreamName](#), [setStreamName](#)

Fields

CLASS

```
public static final java.lang.Class CLASS
```

CLASSNAME

```
public static final java.lang.String CLASSNAME
```

Constant value: **PushPublishHTTPBase**

MIN_PLAYLIST_COUNT

```
protected static final int MIN_PLAYLIST_COUNT
```

Constant value: **3**

playlistCountFromMap

```
protected int playlistCountFromMap
```

playlistCountFromPacketizer

```
protected int playlistCountFromPacketizer
```

playlistTimeout

```
protected int playlistTimeout
```

playlistAcrossSessions

```
protected boolean playlistAcrossSessions
```

fakePosts

```
protected boolean fakePosts
```

reuseSockets

```
protected boolean reuseSockets
```

(continued from last page)

maxRetries

protected int **maxRetries**

connectionTimeout

protected int **connectionTimeout**

readWriteTimeout

protected int **readWriteTimeout**

httpWriterDebug

protected boolean **httpWriterDebug**

playlistMap

protected com.wowza.wms.pushpublish.model.PushPublishHTTPPlaylistMap **playlistMap**

playlistCrossName

protected java.lang.String **playlistCrossName**

playlistName

protected java.lang.String **playlistName**

pushPublishHTTPPlayListDebug

protected boolean **pushPublishHTTPPlayListDebug**

Constructors

PushPublishHTTPBase

public **PushPublishHTTPBase**()

(continued from last page)

Methods

init

```
public void init(IApplicationInstance appInstance,  
                String streamName,  
                IMediaStream stream,  
                java.util.HashMap profileData,  
                java.util.HashMap maps,  
                boolean streamDebug)
```

Init's the instance.

setPlaylistCountFromMap

```
public void setPlaylistCountFromMap(int playlistCountFromMap)
```

setPlaylistCountFromPacketizer

```
public void setPlaylistCountFromPacketizer(int playlistCountFromPacketizer)
```

getPlaylistCount

```
public int getPlaylistCount()
```

getPlaylistTimeout

```
public int getPlaylistTimeout()
```

setPlaylistTimeout

```
public void setPlaylistTimeout(int playlistTimeout)
```

isPlaylistAcrossSessions

```
public boolean isPlaylistAcrossSessions()
```

setPlaylistAcrossSessions

```
public void setPlaylistAcrossSessions(boolean playlistAcrossSessions)
```

isFakePosts

```
public boolean isFakePosts()
```

(continued from last page)

setFakePosts

```
public void setFakePosts(boolean fakePosts)
```

isReuseSockets

```
public boolean isReuseSockets()
```

setReuseSockets

```
public void setReuseSockets(boolean reuseSockets)
```

isHttpWriterDebug

```
public boolean isHttpWriterDebug()
```

setHttpWriterDebug

```
public void setHttpWriterDebug(boolean httpWriterDebug)
```

getConnectionTimeout

```
public int getConnectionTimeout()
```

setConnectionTimeout

```
public void setConnectionTimeout(int connectionTimeout)
```

getReadWriteTimeout

```
public int getReadWriteTimeout()
```

setReadWriteTimeout

```
public void setReadWriteTimeout(int readTimeout)
```

(continued from last page)

getMaxRetries

```
public int getMaxRetries()
```

setMaxRetries

```
public void setMaxRetries(int maxRetries)
```

addAdaptiveGroupItem

```
public void addAdaptiveGroupItem(String adaptiveGroupName)
```

getAdaptiveGroupItems

```
public java.util.List getAdaptiveGroupItems()
```

isInAdaptiveGroup

```
public boolean isInAdaptiveGroup()
```

load

```
public void load(java.util.HashMap dataMap)
```

com.wowza.wms.pushpublish.model Class PushPublishHTTPOutputItem

java.lang.Object

└-com.wowza.wms.pushpublish.model.PushPublishHTTPOutputItem

```
public class PushPublishHTTPOutputItem
extends Object
```

Field Summary

public	chunkLists
public	chunks
public	liveStreamPacketizer
public	playlistStrA
public	playlistStrB

Constructor Summary

public	PushPublishHTTPOutputItem (String[] chunkLists)
--------	---

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

playlistStrA

```
public java.lang.String playlistStrA
```

playlistStrB

```
public java.lang.String playlistStrB
```

chunkLists

```
public java.lang.String chunkLists
```

(continued from last page)

liveStreamPacketizer

```
public com.wowza.wms.stream.livepacketizer.ILiveStreamPacketizer liveStreamPacketizer
```

chunks

```
public java.util.List chunks
```

Constructors

PushPublishHTTPOutputItem

```
public PushPublishHTTPOutputItem(String[] chunkLists)
```

com.wowza.wms.pushpublish.model Class PushPublishHTTPOutputItemChunk

java.lang.Object

└-com.wowza.wms.pushpublish.model.PushPublishHTTPOutputItemChunk

```
public class PushPublishHTTPOutputItemChunk
    extends Object
```

Field Summary

public	chunk
public	playlistEntry
public	renditionIndex

Constructor Summary

public	PushPublishHTTPOutputItemChunk (int renditionIndex, PushPublishHTTPPlaylistEntry playlistEntry)
--------	---

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

renditionIndex

```
public int renditionIndex
```

chunk

```
public java.lang.Object chunk
```

playlistEntry

```
public com.wowza.wms.pushpublish.model.PushPublishHTTPPlaylistEntry playlistEntry
```

Constructors

(continued from last page)

PushPublishHTTPOutputItemChunk

```
public PushPublishHTTPOutputItemChunk(int renditionIndex,  
                                       PushPublishHTTPPlaylistEntry playlistEntry)
```

com.wowza.wms.pushpublish.model Class PushPublishHTTPPlaylistEntry

java.lang.Object

└-com.wowza.wms.pushpublish.model.PushPublishHTTPPlaylistEntry

public abstract class **PushPublishHTTPPlaylistEntry**
extends Object

Field Summary

public	delete
--------	------------------------

Constructor Summary

public	PushPublishHTTPPlaylistEntry()
--------	--

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

delete

public boolean **delete**

Constructors

PushPublishHTTPPlaylistEntry

public **PushPublishHTTPPlaylistEntry()**

com.wowza.wms.pushpublish.model Class PushPublishHTTPPlaylistList

java.lang.Object

└─com.wowza.wms.pushpublish.model.PushPublishHTTPPlaylistList

public class **PushPublishHTTPPlaylistList**
extends Object

Field Summary

public	bitrate
public static final	CHUNKIDDATEOFFSET Value: 1404253012
public	entries
public	markEntryAsDiscontinuity
public	nextChunkId

Constructor Summary

public	PushPublishHTTPPlaylistList()
--------	---

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

CHUNKIDDATEOFFSET

public static final long **CHUNKIDDATEOFFSET**

Constant value: **1404253012**

entries

public java.util.List **entries**

(continued from last page)

nextChunkId

```
public long nextChunkId
```

markEntryAsDiscontinuity

```
public boolean markEntryAsDiscontinuity
```

bitrate

```
public long bitrate
```

Constructors

PushPublishHTTPPlaylistList

```
public PushPublishHTTPPlaylistList()
```


com.wowza.wms.pushpublish.model Class PushPublishHTTPPlaylistMap

java.lang.Object

└─com.wowza.wms.pushpublish.model.PushPublishHTTPPlaylistMap

All Implemented Interfaces:

[IPushPublishCrossSession](#)

public class **PushPublishHTTPPlaylistMap**
extends Object
implements [IPushPublishCrossSession](#)

Field Summary

public	lastUpdate
public	map

Constructor Summary

public	PushPublishHTTPPlaylistMap (PushPublishHTTPBase builder)
--------	---

Method Summary

PushPublishHTTPBase	getBuilder ()
boolean	isSessionValid ()
void	markAllAsDiscontinuity ()
void	setBuilder (PushPublishHTTPBase builder)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface [com.wowza.wms.pushpublish.model.IPushPublishCrossSession](#)

[isSessionValid](#)

Fields

(continued from last page)

map

```
public java.util.Map map
```

lastUpdate

```
public long lastUpdate
```

Constructors

PushPublishHTTPPlaylistMap

```
public PushPublishHTTPPlaylistMap(PushPublishHTTPBase builder)
```

Methods

isSessionValid

```
public boolean isSessionValid()
```

getBuilder

```
public PushPublishHTTPBase getBuilder()
```

setBuilder

```
public void setBuilder(PushPublishHTTPBase builder)
```

markAllAsDiscontinuity

```
public void markAllAsDiscontinuity()
```

com.wowza.wms.pushpublish.model Class PushPublishHTTPPusher

java.lang.Object

└─com.wowza.wms.pushpublish.model.PushPublishHTTPPusher

public class **PushPublishHTTPPusher**
extends Object

Nested Class Summary

class	PushPublishHTTPPusher.HTTPPoller PushPublishHTTPPusher.HTTPPoller
-------	--

Field Summary

public static final	CLASS
public static final	CLASSNAME Value: PushPublishHTTPPusher

Constructor Summary

public	PushPublishHTTPPusher (IPushPublishHTTPOutputBuilder builder, String name)
--------	---

Method Summary

void	addOutput (PushPublishHTTPOutputItem outputItem)
void	clearOutput ()
void	disconnect ()
String	getContextStr ()
ILiveStreamPacketizer	getLastPacketizer ()
void	resetNewPacketizer (IPushPublishHTTPOutputBuilder builder, ILiveStreamPacketizer liveStreamPacketizer)
void	startOutputRunner ()

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

CLASS

```
public static final java.lang.Class CLASS
```

CLASSNAME

```
public static final java.lang.String CLASSNAME
```

Constant value: **PushPublishHTTTPusher**

Constructors

PushPublishHTTTPusher

```
public PushPublishHTTTPusher(IPushPublishHTTPOutputBuilder builder,  
                             String name)
```

Methods

getContextStr

```
public String getContextStr()
```

disconnect

```
public void disconnect()
```

getLastPacketizer

```
public ILiveStreamPacketizer getLastPacketizer()
```

resetNewPacketizer

```
public void resetNewPacketizer(IPushPublishHTTPOutputBuilder builder,  
                               ILiveStreamPacketizer liveStreamPacketizer)
```

startOutputRunner

```
public void startOutputRunner()
```

clearOutput

```
public void clearOutput()
```

addOutput

```
public void addOutput(PushPublishHTTPOutputItem outputItem)
```

com.wowza.wms.pushpublish.model Class PushPublishHTTTPusher.HTTTPoller

java.lang.Object

└- java.lang.Thread

└- com.wowza.wms.pushpublish.model.PushPublishHTTTPusher.HTTTPoller

All Implemented Interfaces:

Runnable

```
public class PushPublishHTTTPusher.HTTTPoller
extends Thread
```

Field Summary

protected	didIt
protected	lastUpdateTime
protected	pollInterval
protected	sendPacketizerWarning

Fields inherited from class java.lang.Thread

MAX_PRIORITY, MIN_PRIORITY, NORM_PRIORITY

Constructor Summary

public	PushPublishHTTTPusher.HTTTPoller()
--------	--

Method Summary

void	run()
------	-----------------------

Methods inherited from class java.lang.Thread

activeCount, checkAccess, clone, countStackFrames, currentThread, destroy, dumpStack, enumerate, getAllStackTraces, getContextClassLoader, getDefaultUncaughtExceptionHandler, getId, getName, getPriority, getStackTrace, getState, getThreadGroup, getUncaughtExceptionHandler, holdsLock, interrupt, interrupted, isAlive, isDaemon, isInterrupted, join, join, join, resume, run, setContextClassLoader, setDaemon, setDefaultUncaughtExceptionHandler, setName, setPriority, setUncaughtExceptionHandler, sleep, sleep, start, stop, stop, suspend, toString, yield

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

Methods inherited from interface `java.lang.Runnable`

```
run
```

Fields

lastUpdateTime

```
protected long lastUpdateTime
```

didIt

```
protected boolean didIt
```

pollInterval

```
protected int pollInterval
```

sendPacketizerWarning

```
protected int sendPacketizerWarning
```

Constructors

PushPublishHTTPPusher.HTTPPoller

```
public PushPublishHTTPPusher.HTTPPoller()
```

Methods

run

```
public void run()
```

com.wowza.wms.pushpublish.model Class PushPublishSessionBase

java.lang.Object

└-com.wowza.wms.pushpublish.model.PushPublishSessionBase

All Implemented Interfaces:

[IPushPublishSession](#)

Direct Known Subclasses:

[PushPublishSessionRTP](#), [PushPublishSessionRTMP](#), [PushPublishSessionHTTPBase](#)

```
public class PushPublishSessionBase
  extends Object
  implements IPushPublishSession
```

Field Summary

protected	profile
protected	pushPublishImpl
protected	stream
protected	type

Fields inherited from interface [com.wowza.wms.pushpublish.model.IPushPublishSession](#)

[PUBLISHSESSION_TYPE_RTMP](#), [PUBLISHSESSION_TYPE_RTP](#),
[PUBLISHSESSION_TYPE_UNKNOWN](#)

Constructor Summary

public	PushPublishSessionBase()
--------	--

Method Summary

String	getContextStr()
String	getProfile()
IPushPublish	getPushPublishImpl()
IMediaStream	getStream()
int	getType()

void	setProfile (String profile)
void	setPushPublishImpl (IPushPublish pushPublishImpl)
void	setStream (IMediaStream stream)

Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Methods inherited from interface [com.wowza.wms.pushpublish.model.IPushPublishSession](#)

[getContextStr](#), [getProfile](#), [getPushPublishImpl](#), [getStream](#), [getType](#), [setProfile](#), [setPushPublishImpl](#), [setStream](#)

Fields

type
protected int **type**

pushPublishImpl
protected `com.wowza.wms.pushpublish.model.IPushPublish` **pushPublishImpl**

stream
protected `com.wowza.wms.stream.IMediaStream` **stream**

profile
protected `java.lang.String` **profile**

Constructors

PushPublishSessionBase
public **PushPublishSessionBase**()

Methods

(continued from last page)

getType

```
public int getType()
```

getPushPublishImpl

```
public IPushPublish getPushPublishImpl()
```

setPushPublishImpl

```
public void setPushPublishImpl(IPushPublish pushPublishImpl)
```

getStream

```
public IMediaStream getStream()
```

setStream

```
public void setStream(IMediaStream stream)
```

getProfile

```
public String getProfile()
```

setProfile

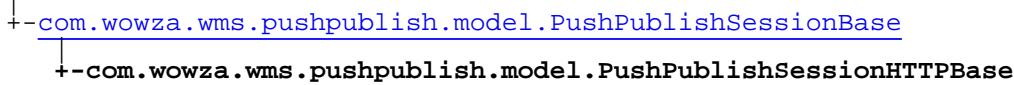
```
public void setProfile(String profile)
```

getContextStr

```
public String getContextStr()
```

com.wowza.wms.pushpublish.model Class PushPublishSessionHTTPBase

java.lang.Object



All Implemented Interfaces:
[IPushPublishSession](#)

public abstract class **PushPublishSessionHTTPBase**
extends [PushPublishSessionBase](#)

Fields inherited from class [com.wowza.wms.pushpublish.model.PushPublishSessionBase](#)

[profile](#), [pushPublishImpl](#), [stream](#), [type](#)

Fields inherited from interface [com.wowza.wms.pushpublish.model.IPushPublishSession](#)

[PUSHPUBLISHSESSION_TYPE_RTMP](#), [PUSHPUBLISHSESSION_TYPE_RTP](#),
[PUSHPUBLISHSESSION_TYPE_UNKNOWN](#)

Constructor Summary

public	PushPublishSessionHTTPBase()
--------	--

Methods inherited from class [com.wowza.wms.pushpublish.model.PushPublishSessionBase](#)

[getContextStr](#), [getProfile](#), [getPushPublishImpl](#), [getStream](#), [getType](#), [setProfile](#),
[setPushPublishImpl](#), [setStream](#)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait,
wait

Methods inherited from interface [com.wowza.wms.pushpublish.model.IPushPublishSession](#)

[getContextStr](#), [getProfile](#), [getPushPublishImpl](#), [getStream](#), [getType](#), [setProfile](#),
[setPushPublishImpl](#), [setStream](#)

Constructors

PushPublishSessionHTTPBase

public **PushPublishSessionHTTPBase()**

com.wowza.wms.pushpublish.model Class PushPublishSessionNotifyBase

java.lang.Object

└─com.wowza.wms.pushpublish.model.PushPublishSessionNotifyBase

All Implemented Interfaces:

[IPushPublishSessionNotify](#)

public abstract class **PushPublishSessionNotifyBase**

extends Object

implements [IPushPublishSessionNotify](#)

Abstract base class for [IPushPublishSessionNotify](#) interface

See Also:

[IApplicationInstance.addPushPublishSessionListener\(IPushPublishSessionNotify\)](#)

Constructor Summary

public	PushPublishSessionNotifyBase()
--------	--

Method Summary

void	onPushPublishSessionCreate (String streamName, IMediaStream stream, IPushPublishSession pushPublishSession)
void	onPushPublishSessionDestroy (String streamName, IMediaStream stream, IPushPublishSession pushPublishSession)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface [com.wowza.wms.pushpublish.model.IPushPublishSessionNotify](#)

[onPushPublishSessionCreate](#), [onPushPublishSessionDestroy](#)

Constructors

PushPublishSessionNotifyBase

public **PushPublishSessionNotifyBase**()

Methods

(continued from last page)

onPushPublishSessionCreate

```
public void onPushPublishSessionCreate(String streamName,  
    IMediaStream stream,  
    IPushPublishSession pushPublishSession)
```

onPushPublishSessionDestroy

```
public void onPushPublishSessionDestroy(String streamName,  
    IMediaStream stream,  
    IPushPublishSession pushPublishSession)
```

Package

com.wowza.wms.pushpublish.protocol.rtmp

com.wowza.wms.pushpublish.protocol.rtmp Interface IPushPublishRTMPAuthProvider

All Known Implementing Classes:

[PushPublishRTMPAuthProviderBase](#)

public interface **IPushPublishRTMPAuthProvider**
extends

Method Summary

String	addAuthQueryString (String queryString)
String	getPassword ()
PushPublishRTMP	getPublisher ()
String	getUserName ()
void	handleConnectResponse (com.wowza.wms.request.RequestFunction function, AMFDataList params)
void	init (PushPublishRTMP publisher)
void	setPassword (String password)
void	setUserName (String userName)

Methods

init

public void **init**([PushPublishRTMP](#) publisher)

addAuthQueryString

public String **addAuthQueryString**(String queryString)

handleConnectResponse

public void **handleConnectResponse**(com.wowza.wms.request.RequestFunction function, [AMFDataList](#) params)

getUserName

```
public String getUserName()
```

setUserName

```
public void setUserName(String userName)
```

getPassword

```
public String getPassword()
```

setPassword

```
public void setPassword(String password)
```

getPublisher

```
public PushPublishRTMP getPublisher()
```


com.wowza.wms.pushpublish.protocol.rtmp Interface IPushPublishRTMPNotify

All Known Implementing Classes:

[PushPublishRTMPNotifyBase](#)

public interface **IPushPublishRTMPNotify**
extends

Method Summary

void	onAkamaiClientLogin (PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)
void	onAkamaiSetChallenge (PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)
void	onAkamaiSetOriginConnectionInfo (PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)
void	onConnect (PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)
void	onConnectFailure (PushPublishRTMPNetConnectionSession pushPublisherSession)
void	onConnectStart (PushPublishRTMPNetConnectionSession pushPublisherSession)
void	onConnectSuccess (PushPublishRTMPNetConnectionSession pushPublisherSession)
void	onFCAnnounce (PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)
void	onFCPublish (PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)
void	onHandshakeResult (PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)
void	onPublishHandlerPlay (PushPublishRTMPNetConnectionSession pushPublisherSession, java.io.OutputStream out, long[] playSizes)
void	onPushPublisherSessionCreate (PushPublishRTMPNetConnectionSession pushPublisherSession)

void	onPushPublisherSessionDestroy (PushPublishRTMPNetConnectionSession pushPublisherSession)
void	onReleaseStream (PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)
void	onSessionClosed (PushPublishRTMPNetConnectionSession pushPublisherSession)
void	onSessionIdle (PushPublishRTMPNetConnectionSession pushPublisherSession)
void	onSessionOpened (PushPublishRTMPNetConnectionSession pushPublisherSession)
void	onStreamCreate (PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)
void	onStreamOnPlayStatus (PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)
void	onStreamOnStatus (PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)
void	onValidateSession (PushPublishRTMPNetConnectionSession pushPublisherSession)
void	onValidateSessionResult (PushPublishRTMPNetConnectionSession pushPublisherSession, boolean result)

Methods

onConnectStart

```
public void onConnectStart(PushPublishRTMPNetConnectionSession pushPublisherSession)
```

onConnectSuccess

```
public void onConnectSuccess(PushPublishRTMPNetConnectionSession pushPublisherSession)
```

onConnectFailure

```
public void onConnectFailure(PushPublishRTMPNetConnectionSession pushPublisherSession)
```

(continued from last page)

onHandshakeResult

```
public void onHandshakeResult(PushPublishRTMPNetConnectionSession
    pushPublisherSession,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

onAkamaiSetChallenge

```
public void onAkamaiSetChallenge(PushPublishRTMPNetConnectionSession
    pushPublisherSession,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

onAkamaiSetOriginConnectionInfo

```
public void onAkamaiSetOriginConnectionInfo(PushPublishRTMPNetConnectionSession
    pushPublisherSession,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

onAkamaiClientLogin

```
public void onAkamaiClientLogin(PushPublishRTMPNetConnectionSession
    pushPublisherSession,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

onFCPublish

```
public void onFCPublish(PushPublishRTMPNetConnectionSession pushPublisherSession,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

onFCAnnounce

```
public void onFCAnnounce(PushPublishRTMPNetConnectionSession pushPublisherSession,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

onReleaseStream

```
public void onReleaseStream(PushPublishRTMPNetConnectionSession pushPublisherSession,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

(continued from last page)

onConnect

```
public void onConnect(PushPublishRTMPNetConnectionSession pushPublisherSession,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

onStreamCreate

```
public void onStreamCreate(PushPublishRTMPNetConnectionSession pushPublisherSession,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

onStreamOnPlayStatus

```
public void onStreamOnPlayStatus(PushPublishRTMPNetConnectionSession  
pushPublisherSession,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

onStreamOnStatus

```
public void onStreamOnStatus(PushPublishRTMPNetConnectionSession pushPublisherSession,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

onPublishHandlerPlay

```
public void onPublishHandlerPlay(PushPublishRTMPNetConnectionSession  
pushPublisherSession,  
    java.io.OutputStream out,  
    long[] playSizes)
```

onSessionIdle

```
public void onSessionIdle(PushPublishRTMPNetConnectionSession pushPublisherSession)
```

onSessionOpened

```
public void onSessionOpened(PushPublishRTMPNetConnectionSession pushPublisherSession)
```

onSessionClosed

```
public void onSessionClosed(PushPublishRTMPNetConnectionSession pushPublisherSession)
```

onValidateSession

```
public void onValidateSession(PushPublishRTMPNetConnectionSession
pushPublisherSession)
```

onValidateSessionResult

```
public void onValidateSessionResult(PushPublishRTMPNetConnectionSession
pushPublisherSession,
    boolean result)
```

onPushPublisherSessionCreate

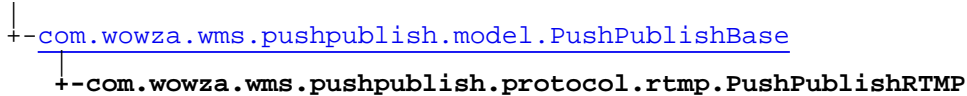
```
public void onPushPublisherSessionCreate(PushPublishRTMPNetConnectionSession
pushPublisherSession)
```

onPushPublisherSessionDestroy

```
public void onPushPublisherSessionDestroy(PushPublishRTMPNetConnectionSession
pushPublisherSession)
```

com.wowza.wms.pushpublish.protocol.rtmp Class PushPublishRTMP

java.lang.Object



All Implemented Interfaces:

[IPushPublish](#)

Direct Known Subclasses:

[PushPublishRTMPProfileHandler](#)

```
public class PushPublishRTMP
extends PushPublishBase
```

PushPublisher

Sample Code to push to Wowza Server or Adobe Media Server (no authentication)

This method should work with **Level 3** as well as long as the publishing application on not authenticated.

```
try
{
    PushPublishRTMP publisher = new PushPublishRTMP();

    // Source stream
    publisher.setAppInstance(appInstance);
    publisher.setSrcStreamName("myStream");

    // Destination stream
    publisher.setHostname("localhost");
    publisher.setPort(1935);
    publisher.setDstApplicationName("live");
    publisher.setDstStreamName("myStream");

    // Uncomment if pushing to Adobe Media Server
    //publisher.setSendOriginalTimecodes(true);
    //publisher.setOriginalTimecodeThreshold(0x100000);

    // SecureToken shared secret
    //publisher.setSecureTokenSharedSecret("#ed%h0#w@1");

    publisher.setSendFCPublish(true);
    publisher.setSendReleaseStream(true);
    publisher.setDebugLog(true);

    publisher.connect();
}
catch(Exception e)
{
    WMSLoggerFactory.getLogger(null).info(CLASSNAME+"RTMP: ", e);
}
```

Sample Code to push to Wowza Server or Adobe Media Server (RTMP username/password authentication)

This method should work with **Level 3** as well if the publishing application is protected using RTMP authentication.

```
try
{
    PushPublishRTMP publisher = new PushPublishRTMP();

    // Source stream
    publisher.setAppInstance(appInstance);
    publisher.setSrcStreamName("myStream");

    // Destination stream
    publisher.setHostname("localhost");
    publisher.setPort(1935);
    publisher.setDstApplicationName("live");
    publisher.setDstStreamName("myStream");
    publisher.setConnectionFlashVersion(PushPublishRTMP.CURRENTFMLEVERSION);

    // Uncomment if pushing to Adobe Media Server
    //publisher.setSendOriginalTimecodes(true);
    //publisher.setOriginalTimecodeThreshold(0x100000);

    // SecureToken shared secret
    //publisher.setSecureTokenSharedSecret("#ed%h0#w@1");

    publisher.setSendFCPublish(true);
    publisher.setSendReleaseStream(true);
    publisher.setDebugLog(true);

    PushPublishRTMPAuthProviderAdobe adobeRTMPAuthProvider = new
PushPublishRTMPAuthProviderAdobe();

    adobeRTMPAuthProvider.init(publisher);
    adobeRTMPAuthProvider.setUserName("username");
    adobeRTMPAuthProvider.setPassword("password");
    publisher.setRTMPAuthProvider(adobeRTMPAuthProvider);

    publisher.connect();
}
catch(Exception e)
{
    WMSLoggerFactory.getLogger(null).info(CLASSNAME+"RTMP: ", e);
}
```

Sample Code to push to Akamai


```
try
{
    PushPublishRTMP publisher = new PushPublishRTMP();

    // Source stream
    publisher.setAppInstance(appInstance);
    publisher.setSrcStreamName("myStream");

    // Destination stream
    String streamId = "12345";
    String akamaiUsername = "56789";
    String akamaiPassword = "changeme";
    boolean isPrimary = true;
    String hostname = (isPrimary?"p":"b")+".ep"+streamId+".i.akamaientrypoint.net";
    String dstApplicationName = "EntryPoint";
    String dstStreamName = "myStream"+"@"+streamId;

    publisher.setHostname(hostname);
    publisher.setPort(1935);
    publisher.setDstApplicationName(dstApplicationName);
    publisher.setDstStreamName(dstStreamName);
    publisher.setDebugLog(true);

    publisher.setConnectionFlashVersion(PushPublishRTMP.getPushPublishVersionStr());

    if (PushPublishRTMP.isFlashVersionFMLE(publisher.getConnectionFlashVersion()))
    {
        PushPublishRTMPAuthProviderAdobe adobeRTMPAuthProvider = new
PushPublishRTMPAuthProviderAdobe();

        adobeRTMPAuthProvider.init(publisher);
        adobeRTMPAuthProvider.setUserName(akamaiUsername);
        adobeRTMPAuthProvider.setPassword(akamaiPassword);

        publisher.setRTMPAuthProvider(adobeRTMPAuthProvider);
    }
    else
    {
        publisher.setAkamaiUserName(akamaiUsername);
        publisher.setAkamaiPassword(akamaiPassword);
    }

    publisher.setSendFCPublish(true);
    publisher.setSendReleaseStream(true);
    publisher.setSendStreamCloseCommands(true);

    publisher.setSendOriginalTimecodes(true);
    publisher.setOriginalTimecodeThreshold(0x100000);

    publisher.connect();
}
```

```
catch(Exception e)
{
    WMSLoggerFactory.getLogger(null).info(CLASSNAME+"RTMP: ", e);
}
```

Sample Code to push to **Limelight**

```
try
{
    PushPublishRTMP publisher = new PushPublishRTMP();

    // Source stream
    publisher.setAppInstance(appInstance);
    publisher.setSrcStreamName("myStream");

    // Destination stream
    publisher.setHostname("fmspush.lax.llnw.net");
    publisher.setPort(1935);
    publisher.setDstApplicationName("live");
    publisher.setDstStreamName("myStream");

    publisher.setSendFCPublish(true);
    publisher.setSendReleaseStream(true);
    publisher.setDebugLog(true);

    publisher.setSendOriginalTimecodes(true);
    publisher.setOriginalTimecodeThreshold(0x100000);

    PushPublishRTMPAuthProviderLimelight limelightRTMPAuthProvider = new
PushPublishRTMPAuthProviderLimelight();

    limelightRTMPAuthProvider.init(publisher);
    limelightRTMPAuthProvider.setUserName("username");
    limelightRTMPAuthProvider.setPassword("password");
    publisher.setRTMPAuthProvider(limelightRTMPAuthProvider);

    publisher.connect();
}
catch(Exception e)
{
    WMSLoggerFactory.getLogger(null).info(CLASSNAME+"RTMP: ", e);
}
```

Sample Code to push to **BitGravity** (streaming needs to be setup per instructions in **BG Live 3rd Party Encoder Configuration** guide.)

```
try
{
    PushPublishRTMP publisher = new PushPublishRTMP();

    // Source stream
    publisher.setAppInstance(client.getAppInstance());
    publisher.setSrcStreamName("myStream");

    // Destination stream
    publisher.setHostname("rtpdev1.iad1.bitgravity.com");
    publisher.setPort(1935);
    publisher.setDstApplicationName("rtmp");
    publisher.setDstAppInstanceName("push");

    publisher.setDstStreamName("test@test.com/71d73d4cfd1e2f2fed77238021a2cbbbe/test/live/feed01");
    ;

    publisher.setSendFCPublish(true);
    publisher.setSendReleaseStream(true);
    publisher.setDebugLog(true);

    publisher.connect();
}
catch(Exception e)
{
    WMSLoggerFactory.getLogger(null).info(CLASSNAME+"RTMP: ", e);
}
```

Field Summary

protected	akamaiOriginIp
protected	akamaiOriginPort
protected	akamaiPassword
protected	akamaiUserName
protected	bindAddress
public	bytesAvg

public static final	CLASS
public static final	CLASSNAME Value: PushPublishRTMP
protected	connected
protected	connectionFlashVersion
protected	connectionPageURL
protected	connectionQueryStr
protected	connectionResetCount
protected	connectionSwfURL
protected	connectionTimeout
protected	connectLastAttempt
protected	connectLastSuccess
protected	connectMetaData
public static final	CURRENTFLASHVERSION Value: WIN 10,0,12,36
public static final	CURRENTFMLEVERSION Value: FMLE/3.0 (compatible; FMSc/1.0)
protected	debugLogLastPacketCount
protected	debugPackets
protected	doSendOnMetaData
protected	dstAppInstanceName
protected	dstApplicationName
protected	dstStreamName
protected	howToPublish
protected	ioSessionToSessionHolder
protected	isReconnectorRunning

protected	<u>isSSL</u>
protected	<u>lastAudioTC</u>
protected	<u>lastDataTC</u>
protected	<u>lastIdleReadTime</u>
protected	<u>lastIdleWriteTime</u>
protected	<u>lastVideoTC</u>
protected	<u>listeners</u>
protected	<u>lock</u>
protected	<u>netconnectToSessionHolder</u>
public	<u>offlineTime</u>
public	<u>onlineTime</u>
protected	<u>onMetadataToSetDataFrame</u>
public	<u>outprint</u>
protected	<u>packetsSentAudio</u>
protected	<u>packetsSentData</u>
protected	<u>packetsSentVideo</u>
protected	<u>removeDefaultAppInstance</u>
protected	<u>resetOnTimecodeOutOfOrder</u>
protected	<u>rtmpAuthProvider</u>
protected	<u>rtmpDebugLog</u>
protected	<u>secureTokenSharedSecret</u>
protected	<u>sendFCAnnounce</u>
protected	<u>sendFCPublish</u>
protected	<u>sendOnMetadata</u>

protected	sendOriginalTimecodes
protected	sendReleaseStream
protected	sendStreamCloseCommands
protected	sendStreamCloseCommandsSendTime
protected	sessionOrder
protected	srcStreamBufferSize
protected	streamCloseWaitTime
protected	syncPacketType
protected	syncTimecode
protected	timecodesOutOfOrderThreshold
protected	totalPacketsSent
protected	tryConnect
protected	validationFrequency
protected	waitOnMetadataAudio
protected	waitOnMetadataAudioDataRate
protected	waitOnMetadataAvailable
protected	waitOnMetadataReady
protected	waitOnMetadataStartTime
protected	waitOnMetadataTimeout
protected	waitOnMetadataVideo
protected	waitOnMetadataVideoDataRate

Fields inherited from class [com.wowza.wms.pushpublish.model.PushPublishBase](#)

[appInstance](#), [CLASS](#), [CLASSNAME](#), [dstStreamName](#), [hostname](#), [inetAddr](#), [mapFileIndex](#), [mImplementation](#), [mType](#), [originalTimecodeThreshold](#), [password](#), [port](#), [properties](#), [pushPublishStreamDebug](#), [pushPublishType](#), [reconnectWaitTime](#), [sendingPerformance](#), [sendOriginalTimecodes](#), [srcStream](#), [srcStreamName](#), [userName](#)

Fields inherited from interface [com.wowza.wms.pushpublish.model.IPushPublish](#)[PUSHPUBLISH_TYPE_RTMP](#), [PUSHPUBLISH_TYPE_RTP](#), [PUSHPUBLISH_TYPE_UNKNOWN](#)

Constructor Summary

public	PushPublishRTMP()
--------	-----------------------------------

Method Summary

void	addConnectMetaData (String key, AMFData value) Added metadata values to the connect metadata object
void	addListener (IPushPublishRTMPNotify listener) Add IPushPublishRTMPNotify listener
void	addSession (PushPublishRTMPNetConnectionSession sessionHolder) Add session
void	bumpReconnectWaitTime () Bump the reconnect wait time
void	callConnect (PushPublishRTMPNetConnectionSession pushPublisherSession, String addQueryStr) Internal connect
void	connect () Connect to server
AMFDataList	convertDataEvenToAMFData (AMFPacket packet)
AMFPacket	convertOnMetadataToSetDataFrame (AMFPacket packet)
IPushPublishSession	createPushPublishSession ()
void	disconnect () Disconnect from server
String	getAkamaiOriginIp () Get Akamai Origin IP
int	getAkamaiOriginPort () Get Akamai Origin Port
String	getAkamaiPassword () Get Akamai password
String	getAkamaiSessionKey (String sessionId) Get Akamai session key
String	getAkamaiUserName () Get Akamai user name
IApplicationInstance	getAppInstance () Get appInstance interface

String	<u>getBindAddress()</u>
String	<u>getConnectionFlashVersion()</u> Get Flash version used in connection metadata
String	<u>getConnectionFlashVersion()</u> Get Flash version used in connection metadata
String	<u>getConnectionPageURL()</u> Get Page URL used in connection metadata
String	<u>getConnectionQueryStr()</u> Get the connection query string
String	<u>getConnectionSwfURL()</u> Get SWF URL used in connection metadata
int	<u>getConnectionTimeout()</u> Get the connection timeout (milliseconds)
long	<u>getConnectLastAttempt()</u> Get timestamp of last connection attempt
long	<u>getConnectLastSuccess()</u> Get timestamp of last successful connection
String	<u>getContextStr()</u> Get the connection details
String	<u>getDstAppInstanceName()</u> Get destination appInstance name
String	<u>getDstApplicationName()</u> Get destination application name
String	<u>getDstStreamName()</u> Get destination stream name
String	<u>getHostname()</u> Get destination server host name
String	<u>getHowToPublish()</u> Get the howToPublish string (valid values are "live", "record" and "append")
long	<u>getLastAudioTC()</u> Get the absolute timecode of the last sent audio packet (milliseconds)
long	<u>getLastDataTC()</u> Get the absolute timecode of the last sent data packet (milliseconds)
<u>PushPublishRTMPNetConnectionSession</u>	<u>getLastSession()</u> Get the last session interface
long	<u>getLastVideoTC()</u> Get the absolute timecode of the last sent video packet (milliseconds)
java.util.List	<u>getLocalListeners()</u> Get a list of IPushPublishRTMPNotify listeners

Object	<u>getLock()</u> Get the synchronization lock
long	<u>getMessagesAttemptedCount()</u>
long	<u>getMessagesFailedCount()</u>
long	<u>getMessagesRetriedCount()</u>
com.wowza.wms.netconnection.INetConnection	<u>getNetConnection()</u> Get the underlying INetConnection interface for this session
long	<u>getOriginalTimecodeThreshold()</u> Get the original timecode threshold.
long	<u>getPacketsSentAudio()</u>
long	<u>getPacketsSentData()</u>
long	<u>getPacketsSentVideo()</u>
int	<u>getPort()</u> Get destination server port
int	<u>getReconnectDelay()</u> (long currTime) Get the reconnect delay
<u>IPushPublishRTMPAuthProvider</u>	<u>getRTMPAuthProvider()</u> Get RTMP authentication provider
String	<u>getSecureTokenSharedSecret()</u> Get SecureToken shared secret
<u>PushPublishRTMPNetConnectionSession</u>	<u>getSessionHolderByNetConnection()</u> (com.wowza.wms.netconnection.INetConnection netConnection) Get the session holder by INetConnection interface
<u>PushPublishRTMPNetConnectionSession</u>	<u>getSessionHolderBySession()</u> (org.apache.mina.common.IoSession session) Get session holder by IoSession
String	<u>getSessionListDebug()</u> Get the list of active IoSession ids for this publishing session
long	<u>getSrcStreamBufferSize()</u> The amount of audio/video in milliseconds that must be available in the source streams buffer before push publishing will begin.
String	<u>getSrcStreamName()</u> Get source stream name
int	<u>getStreamCloseWaitTime()</u> Get stream close timeout (milliseconds)
int	<u>getSyncPacketType()</u>
long	<u>getSyncTimecode()</u>

int	<u>getTimecodesOutOfOrderThreshold()</u>
long	<u>getTotalPacketsSent()</u> Get total number of packets sent during push publishing session
long	<u>getValidationFrequency()</u> Get connection validation frequency (millisecond).
long	<u>getWaitOnMetadataTimeout()</u> Timeout (milliseconds) that the wait on onMetadata system will wait looking for video and audio data to be populated.
org.apache.mina.common. IoSession	<u>internalConnect(PushPublishRTMPNetConnectionSession pushPublisherSession)</u> Internal connect method
Boolean	<u>isConnected()</u>
boolean	<u>isCurrentSession(org.apache.mina.common.IoSession session)</u> Test to see if passed in session is current session
boolean	<u>isDebugEnabled()</u> Is debug logging turned on
boolean	<u>isDebugPackets()</u> Set to true to log every packet being sent
static boolean	<u>isFlashVersionFMLE(String flashVersion)</u> Returns true if the Flash player version string is a Flash Media Live Encoder version (start with FMLE/ or FME/)
static boolean	<u>isFlashVersionFMLE(String flashVersion)</u> Returns true if the Flash player version string is a Flash Media Live Encoder version (start with FMLE/ or FME/)
boolean	<u>isOnMetadataToSetDataFrame()</u> If true all onMetadata events are converted to @setDataFrame calls
boolean	<u>isRemoveDefaultAppInstance()</u> If true and destination application instance is _definst_ then the application instance name will not be sent to the remove server as part of the connection information.
boolean	<u>isResetOnTimecodeOutOfOrder()</u> Set to true if you wish push to reset if timecodes jump out of order.
boolean	<u>isSendFCAnnounce()</u> If true call FCAnnounce(streamName) after connecting to server
boolean	<u>isSendFCPublish()</u> If true call FCPublish(streamName) after connecting to server
boolean	<u>isSendOnMetadata()</u> True if sending onMetadata event at start of stream
boolean	<u>isSendOriginalTimecodes()</u> true if sending the original timecodes of source stream.
boolean	<u>isSendReleaseStream()</u> If true call releaseStream(streamName) after connecting to server

boolean	<u>isSendStreamCloseCommands()</u> If true will send FCUnpublish, closeStream, deleteStream on disconnect
boolean	<u>isSSL()</u>
boolean	<u>isWaitOnMetadataAudio()</u> If true and waitOnMetadataAvailable, publishing will not start until onMetadata packet is populated with audio data.
boolean	<u>isWaitOnMetadataAudioDatarate()</u>
boolean	<u>isWaitOnMetadataAvailable()</u> Set to true to turn on system that will monitor onMetadata packet and wait for it to be populated before it starts sending the stream.
boolean	<u>isWaitOnMetadataVideo()</u> If true and waitOnMetadataAvailable, publishing will not start until onMetadata packet is populated with video data.
boolean	<u>isWaitOnMetadataVideoDatarate()</u>
void	<u>notifyAkamaiClientLogin(PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)</u> Notify Akamai ClientLogin
void	<u>notifyAkamaiSetChallenge(PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)</u> Notify Akamai SetChallenge
void	<u>notifyAkamaiSetOriginConnectionInfo(PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)</u> Notify Akamai SetOriginConnectionInfo
void	<u>notifyConnect(PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)</u> Notify connect
void	<u>notifyConnectFailure(PushPublishRTMPNetConnectionSession pushPublisherSession)</u> Notify connect failure
void	<u>notifyConnectStart(PushPublishRTMPNetConnectionSession pushPublisherSession)</u> Notify connect start
void	<u>notifyConnectSuccess(PushPublishRTMPNetConnectionSession pushPublisherSession)</u> Notify connect success
void	<u>notifyFCAnnounce(PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)</u> Notify FCAnnounce
void	<u>notifyFCPublish(PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)</u> Notify FCPublish

void	<u>notifyHandshakeResult</u> (<u>PushPublishRTMPNetConnectionSession</u> pushPublisherSession, <u>com.wowza.wms.request.RequestFunction</u> function, <u>AMFDataList</u> params) Notify handshake result
void	<u>notifyPublishHandlerPlay</u> (<u>PushPublishRTMPNetConnectionSession</u> pushPublisherSession, <u>java.io.OutputStream</u> out, <u>long[]</u> playSizes) Notify publish handler play
void	<u>notifyPushPublisherSessionCreate</u> (<u>PushPublishRTMPNetConnectionSession</u> pushPublisherSession) Notify session create
void	<u>notifyPushPublisherSessionDestroy</u> (<u>PushPublishRTMPNetConnectionSession</u> pushPublisherSession) Notify session destroy
void	<u>notifySessionClosed</u> (<u>PushPublishRTMPNetConnectionSession</u> pushPublisherSession) Notify session closed
void	<u>notifySessionIdle</u> (<u>PushPublishRTMPNetConnectionSession</u> pushPublisherSession) Notify session idle event
void	<u>notifySessionOpened</u> (<u>PushPublishRTMPNetConnectionSession</u> pushPublisherSession) Notify session opened
void	<u>notifyStreamCreate</u> (<u>PushPublishRTMPNetConnectionSession</u> pushPublisherSession, <u>com.wowza.wms.request.RequestFunction</u> function, <u>AMFDataList</u> params) Notify stream create
void	<u>notifyStreamOnPlayStatus</u> (<u>PushPublishRTMPNetConnectionSession</u> pushPublisherSession, <u>com.wowza.wms.request.RequestFunction</u> function, <u>AMFDataList</u> params) Notify OnPlayStatus
void	<u>notifyStreamOnStatus</u> (<u>PushPublishRTMPNetConnectionSession</u> pushPublisherSession, <u>com.wowza.wms.request.RequestFunction</u> function, <u>AMFDataList</u> params) Notify StreamOnStatus
void	<u>notifyValidateSession</u> (<u>PushPublishRTMPNetConnectionSession</u> pushPublisherSession)
void	<u>notifyValidateSessionResult</u> (<u>PushPublishRTMPNetConnectionSession</u> pushPublisherSession, <u>boolean</u> result)
boolean	<u>removeListener</u> (<u>IPushPublishRTMPNotify</u> listener) remove IPushPublishRTMPNotify listener
void	<u>removeSession</u> (<u>PushPublishRTMPNetConnectionSession</u> sessionHolder) Remove session
void	<u>resetConnection</u> () Reset server connection
void	<u>resetConnection</u> (<u>boolean</u> noDelay) Reset server connection

String	<u>resolveHostName</u> () Resolve host name to connect to
void	<u>sessionClosed</u> (org.apache.mina.common.IoSession session) Internal callback
void	<u>sessionOpened</u> (org.apache.mina.common.IoSession session) Internal callback
void	<u>setAkamaiOriginIp</u> (String akamaiOriginIp) Set Akamai Origin IP
void	<u>setAkamaiOriginPort</u> (int akamaiOriginPort) Set Akamai Origin Port
void	<u>setAkamaiPassword</u> (String akamaiPassword) Set Akamai password
void	<u>setAkamaiUserName</u> (String akamaiUserName) Set Akamai user name
void	<u>setAppInstance</u> (IApplicationInstance appInstance) Set appInstance interface
void	<u>setBindAddress</u> (String bindAddress)
void	<u>setConnectionFlashVerion</u> (String connectionFlashVersion) Set Flash version used in connection metadata
void	<u>setConnectionFlashVersion</u> (String connectionFlashVersion) Set Flash version used in connection metadata
void	<u>setConnectionPageURL</u> (String connectionPageURL) Set Page URL used in connection metadata
void	<u>setConnectionQueryStr</u> (String connectionQueryStr) Set the connection query string
void	<u>setConnectionSwfURL</u> (String connectionSwfURL) Set SWF URL used in connection metadata
void	<u>setConnectionTimeout</u> (int connectionTimeout) Set the connection timeout (milliseconds)
void	<u>setConnectLastAttempt</u> (long connectLastAttempt) Set timestamp of last connection attempt
void	<u>setConnectLastSuccess</u> (long connectLastSuccess) Set timestamp of last successful connection
void	<u>setDebugLog</u> (boolean debugLog) Set debug logging
void	<u>setDebugPackets</u> (boolean debugPackets) Set to true to log every packet being sent
void	<u>setDstAppInstanceName</u> (String dstAppInstanceName) Set destination appInstance name

void	<u>setDstApplicationName</u> (String dstApplicationName) Set destination application name
void	<u>setDstStreamName</u> (String dstStreamName) Set destination stream name
void	<u>setHostname</u> (String hostname) Set destination server host name
void	<u>setHowToPublish</u> (String howToPublish) Set the howToPublish string (valid values are "live", "record" and "append")
void	<u>setOnMetadataToSetDataFrame</u> (boolean onMetadataToSetDataFrame) If true all onMetadata events are converted to @setDataFrame calls
void	<u>setOriginalTimecodeThreshold</u> (long originalTimecodeThreshold) Set the original timecode threshold.
void	<u>setPort</u> (int port) Set destination server port
void	<u>setRemoveDefaultAppInstance</u> (boolean removeDefaultAppInstance) If true and destination application instance is _definst_ then the application instance name will not be sent to the remove server as part of the connection information.
void	<u>setResetOnTimecodeOutOfOrder</u> (boolean resetOnTimecodeOutOfOrder) Set to true if you wish push to reset if timecodes jump out of order.
void	<u>setRTMPAuthProvider</u> (<u>IPushPublishRTMPAuthProvider</u> rtmpAuthProvider) Set RTMP authentication provider
void	<u>setSecureTokenSharedSecret</u> (String secureTokenSharedSecret) Set SecureToken shared secret
void	<u>setSendFCAnnounce</u> (boolean sendFCAnnounce) If true call FCAnnounce(streamName) after connecting to server
void	<u>setSendFCPublish</u> (boolean sendFCPublish) If true call FCPublish(streamName) after connecting to server
void	<u>setSendOnMetadata</u> (boolean sendOnMetadata) True if sending onMetadata event at start of stream
void	<u>setSendOriginalTimecodes</u> (boolean sendOriginalTimecodes) true if sending the original timecodes of source stream.
void	<u>setSendReleaseStream</u> (boolean sendReleaseStream) If true call releaseStream(streamName) after connecting to server
void	<u>setSendStreamCloseCommands</u> (boolean sendStreamCloseCommands) If true will send FCUnpublish, closeStream, deleteStream on disconnect
void	<u>setSrcStreamBufferSize</u> (long srcStreamBufferSize) The amount of audio/video in milliseconds that must be available in the source streams buffer before push publishing will begin.
void	<u>setSrcStreamName</u> (String srcStreamName) Set source stream name

void	<u>setSSL</u> (boolean isSSL)
void	<u>setStreamCloseWaitTime</u> (int streamCloseWaitTime) Set stream close timeout (milliseconds)
void	<u>setSyncPacketType</u> (int syncPacketType)
void	<u>setSyncTimecode</u> (long syncTimecode)
void	<u>setTimecodesOutOfOrderThreshold</u> (int timecodesOutOfOrderThreshold)
void	<u>setValidationFrequency</u> (long validationFrequency) Set connection validation frequency (millisecond).
void	<u>setWaitOnMetadataAudio</u> (boolean waitOnMetadataAudio) If true and waitOnMetadataAvailable, publishing will not start until onMetadata packet is populated with audio data.
void	<u>setWaitOnMetadataAudioDatarate</u> (boolean waitOnMetadataAudioDatarate)
void	<u>setWaitOnMetadataAvailable</u> (boolean waitOnMetadataAvailable) Set to true to turn on system that will monitor onMetadata packet and wait for it to be populated before it starts sending the stream.
void	<u>setWaitOnMetadataTimeout</u> (long waitOnMetadataTimeout) Timeout (milliseconds) that the wait on onMetadata system will wait looking for video and audio data to be populated.
void	<u>setWaitOnMetadataVideo</u> (boolean waitOnMetadataVideo) If true and waitOnMetadataAvailable, publishing will not start until onMetadata packet is populated with video data.
void	<u>setWaitOnMetadataVideoDatarate</u> (boolean waitOnMetadataVideoDatarate)
void	<u>tryToConnect</u> () Try to connect to server

Methods inherited from class [com.wowza.wms.pushpublish.model.PushPublishBase](#)

[createPushPublishSession](#), [getAppInstance](#), [getBandwidth](#), [getConfigDataOrDefault](#), [getConfigDataOrDefault](#), [getConfigDataOrDefault](#), [getContextStr](#), [getDstStreamName](#), [getHostname](#), [getImplementation](#), [getMapFileIndex](#), [getNumberFromString](#), [getOriginalTimecodeThreshold](#), [getPassword](#), [getPort](#), [getProperties](#), [getPushPublishType](#), [getPushPublishVersionStr](#), [getReconnectWaitTime](#), [getSrcStream](#), [getSrcStreamName](#), [getUserName](#), [init](#), [isAdaptiveStreaming](#), [isSendOriginalTimecodes](#), [load](#), [setAdaptiveStreaming](#), [setAppInstance](#), [setDebugLog](#), [setDstStreamName](#), [setHost](#), [setHostname](#), [setImplementation](#), [setMapFileIndex](#), [setOriginalTimecodeThreshold](#), [setOriginalTimecodeThreshold](#), [setPassword](#), [setPort](#), [setPort](#), [setReconnectWaitTime](#), [setSendOriginalTimecodes](#), [setSrcStream](#), [setSrcStreamName](#), [setStreamName](#), [setUserName](#)

Methods inherited from class [java.lang.Object](#)

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

Methods inherited from interface [com.wowza.wms.pushpublish.model.IPushPublish](#)

[connect](#), [createPushPublishSession](#), [disconnect](#), [getContextStr](#), [getDstStreamName](#), [getHostname](#), [getMapFileIndex](#), [getMessagesAttemptedCount](#), [getMessagesFailedCount](#), [getMessagesRetriedCount](#), [getPort](#), [getProperties](#), [getPushPublishType](#), [getReconnectWaitTime](#), [getSrcStream](#), [getSrcStreamName](#), [init](#), [isConnected](#), [resetConnection](#), [setDebugLog](#), [setDstStreamName](#), [setHost](#), [setHostname](#), [setMapFileIndex](#), [setPort](#), [setPort](#), [setReconnectWaitTime](#), [setSrcStream](#), [setSrcStreamName](#), [setStreamName](#)

Fields

CLASS

public static final java.lang.Class **CLASS**

CLASSNAME

public static final java.lang.String **CLASSNAME**

Constant value: **PushPublishRTMP**

CURRENTFMLEVERSION

public static final java.lang.String **CURRENTFMLEVERSION**

Constant value: **FMLE/3.0 (compatible; FMSc/1.0)**

CURRENTFLASHVERSION

public static final java.lang.String **CURRENTFLASHVERSION**

Constant value: **WIN 10,0,12,36**

rtmpDebugLog

protected boolean **rtmpDebugLog**

debugLogLastPacketCount

protected long **debugLogLastPacketCount**

debugPackets

protected boolean **debugPackets**

(continued from last page)

netconnectToSessionHolder

protected java.util.Map **netconnectToSessionHolder**

ioSessionToSessionHolder

protected java.util.Map **ioSessionToSessionHolder**

sessionOrder

protected java.util.List **sessionOrder**

listeners

protected java.util.List **listeners**

dstStreamName

protected java.lang.String **dstStreamName**

dstApplicationName

protected java.lang.String **dstApplicationName**

dstAppInstanceName

protected java.lang.String **dstAppInstanceName**

lock

protected java.lang.Object **lock**

howToPublish

protected java.lang.String **howToPublish**

sendFCPublish

protected boolean **sendFCPublish**

(continued from last page)

sendFCAnnounce

protected boolean **sendFCAnnounce**

sendReleaseStream

protected boolean **sendReleaseStream**

sendStreamCloseCommands

protected boolean **sendStreamCloseCommands**

sendOnMetadata

protected boolean **sendOnMetadata**

onMetadataToSetDataFrame

protected boolean **onMetadataToSetDataFrame**

sendOriginalTimecodes

protected boolean **sendOriginalTimecodes**

timecodesOutOfOrderThreshold

protected int **timecodesOutOfOrderThreshold**

tryConnect

protected boolean **tryConnect**

isReconnectorRunning

protected boolean **isReconnectorRunning**

(continued from last page)

connectionQueryStr

protected java.lang.String **connectionQueryStr**

connectionFlashVersion

protected java.lang.String **connectionFlashVersion**

connectionSwfURL

protected java.lang.String **connectionSwfURL**

connectionPageURL

protected java.lang.String **connectionPageURL**

secureTokenSharedSecret

protected java.lang.String **secureTokenSharedSecret**

akamaiUserName

protected java.lang.String **akamaiUserName**

akamaiPassword

protected java.lang.String **akamaiPassword**

akamaiOriginIp

protected java.lang.String **akamaiOriginIp**

akamaiOriginPort

protected java.lang.Integer **akamaiOriginPort**

validationFrequency

protected long **validationFrequency**

(continued from last page)

connectionTimeout

protected int **connectionTimeout**

connectLastAttempt

protected long **connectLastAttempt**

connectLastSuccess

protected long **connectLastSuccess**

connectionResetCount

protected long **connectionResetCount**

connected

protected boolean **connected**

doSendOnMetaData

protected boolean **doSendOnMetaData**

removeDefaultAppInstance

protected boolean **removeDefaultAppInstance**

resetOnTimecodeOutOfOrder

protected boolean **resetOnTimecodeOutOfOrder**

sendStreamCloseCommandsSendTime

protected long **sendStreamCloseCommandsSendTime**

(continued from last page)

streamCloseWaitTimeprotected int **streamCloseWaitTime**

lastIdleReadTimeprotected long **lastIdleReadTime**

lastIdleWriteTimeprotected long **lastIdleWriteTime**

totalPacketsSentprotected long **totalPacketsSent**

lastVideoTCprotected long **lastVideoTC**

lastAudioTCprotected long **lastAudioTC**

lastDataTCprotected long **lastDataTC**

packetsSentAudioprotected long **packetsSentAudio**

packetsSentVideoprotected long **packetsSentVideo**

packetsSentDataprotected long **packetsSentData**

(continued from last page)

waitOnMetadataAvailable

protected boolean **waitOnMetadataAvailable**

waitOnMetadataVideo

protected boolean **waitOnMetadataVideo**

waitOnMetadataVideoDatarate

protected boolean **waitOnMetadataVideoDatarate**

waitOnMetadataAudio

protected boolean **waitOnMetadataAudio**

waitOnMetadataAudioDatarate

protected boolean **waitOnMetadataAudioDatarate**

waitOnMetadataTimeout

protected long **waitOnMetadataTimeout**

waitOnMetadataReady

protected boolean **waitOnMetadataReady**

waitOnMetadataStarttime

protected long **waitOnMetadataStarttime**

srcStreamBufferSize

protected long **srcStreamBufferSize**

(continued from last page)

bindAddress

protected java.lang.String **bindAddress**

rtmpAuthProvider

protected com.wowza.wms.pushpublish.protocol.rtmp.IPushPublishRTMPAuthProvider
rtmpAuthProvider

connectMetaData

protected java.util.Map **connectMetaData**

offlineTime

public long **offlineTime**

onlineTime

public long **onlineTime**

outprint

public int **outprint**

bytesAvg

public float **bytesAvg**

isSSL

protected boolean **isSSL**

syncTimecode

protected long **syncTimecode**

syncPacketType

protected int **syncPacketType**

(continued from last page)

Constructors

PushPublishRTMP

```
public PushPublishRTMP()
```

Methods

getNetConnection

```
public com.wowza.wms.netconnection.INetConnection getNetConnection()
```

Get the underlying INetConnection interface for this session

Returns:

INetConnection interface for this session

getLock

```
public Object getLock()
```

Get the synchronization lock

Returns:

synchronization lock

getConnectionTimeout

```
public int getConnectionTimeout()
```

Get the connection timeout (milliseconds)

Returns:

connection timeout (milliseconds)

setConnectionTimeout

```
public void setConnectionTimeout(int connectionTimeout)
```

Set the connection timeout (milliseconds)

Parameters:

connectionTimeout - connection timeout (milliseconds)

getHowToPublish

```
public String getHowToPublish()
```

Get the howToPublish string (valid values are "live", "record" and "append")

Returns:

howToPublish string

setHowToPublish

```
public void setHowToPublish(String howToPublish)
```

Set the howToPublish string (valid values are "live", "record" and "append")

Parameters:

howToPublish - howToPublish string

isSendFCPublish

```
public boolean isSendFCPublish( )
```

If true call FCPublish(streamName) after connecting to server

Returns:

true if calling FCPublish(streamName) after connecting to server

setSendFCPublish

```
public void setSendFCPublish(boolean sendFCPublish)
```

If true call FCPublish(streamName) after connecting to server

Parameters:

sendFCPublish - true if calling FCPublish(streamName) after connecting to server

isSendReleaseStream

```
public boolean isSendReleaseStream( )
```

If true call releaseStream(streamName) after connecting to server

Returns:

true if calling releaseStream(streamName) after connecting to server

setSendReleaseStream

```
public void setSendReleaseStream(boolean sendReleaseStream)
```

If true call releaseStream(streamName) after connecting to server

Parameters:

sendReleaseStream - true if calling releaseStream(streamName) after connecting to server

getSessionListDebug

```
public String getSessionListDebug( )
```

Get the list of active IoSession ids for this publishing session

Returns:

list of active IoSession ids for this publishing session

isCurrentSession

```
public boolean isCurrentSession(org.apache.mina.common.IoSession session)
```

(continued from last page)

Test to see if passed in session is current session

Parameters:

session - IoSession interface

Returns:

true if current session

getLastSession

```
public PushPublishRTMPNetConnectionSession getLastSession()
```

Get the last session interface

Returns:

last session interface

addSession

```
public void addSession(PushPublishRTMPNetConnectionSession sessionHolder)
```

Add session

Parameters:

sessionHolder - session holder

removeSession

```
public void removeSession(PushPublishRTMPNetConnectionSession sessionHolder)
```

Remove session

Parameters:

sessionHolder - session holder

getSessionHolderBySession

```
public PushPublishRTMPNetConnectionSession  
getSessionHolderBySession(org.apache.mina.common.IoSession session)
```

Get session holder by IoSession

Parameters:

session - IoSession

Returns:

session holder

getSessionHolderByNetConnection

```
public PushPublishRTMPNetConnectionSession  
getSessionHolderByNetConnection(com.wowza.wms.netconnection.INetConnection  
netConnection)
```

Get the session holder by INetConnection interface

Parameters:

netConnection - INetConnection interface

(continued from last page)

Returns:

session holder

sessionOpened

```
public void sessionOpened(org.apache.mina.common.Session session)
```

Internal callback

Parameters:

session - Session

sessionClosed

```
public void sessionClosed(org.apache.mina.common.Session session)
```

Internal callback

Parameters:

session - Session

tryToConnect

```
protected void tryToConnect()
```

Try to connect to server

resetConnection

```
public void resetConnection()
```

Reset server connection

resetConnection

```
public void resetConnection(boolean noDelay)
```

Reset server connection

Parameters:

noDelay - if true reset immediately

getContextStr

```
public String getContextStr()
```

Get the connection details

Returns:

connection details

resolveHostName

```
protected String resolveHostName()
```

Resolve host name to connect to

(continued from last page)

Returns:

hostname

internalConnect

protected org.apache.mina.common.Session

internalConnect([PushPublishRTMPNetConnectionSession](#) pushPublisherSession)

Internal connect method

Parameters:

pushPublisherSession - session holder

Returns:

Io session

getConnectionQueryStr

public String **getConnectionQueryStr**()

Get the connection query string

Returns:

connection query string

setConnectionQueryStr

public void **setConnectionQueryStr**(String connectionQueryStr)

Set the connection query string

Parameters:

connectionQueryStr - connection query string

connect

public void **connect**()

Connect to server

callConnect

protected void **callConnect**([PushPublishRTMPNetConnectionSession](#) pushPublisherSession, String addQueryStr)

Internal connect

Parameters:

pushPublisherSession - session holder

addQueryStr - additional query params

getAkamaiSessionKey

public String **getAkamaiSessionKey**(String sessionId)

Get Akamai session key

Parameters:

sessionId - session Id

(continued from last page)

Returns:

Akamai session key

getConnectionFlashVersion

```
public String getConnectionFlashVersion()
```

Get Flash version used in connection metadata

Returns:

Flash version used in connection metadata

getConnectionFlashVersion

```
public String getConnectionFlashVersion()
```

Get Flash version used in connection metadata

Returns:

Flash version used in connection metadata

setConnectionFlashVersion

```
public void setConnectionFlashVersion(String connectionFlashVersion)
```

Set Flash version used in connection metadata

Parameters:

connectionFlashVersion - Flash version used in connection metadata

setConnectionFlashVersion

```
public void setConnectionFlashVersion(String connectionFlashVersion)
```

Set Flash version used in connection metadata

Parameters:

connectionFlashVersion - Flash version used in connection metadata

getConnectionSwfURL

```
public String getConnectionSwfURL()
```

Get SWF URL used in connection metadata

Returns:

SWF URL used in connection metadata

setConnectionSwfURL

```
public void setConnectionSwfURL(String connectionSwfURL)
```

Set SWF URL used in connection metadata

Parameters:

connectionSwfURL - SWF URL used in connection metadata

getConnectionPageURL

```
public String getConnectionPageURL()
```

Get Page URL used in connection metadata

Returns:

Page URL used in connection metadata

setConnectionPageURL

```
public void setConnectionPageURL(String connectionPageURL)
```

Set Page URL used in connection metadata

Parameters:

connectionPageURL - Page URL used in connection metadata

getSecureTokenSharedSecret

```
public String getSecureTokenSharedSecret()
```

Get SecureToken shared secret

Returns:

SecureToken shared secret

setSecureTokenSharedSecret

```
public void setSecureTokenSharedSecret(String secureTokenSharedSecret)
```

Set SecureToken shared secret

Parameters:

secureTokenSharedSecret - SecureToken shared secret

getAkamaiUserName

```
public String getAkamaiUserName()
```

Get Akamai user name

Returns:

Akamai user name

setAkamaiUserName

```
public void setAkamaiUserName(String akamaiUserName)
```

Set Akamai user name

Parameters:

akamaiUserName - Akamai user name

getAkamaiPassword

```
public String getAkamaiPassword()
```

(continued from last page)

Get Akamai password

Returns:

Akamai password

setAkamaiPassword

```
public void setAkamaiPassword(String akamaiPassword)
```

Set Akamai password

Parameters:

akamaiPassword - Akamai password

getAkamaiOriginIp

```
public String getAkamaiOriginIp()
```

Get Akamai Origin IP

Returns:

Akamai Origin IP

setAkamaiOriginIp

```
public void setAkamaiOriginIp(String akamaiOriginIp)
```

Set Akamai Origin IP

Parameters:

akamaiOriginIp - Akamai Origin IP

getAkamaiOriginPort

```
public int getAkamaiOriginPort()
```

Get Akamai Origin Port

Returns:

Akamai Origin Port

setAkamaiOriginPort

```
public void setAkamaiOriginPort(int akamaiOriginPort)
```

Set Akamai Origin Port

Parameters:

akamaiOriginPort - Akamai Origin Port

isSendFCAnnounce

```
public boolean isSendFCAnnounce()
```

If true call FCAnnounce(streamName) after connecting to server

Returns:

true if calling FCAnnounce(streamName) after connecting to server

setSendFCAnnounce

```
public void setSendFCAnnounce(boolean sendFCAnnounce)
```

If true call FCAnnounce(streamName) after connecting to server

Parameters:

sendFCAnnounce - true if calling FCAnnounce(streamName) after connecting to server

getValidationFrequency

```
public long getValidationFrequency()
```

Get connection validation frequency (millisecond). Zero to turn off validation.

Returns:

connection validation frequency (millisecond)

setValidationFrequency

```
public void setValidationFrequency(long validationFrequency)
```

Set connection validation frequency (millisecond). Zero to turn off validation.

Parameters:

validationFrequency - connection validation frequency (millisecond)

addListener

```
public void addListener(IPushPublishRTMPNotify listener)
```

Add IPushPublishRTMPNotify listener

Parameters:

listener - IPushPublishRTMPNotify listener

removeListener

```
public boolean removeListener(IPushPublishRTMPNotify listener)
```

remove IPushPublishRTMPNotify listener

Parameters:

listener - IPushPublishRTMPNotify listener

Returns:

true if removed

getLocalListeners

```
protected java.util.List getLocalListeners()
```

Get a list of IPushPublishRTMPNotify listeners

Returns:

list of IPushPublishRTMPNotify listeners

notifyConnectStart

protected void **notifyConnectStart**([PushPublishRTMPNetConnectionSession](#)
pushPublisherSession)

Notify connect start

Parameters:

pushPublisherSession - session holder

notifyConnectSuccess

protected void **notifyConnectSuccess**([PushPublishRTMPNetConnectionSession](#)
pushPublisherSession)

Notify connect success

Parameters:

pushPublisherSession - session holder

notifyConnectFailure

protected void **notifyConnectFailure**([PushPublishRTMPNetConnectionSession](#)
pushPublisherSession)

Notify connect failure

Parameters:

pushPublisherSession - session holder

notifyHandshakeResult

protected void **notifyHandshakeResult**([PushPublishRTMPNetConnectionSession](#)
pushPublisherSession,
com.wowza.wms.request.RequestFunction function,
[AMFDataList](#) params)

Notify handshake result

Parameters:

pushPublisherSession - session holder

function - function

params - params

notifyAkamaiSetChallenge

protected void **notifyAkamaiSetChallenge**([PushPublishRTMPNetConnectionSession](#)
pushPublisherSession,
com.wowza.wms.request.RequestFunction function,
[AMFDataList](#) params)

Notify Akamai SetChallenge

Parameters:

pushPublisherSession - session holder

function - function

params - params

(continued from last page)

notifyAkamaiSetOriginConnectionInfo

```
protected void notifyAkamaiSetOriginConnectionInfo(PushPublishRTMPNetConnectionSession  
pushPublisherSession,  
com.wowza.wms.request.RequestFunction function,  
AMFDataList params)
```

Notify Akamai SetOriginConnectionInfo

Parameters:

pushPublisherSession - session holder
function - function
params - params

notifyAkamaiClientLogin

```
protected void notifyAkamaiClientLogin(PushPublishRTMPNetConnectionSession  
pushPublisherSession,  
com.wowza.wms.request.RequestFunction function,  
AMFDataList params)
```

Notify Akamai ClientLogin

Parameters:

pushPublisherSession - session holder
function - function
params - params

notifyFCPublish

```
protected void notifyFCPublish(PushPublishRTMPNetConnectionSession  
pushPublisherSession,  
com.wowza.wms.request.RequestFunction function,  
AMFDataList params)
```

Notify FCPublish

Parameters:

pushPublisherSession - session holder
function - function
params - params

notifyFCAnnounce

```
protected void notifyFCAnnounce(PushPublishRTMPNetConnectionSession  
pushPublisherSession,  
com.wowza.wms.request.RequestFunction function,  
AMFDataList params)
```

Notify FCAnnounce

Parameters:

pushPublisherSession - session holder
function - function
params - params

(continued from last page)

notifyConnect

```
protected void notifyConnect(PushPublishRTMPNetConnectionSession pushPublisherSession,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Notify connect

Parameters:

pushPublisherSession - session holder
function - function
params - params

notifyStreamCreate

```
protected void notifyStreamCreate(PushPublishRTMPNetConnectionSession  
pushPublisherSession,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Notify stream create

Parameters:

pushPublisherSession - session holder
function - function
params - params

notifyStreamOnPlayStatus

```
protected void notifyStreamOnPlayStatus(PushPublishRTMPNetConnectionSession  
pushPublisherSession,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Notify OnPlayStatus

Parameters:

pushPublisherSession - session holder
function - function
params - params

notifyStreamOnStatus

```
protected void notifyStreamOnStatus(PushPublishRTMPNetConnectionSession  
pushPublisherSession,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Notify StreamOnStatus

Parameters:

pushPublisherSession - session holder
function - function
params - params

(continued from last page)

notifyPublishHandlerPlay

```
protected void notifyPublishHandlerPlay(PushPublishRTMPNetConnectionSession  
pushPublisherSession,  
    java.io.OutputStream out,  
    long[] playSizes)
```

Notify publish handler play

Parameters:

pushPublisherSession - session holder
out - output stream
playSizes - play sizes

notifySessionIdle

```
protected void notifySessionIdle(PushPublishRTMPNetConnectionSession  
pushPublisherSession)
```

Notify session idle event

Parameters:

pushPublisherSession - session holder

notifySessionOpened

```
protected void notifySessionOpened(PushPublishRTMPNetConnectionSession  
pushPublisherSession)
```

Notify session opened

Parameters:

pushPublisherSession - session holder

notifySessionClosed

```
protected void notifySessionClosed(PushPublishRTMPNetConnectionSession  
pushPublisherSession)
```

Notify session closed

Parameters:

pushPublisherSession - session holder

notifyValidateSession

```
protected void notifyValidateSession(PushPublishRTMPNetConnectionSession  
pushPublisherSession)
```

notifyValidateSessionResult

```
protected void notifyValidateSessionResult(PushPublishRTMPNetConnectionSession  
pushPublisherSession,  
    boolean result)
```

(continued from last page)

notifyPushPublisherSessionCreate

```
protected void notifyPushPublisherSessionCreate(PushPublishRTMPNetConnectionSession  
pushPublisherSession)
```

Notify session create

Parameters:

pushPublisherSession - session holder

notifyPushPublisherSessionDestroy

```
protected void notifyPushPublisherSessionDestroy(PushPublishRTMPNetConnectionSession  
pushPublisherSession)
```

Notify session destroy

Parameters:

pushPublisherSession - session holder

getConnectLastAttempt

```
public long getConnectLastAttempt()
```

Get timestamp of last connection attempt

Returns:

timestamp of last connection attempt

setConnectLastAttempt

```
public void setConnectLastAttempt(long connectLastAttempt)
```

Set timestamp of last connection attempt

Parameters:

connectLastAttempt - timestamp of last connection attempt

bumpReconnectWaitTime

```
protected void bumpReconnectWaitTime()
```

Bump the reconnect wait time

getReconnectDelay

```
protected int getReconnectDelay(long currTime)
```

Get the reconnect delay

Parameters:

currTime - current timestamp

Returns:

delay in milliseconds

(continued from last page)

getConnectLastSuccess

```
public long getConnectLastSuccess()
```

Get timestamp of last successful connection

Returns:

timestamp of last successful connection

setConnectLastSuccess

```
public void setConnectLastSuccess(long connectLastSuccess)
```

Set timestamp of last successful connection

Parameters:

connectLastSuccess - timestamp of last successful connection

isSendOriginalTimecodes

```
public boolean isSendOriginalTimecodes()
```

true if sending the original timecodes of source stream. false if starting at zero.

Returns:

true if sending the original timecodes of source stream

setSendOriginalTimecodes

```
public void setSendOriginalTimecodes(boolean sendOriginalTimecodes)
```

true if sending the original timecodes of source stream. false if starting at zero.

Parameters:

sendOriginalTimecodes - true if sending the original timecodes of source stream

getOriginalTimecodeThreshold

```
public long getOriginalTimecodeThreshold()
```

Get the original timecode threshold. This value will attempt to reduce the magnitude of absolute timecodes while keeping multiple-bitrate alignment. It is probably best to set this value quite high to a value such as 0x100000. The value is in milliseconds. The following formula is used to calculate a timecode offset $\text{timecodeOffset} = \text{timecode} - (\text{timecode} \% \text{originalTimecodeThreshold})$;

Returns:

original timecode threshold

setOriginalTimecodeThreshold

```
public void setOriginalTimecodeThreshold(long originalTimecodeThreshold)
```

Set the original timecode threshold. This value will attempt to reduce the magnitude of absolute timecodes while keeping multiple-bitrate alignment. It is probably best to set this value quite high to a value such as 0x100000. The value is in milliseconds. The following formula is used to calculate a timecode offset $\text{timecodeOffset} = \text{timecode} - (\text{timecode} \% \text{originalTimecodeThreshold})$;

Parameters:

originalTimecodeThreshold - original timecode threshold

isDebugLog

```
public boolean isDebugLog()
```

Is debug logging turned on

Returns:

true if debug logging is turned on

setDebugLog

```
public void setDebugLog(boolean debugLog)
```

Set debug logging

Parameters:

debugLog - debug logging

getAppInstance

```
public IApplicationInstance getAppInstance()
```

Get appInstance interface

Returns:

appInstance interface

setAppInstance

```
public void setAppInstance(IApplicationInstance appInstance)
```

Set appInstance interface

Parameters:

appInstance - appInstance interface

getSrcStreamName

```
public String getSrcStreamName()
```

Get source stream name

Returns:

source stream name

setSrcStreamName

```
public void setSrcStreamName(String srcStreamName)
```

Set source stream name

Parameters:

srcStreamName - source stream name

getDstStreamName

```
public String getDstStreamName()
```

(continued from last page)

Get destination stream name

Returns:

destination stream name

setDstStreamName

```
public void setDstStreamName(String dstStreamName)
```

Set destination stream name

Parameters:

dstStreamName - destination stream name

getHostname

```
public String getHostname()
```

Get destination server host name

Returns:

destination server host name

setHostname

```
public void setHostname(String hostname)
```

Set destination server host name

Parameters:

hostname - destination server host name

getPort

```
public int getPort()
```

Get destination server port

Returns:

destination server port

setPort

```
public void setPort(int port)
```

Set destination server port

Parameters:

port - destination server port

getDstApplicationName

```
public String getDstApplicationName()
```

Get destination application name

Returns:

destination application name

setDstApplicationName

```
public void setDstApplicationName(String dstApplicationName)
```

Set destination application name

Parameters:

dstApplicationName - destination application name

getDstAppInstanceName

```
public String getDstAppInstanceName()
```

Get destination appInstance name

Returns:

destination appInstance name

setDstAppInstanceName

```
public void setDstAppInstanceName(String dstAppInstanceName)
```

Set destination appInstance name

Parameters:

dstAppInstanceName - destination appInstance name

getRTMPAuthProvider

```
public IPushPublishRTMPAuthProvider getRTMPAuthProvider()
```

Get RTMP authentication provider

Returns:

RTMP authentication provider

setRTMPAuthProvider

```
public void setRTMPAuthProvider(IPushPublishRTMPAuthProvider rtmpAuthProvider)
```

Set RTMP authentication provider

Parameters:

rtmpAuthProvider - RTMP authentication provider

isSendOnMetadata

```
public boolean isSendOnMetadata()
```

True if sending onMetadata event at start of stream

Returns:

True if sending onMetadata event at start of stream

setSendOnMetadata

```
public void setSendOnMetadata(boolean sendOnMetadata)
```

(continued from last page)

True if sending onMetadata event at start of stream

Parameters:

sendOnMetadata - True if sending onMetadata event at start of stream

isRemoveDefaultAppInstance

```
public boolean isRemoveDefaultAppInstance( )
```

If true and destination application instance is `_definst_` then the application instance name will not be sent to the remove server as part of the connection information.

Returns:

true if omitting `_definst_`

setRemoveDefaultAppInstance

```
public void setRemoveDefaultAppInstance(boolean removeDefaultAppInstance)
```

If true and destination application instance is `_definst_` then the application instance name will not be sent to the remove server as part of the connection information.

Parameters:

removeDefaultAppInstance - true if omitting `_definst_`

isOnMetadataToSetDataFrame

```
public boolean isOnMetadataToSetDataFrame( )
```

If true all onMetadata events are converted to `@setDataFrame` calls

Returns:

true all onMetadata events are converted to `@setDataFrame` calls

setOnMetadataToSetDataFrame

```
public void setOnMetadataToSetDataFrame(boolean onMetadataToSetDataFrame)
```

If true all onMetadata events are converted to `@setDataFrame` calls

Parameters:

onMetadataToSetDataFrame - true all onMetadata events are converted to `@setDataFrame` calls

isSendStreamCloseCommands

```
public boolean isSendStreamCloseCommands( )
```

If true will send FCUnpublish, closeStream, deleteStream on disconnect

Returns:

true will send close commands

setSendStreamCloseCommands

```
public void setSendStreamCloseCommands(boolean sendStreamCloseCommands)
```

If true will send FCUnpublish, closeStream, deleteStream on disconnect

(continued from last page)

Parameters:

sendStreamCloseCommands - true will send close commands

convertDataEvenToAMFData

```
public AMFDataList convertDataEvenToAMFData(AMFPacket packet)
```

convertOnMetadataToSetDataFrame

```
public AMFPacket convertOnMetadataToSetDataFrame(AMFPacket packet)
```

getStreamCloseWaitTime

```
public int getStreamCloseWaitTime()
```

Get stream close timeout (milliseconds)

Returns:

stream close timeout (milliseconds)

setStreamCloseWaitTime

```
public void setStreamCloseWaitTime(int streamCloseWaitTime)
```

Set stream close timeout (milliseconds)

Parameters:

streamCloseWaitTime - stream close timeout (milliseconds)

disconnect

```
public void disconnect()
```

Disconnect from server

isFlashVerionFMLE

```
public static boolean isFlashVerionFMLE(String flashVersion)
```

Returns true if the Flash player version string is a Flash Media Live Encoder version (start with FMLE/ or FME/)

Parameters:

flashVersion - true if Flash Media Live Encoder version

Returns:

true if flash

isFlashVersionFMLE

```
public static boolean isFlashVersionFMLE(String flashVersion)
```

Returns true if the Flash player version string is a Flash Media Live Encoder version (start with FMLE/ or FME/)

Parameters:

(continued from last page)

flashVersion - true if Flash Media Live Encoder version

Returns:

true if flash

isDebugPackets

```
public boolean isDebugPackets( )
```

Set to true to log every packet being sent

Returns:

true to log every packet being sent

setDebugPackets

```
public void setDebugPackets(boolean debugPackets)
```

Set to true to log every packet being sent

Parameters:

debugPackets - true to log every packet being sent

getTotalPacketsSent

```
public long getTotalPacketsSent( )
```

Get total number of packets sent during push publishing session

Returns:

total number of packets sent during push publishing session

isResetOnTimecodeOutOfOrder

```
public boolean isResetOnTimecodeOutOfOrder( )
```

Set to true if you wish push to reset if timecodes jump out of order.

Returns:

true if you wish push to reset if timecodes jump out of order

setResetOnTimecodeOutOfOrder

```
public void setResetOnTimecodeOutOfOrder(boolean resetOnTimecodeOutOfOrder)
```

Set to true if you wish push to reset if timecodes jump out of order.

Parameters:

resetOnTimecodeOutOfOrder - true if you wish push to reset if timecodes jump out of order

getTimecodesOutOfOrderThreshold

```
public int getTimecodesOutOfOrderThreshold( )
```

(continued from last page)

setTimecodesOutOfOrderThreshold

```
public void setTimecodesOutOfOrderThreshold(int timecodesOutOfOrderThreshold)
```

getLastVideoTC

```
public long getLastVideoTC()
```

Get the absolute timecode of the last sent video packet (milliseconds)

Returns:

absolute timecode of the last sent video packet (milliseconds)

getLastAudioTC

```
public long getLastAudioTC()
```

Get the absolute timecode of the last sent audio packet (milliseconds)

Returns:

absolute timecode of the last sent audio packet (milliseconds)

getLastDataTC

```
public long getLastDataTC()
```

Get the absolute timecode of the last sent data packet (milliseconds)

Returns:

absolute timecode of the last sent data packet (milliseconds)

isWaitOnMetadataAvailable

```
public boolean isWaitOnMetadataAvailable()
```

Set to true to turn on system that will monitor onMetadata packet and wait for it to be populated before it starts sending the stream. Default is true. If stream is audio or video only, be sure to use setWaitOnMetadataVideo and setWaitOnMetadataAudio to properly turn off the channel you are not using or the system will always timeout.

Returns:

true to turn on system that will monitor onMetadata packet

setWaitOnMetadataAvailable

```
public void setWaitOnMetadataAvailable(boolean waitOnMetadataAvailable)
```

Set to true to turn on system that will monitor onMetadata packet and wait for it to be populated before it starts sending the stream. Default is true. If stream is audio or video only, be sure to use setWaitOnMetadataVideo and setWaitOnMetadataAudio to properly turn off the channel you are not using or the system will always timeout.

Parameters:

waitOnMetadataAvailable - true to turn on system that will monitor onMetadata packet

isWaitOnMetadataVideo

```
public boolean isWaitOnMetadataVideo()
```

(continued from last page)

If true and waitOnMetadataAvailable, publishing will not start until onMetadata packet is populated with video data.

Returns:

publishing will not start until onMetadata packet is populated with video data

setWaitOnMetadataVideo

```
public void setWaitOnMetadataVideo(boolean waitOnMetadataVideo)
```

If true and waitOnMetadataAvailable, publishing will not start until onMetadata packet is populated with video data.

Parameters:

waitOnMetadataVideo - publishing will not start until onMetadata packet is populated with video data

isWaitOnMetadataAudio

```
public boolean isWaitOnMetadataAudio()
```

If true and waitOnMetadataAvailable, publishing will not start until onMetadata packet is populated with audio data.

Returns:

publishing will not start until onMetadata packet is populated with audio data

setWaitOnMetadataAudio

```
public void setWaitOnMetadataAudio(boolean waitOnMetadataAudio)
```

If true and waitOnMetadataAvailable, publishing will not start until onMetadata packet is populated with audio data.

Parameters:

waitOnMetadataAudio - publishing will not start until onMetadata packet is populated with audio data

getWaitOnMetadataTimeout

```
public long getWaitOnMetadataTimeout()
```

Timeout (milliseconds) that the wait on onMetadata system will wait looking for video and audio data to be populated.

Returns:

timeout (milliseconds) that the wait on onMetadata system will wait looking for video and audio data to be populated

setWaitOnMetadataTimeout

```
public void setWaitOnMetadataTimeout(long waitOnMetadataTimeout)
```

Timeout (milliseconds) that the wait on onMetadata system will wait looking for video and audio data to be populated.

Parameters:

waitOnMetadataTimeout - timeout (milliseconds) that the wait on onMetadata system will wait looking for video and audio data to be populated

isWaitOnMetadataVideoDataRate

```
public boolean isWaitOnMetadataVideoDataRate()
```

(continued from last page)

setWaitOnMetadataVideoDatarate

```
public void setWaitOnMetadataVideoDatarate(boolean waitOnMetadataVideoDatarate)
```

isWaitOnMetadataAudioDatarate

```
public boolean isWaitOnMetadataAudioDatarate()
```

setWaitOnMetadataAudioDatarate

```
public void setWaitOnMetadataAudioDatarate(boolean waitOnMetadataAudioDatarate)
```

getSrcStreamBufferSize

```
public long getSrcStreamBufferSize()
```

The amount of audio/video in milliseconds that must be available in the source streams buffer before push publishing will being.

Returns:

amount of audio/video in milliseconds that must be available

setSrcStreamBufferSize

```
public void setSrcStreamBufferSize(long srcStreamBufferSize)
```

The amount of audio/video in milliseconds that must be available in the source streams buffer before push publishing will being.

Parameters:

srcStreamBufferSize - amount of audio/video in milliseconds that must be available

addConnectMetaData

```
public void addConnectMetaData(String key,  
    AMFData value)
```

Added metadata values to the connect metadata object

Parameters:

key - key

value - AMFDataObj value

createPushPublishSession

```
public IPushPublishSession createPushPublishSession()
```

getPacketsSentAudio

```
public long getPacketsSentAudio()
```

(continued from last page)

getPacketsSentVideo

```
public long getPacketsSentVideo()
```

getPacketsSentData

```
public long getPacketsSentData()
```

isSSL

```
public boolean isSSL()
```

setSSL

```
public void setSSL(boolean isSSL)
```

getBindAddress

```
public String getBindAddress()
```

setBindAddress

```
public void setBindAddress(String bindAddress)
```

isConnected

```
public Boolean isConnected()
```

getMessagesAttemptedCount

```
public long getMessagesAttemptedCount()
```

getMessagesRetriedCount

```
public long getMessagesRetriedCount()
```

(continued from last page)

getMessagesFailedCount

```
public long getMessagesFailedCount()
```

getSyncTimecode

```
public long getSyncTimecode()
```

setSyncTimecode

```
public void setSyncTimecode(long syncTimecode)
```

getSyncPacketType

```
public int getSyncPacketType()
```

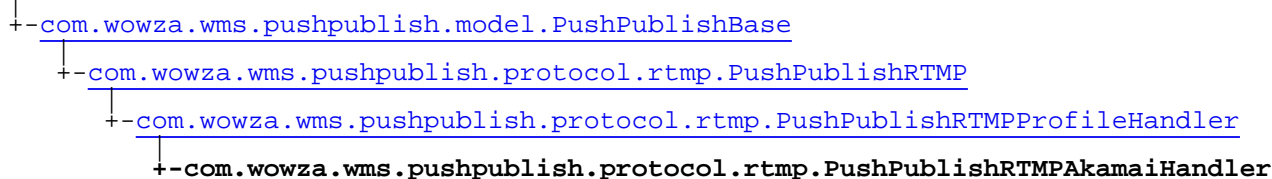
setSyncPacketType

```
public void setSyncPacketType(int syncPacketType)
```

com.wowza.wms.pushpublish.protocol.rtmp

Class PushPublishRTMPAkamaiHandler

java.lang.Object



All Implemented Interfaces:

[IPushPublish](#)

public class **PushPublishRTMPAkamaiHandler**

extends [PushPublishRTMPProfileHandler](#)

Field Summary

public static final	CLASS
public static final	CLASSNAME Value: PushPublishRTMPAkamaiHandler
protected	SendToBackupServer The is backup.

Fields inherited from class [com.wowza.wms.pushpublish.protocol.rtmp.PushPublishRTMPProfileHandler](#)

[CLASS](#), [CLASSNAME](#), [streamId](#)

Fields inherited from class [com.wowza.wms.pushpublish.protocol.rtmp.PushPublishRTMP](#)

[akamaiOriginIp](#), [akamaiOriginPort](#), [akamaiPassword](#), [akamaiUserName](#), [bindAddress](#), [bytesAvg](#), [CLASS](#), [CLASSNAME](#), [connected](#), [connectionFlashVersion](#), [connectionPageURL](#), [connectionQueryStr](#), [connectionResetCount](#), [connectionSwfURL](#), [connectionTimeout](#), [connectLastAttempt](#), [connectLastSuccess](#), [connectMetaData](#), [CURRENTFLASHVERSION](#), [CURRENTFMLEVERSION](#), [debugLogLastPacketCount](#), [debugPackets](#), [doSendOnMetaData](#), [dstAppInstanceName](#), [dstApplicationName](#), [dstStreamName](#), [howToPublish](#), [ioSessionToSessionHolder](#), [isReconnectorRunning](#), [isSSL](#), [lastAudioTC](#), [lastDataTC](#), [lastIdleReadTime](#), [lastIdleWriteTime](#), [lastVideoTC](#), [listeners](#), [lock](#), [netconnectToSessionHolder](#), [offlineTime](#), [onlineTime](#), [onMetadataToSetDataFrame](#), [outprint](#), [packetsSentAudio](#), [packetsSentData](#), [packetsSentVideo](#), [removeDefaultAppInstance](#), [resetOnTimecodeOutOfOrder](#), [rtmpAuthProvider](#), [rtmpDebugLog](#), [secureTokenSharedSecret](#), [sendFCAnnounce](#), [sendFCPublish](#), [sendOnMetadata](#), [sendOriginalTimecodes](#), [sendReleaseStream](#), [sendStreamCloseCommands](#), [sendStreamCloseCommandsSendTime](#), [sessionOrder](#), [srcStreamBufferSize](#), [streamCloseWaitTime](#), [syncPacketType](#), [syncTimecode](#), [timecodesOutOfOrderThreshold](#), [totalPacketsSent](#), [tryConnect](#), [validationFrequency](#), [waitOnMetadataAudio](#), [waitOnMetadataAudioDataRate](#), [waitOnMetadataAvailable](#), [waitOnMetadataReady](#), [waitOnMetadataStartime](#), [waitOnMetadataTimeout](#), [waitOnMetadataVideo](#), [waitOnMetadataVideoDataRate](#)

Fields inherited from class [com.wowza.wms.pushpublish.model.PushPublishBase](#)

[appInstance](#), [CLASS](#), [CLASSNAME](#), [dstStreamName](#), [hostname](#), [inetAddr](#), [mapFileIndex](#), [mImplementation](#), [mType](#), [originalTimecodeThreshold](#), [password](#), [port](#), [properties](#), [pushPublishStreamDebug](#), [pushPublishType](#), [reconnectWaitTime](#), [sendingPerformance](#), [sendOriginalTimecodes](#), [srcStream](#), [srcStreamName](#), [userName](#)

Fields inherited from interface [com.wowza.wms.pushpublish.model.IPushPublish](#)

[PUSHPUBLISH_TYPE_RTMP](#), [PUSHPUBLISH_TYPE_RTP](#), [PUSHPUBLISH_TYPE_UNKNOWN](#)

Constructor Summary

public	PushPublishRTMPAkamaiHandler()
--------	--

Method Summary

String	createAkamaiStreamName (String dstStreamName)
String	getStreamAngle ()
String	getStreamBandWidth ()
void	init (IApplicationInstance appInstance2, String streamName, IMediaStream stream, java.util.HashMap profileData, java.util.HashMap maps, boolean debug)
void	load (java.util.HashMap dataMap)
void	setAkamaiOriginIp (String akamaiOriginIp) Set Akamai Origin IP.
void	setAkamaiOriginPort (String akamaiOriginPort)
void	setAkamaiPassword (String string)
void	setAkamaiStreamId (String id) Sets the akamai stream id.
void	setAkamaiUserName (String string)
void	setPassword (String password)
void	setStreamAngle (String streamAngleIn)
void	setStreamBandWidth (String streamBandWidth)
void	setStreamId (String id)
void	setUserName (String userName)

Methods inherited from class

[com.wowza.wms.pushpublish.protocol.rtmp.PushPublishRTMPProfileHandler](#)

[connect_Out](#), [getDstAppInstanceName](#), [getDstApplicationName](#), [getLocalBindAddress](#), [init](#), [isDebugLog](#), [isDebugPackets](#), [load](#), [setApplication](#), [setConnectionFlashVersion](#), [setConnectionInfoPageURL](#), [setConnectionInfoQueryString](#), [setConnectionInfoSwfURL](#), [setConnectionTimeout](#), [setConnectLastAttempt](#), [setConnectLastSuccess](#), [setDebugPackets](#), [setDebugPackets](#), [setDstAppInstanceName](#), [setDstApplicationName](#), [setDstStreamName](#), [setLocalBindAddress](#), [setOnMetadataToSetDataFrame](#), [setOriginalTimecodeThreshold](#), [setReconnectWaitTime](#), [setRemoveDefaultAppInstance](#), [setResetOnTimecodeOutOfOrder](#), [setSecureTokenSharedSecret](#), [setSendFCAnnounce](#), [setSendFCPublish](#), [setSendOnMetadata](#), [setSendOriginalTimecodes](#), [setSendReleaseStream](#), [setSendStreamCloseCommands](#), [setSSL](#), [setStreamCloseWaitTime](#), [setTimecodesOutOfOrderThreshold](#), [setValidationFrequency](#), [setWaitOnMetadataAudio](#), [setWaitOnMetadataAudioDataRate](#), [setWaitOnMetadataAvailable](#), [setWaitOnMetadataTimeout](#), [setWaitOnMetadataVideo](#), [setWaitOnMetadataVideoDataRate](#)

Methods inherited from class [com.wowza.wms.pushpublish.protocol.rtmp.PushPublishRTMP](#)

[addConnectMetaData](#), [addListener](#), [addSession](#), [bumpReconnectWaitTime](#), [callConnect](#),
[connect](#), [convertDataEvenToAMFData](#), [convertOnMetadataToSetDataFrame](#),
[createPushPublishSession](#), [disconnect](#), [getAkamaiOriginIp](#), [getAkamaiOriginPort](#),
[getAkamaiPassword](#), [getAkamaiSessionKey](#), [getAkamaiUserName](#), [getAppInstance](#),
[getBindAddress](#), [getConnectionFlashVerion](#), [getConnectionFlashVersion](#),
[getConnectionPageURL](#), [getConnectionQueryStr](#), [getConnectionSwfURL](#),
[getConnectionTimeout](#), [getConnectLastAttempt](#), [getConnectLastSuccess](#), [getContextStr](#),
[getDstAppInstanceName](#), [getDstApplicationName](#), [getDstStreamName](#), [getHostname](#),
[getHowToPublish](#), [getLastAudioTC](#), [getLastDataTC](#), [getLastSession](#), [getLastVideoTC](#),
[getLocalListeners](#), [getLock](#), [getMessagesAttemptedCount](#), [getMessagesFailedCount](#),
[getMessagesRetriedCount](#), [getNetConnection](#), [getOriginalTimecodeThreshold](#),
[getPacketsSentAudio](#), [getPacketsSentData](#), [getPacketsSentVideo](#), [getPort](#),
[getReconnectDelay](#), [getRTMPAuthProvider](#), [getSecureTokenSharedSecret](#),
[getSessionHolderByNetConnection](#), [getSessionHolderBySession](#), [getSessionListDebug](#),
[getSrcStreamBufferSize](#), [getSrcStreamName](#), [getStreamCloseWaitTime](#), [getSyncPacketType](#),
[getSyncTimecode](#), [getTimecodesOutOfOrderThreshold](#), [getTotalPacketsSent](#),
[getValidationFrequency](#), [getWaitOnMetadataTimeout](#), [internalConnect](#), [isConnected](#),
[isCurrentSession](#), [isDebugLog](#), [isDebugPackets](#), [isFlashVerionFMLE](#), [isFlashVersionFMLE](#),
[isOnMetadataToSetDataFrame](#), [isRemoveDefaultAppInstance](#), [isResetOnTimecodeOutOfOrder](#),
[isSendFCAnnounce](#), [isSendFCPublish](#), [isSendOnMetadata](#), [isSendOriginalTimecodes](#),
[isSendReleaseStream](#), [isSendStreamCloseCommands](#), [isSSL](#), [isWaitOnMetadataAudio](#),
[isWaitOnMetadataAudioDatarate](#), [isWaitOnMetadataAvailable](#), [isWaitOnMetadataVideo](#),
[isWaitOnMetadataVideoDatarate](#), [notifyAkamaiClientLogin](#), [notifyAkamaiSetChallenge](#),
[notifyAkamaiSetOriginConnectionInfo](#), [notifyConnect](#), [notifyConnectFailure](#),
[notifyConnectStart](#), [notifyConnectSuccess](#), [notifyFCAnnounce](#), [notifyFCPublish](#),
[notifyHandshakeResult](#), [notifyPublishHandlerPlay](#), [notifyPushPublisherSessionCreate](#),
[notifyPushPublisherSessionDestroy](#), [notifySessionClosed](#), [notifySessionIdle](#),
[notifySessionOpened](#), [notifyStreamCreate](#), [notifyStreamOnPlayStatus](#),
[notifyStreamOnStatus](#), [notifyValidateSession](#), [notifyValidateSessionResult](#),
[removeListener](#), [removeSession](#), [resetConnection](#), [resetConnection](#), [resolveHostName](#),
[sessionClosed](#), [sessionOpened](#), [setAkamaiOriginIp](#), [setAkamaiOriginPort](#),
[setAkamaiPassword](#), [setAkamaiUserName](#), [setAppInstance](#), [setBindAddress](#),
[setConnectionFlashVerion](#), [setConnectionFlashVersion](#), [setConnectionPageURL](#),
[setConnectionQueryStr](#), [setConnectionSwfURL](#), [setConnectionTimeout](#),
[setConnectLastAttempt](#), [setConnectLastSuccess](#), [setDebugLog](#), [setDebugPackets](#),
[setDstAppInstanceName](#), [setDstApplicationName](#), [setDstStreamName](#), [setHostname](#),
[setHowToPublish](#), [setOnMetadataToSetDataFrame](#), [setOriginalTimecodeThreshold](#), [setPort](#),
[setRemoveDefaultAppInstance](#), [setResetOnTimecodeOutOfOrder](#), [setRTMPAuthProvider](#),
[setSecureTokenSharedSecret](#), [setSendFCAnnounce](#), [setSendFCPublish](#), [setSendOnMetadata](#),
[setSendOriginalTimecodes](#), [setSendReleaseStream](#), [setSendStreamCloseCommands](#),
[setSrcStreamBufferSize](#), [setSrcStreamName](#), [setSSL](#), [setStreamCloseWaitTime](#),
[setSyncPacketType](#), [setSyncTimecode](#), [setTimecodesOutOfOrderThreshold](#),
[setValidationFrequency](#), [setWaitOnMetadataAudio](#), [setWaitOnMetadataAudioDatarate](#),
[setWaitOnMetadataAvailable](#), [setWaitOnMetadataTimeout](#), [setWaitOnMetadataVideo](#),
[setWaitOnMetadataVideoDatarate](#), [tryToConnect](#)

Methods inherited from class [com.wowza.wms.pushpublish.model.PushPublishBase](#)

[createPushPublishSession](#), [getAppInstance](#), [getBandwidth](#), [getConfigDataOrDefault](#), [getConfigDataOrDefault](#), [getConfigDataOrDefault](#), [getContextStr](#), [getDstStreamName](#), [getHostname](#), [getImplementation](#), [getMapFileIndex](#), [getNumberFromString](#), [getOriginalTimecodeThreshold](#), [getPassword](#), [getPort](#), [getProperties](#), [getPushPublishType](#), [getPushPublishVersionStr](#), [getReconnectWaitTime](#), [getSrcStream](#), [getSrcStreamName](#), [getUserName](#), [init](#), [isAdaptiveStreaming](#), [isSendOriginalTimecodes](#), [load](#), [setAdaptiveStreaming](#), [setAppInstance](#), [setDebugLog](#), [setDstStreamName](#), [setHost](#), [setHostname](#), [setImplementation](#), [setMapFileIndex](#), [setOriginalTimecodeThreshold](#), [setOriginalTimecodeThreshold](#), [setPassword](#), [setPort](#), [setPort](#), [setReconnectWaitTime](#), [setSendOriginalTimecodes](#), [setSrcStream](#), [setSrcStreamName](#), [setStreamName](#), [setUserName](#)

Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Methods inherited from interface [com.wowza.wms.pushpublish.model.IPushPublish](#)

[connect](#), [createPushPublishSession](#), [disconnect](#), [getContextStr](#), [getDstStreamName](#), [getHostname](#), [getMapFileIndex](#), [getMessagesAttemptedCount](#), [getMessagesFailedCount](#), [getMessagesRetriedCount](#), [getPort](#), [getProperties](#), [getPushPublishType](#), [getReconnectWaitTime](#), [getSrcStream](#), [getSrcStreamName](#), [init](#), [isConnected](#), [resetConnection](#), [setDebugLog](#), [setDstStreamName](#), [setHost](#), [setHostname](#), [setMapFileIndex](#), [setPort](#), [setPort](#), [setReconnectWaitTime](#), [setSrcStream](#), [setSrcStreamName](#), [setStreamName](#)

Fields

CLASS

`public static final java.lang.Class CLASS`

CLASSNAME

`public static final java.lang.String CLASSNAME`

Constant value: **PushPublishRTMPAkamaiHandler**

SendToBackupServer

`protected boolean SendToBackupServer`

The is backup.

Constructors

PushPublishRTMPAkamaiHandler

`public PushPublishRTMPAkamaiHandler()`

(continued from last page)

Methods

init

```
public void init(IApplicationInstance appInstance2,  
                String streamName,  
                IMediaStream stream,  
                java.util.HashMap profileData,  
                java.util.HashMap maps,  
                boolean debug)
```

Init the instance.

load

```
public void load(java.util.HashMap dataMap)
```

setUserName

```
public void setUserName(String userName)
```

Set Akamai user name.

setPassword

```
public void setPassword(String password)
```

Set Akamai password.

setAkamaiOriginIp

```
public void setAkamaiOriginIp(String akamaiOriginIp)
```

Set Akamai Origin IP.

Parameters:

akamaiOriginIp - Akamai Origin IP

setAkamaiOriginPort

```
public void setAkamaiOriginPort(String akamaiOriginPort)
```

setAkamaiStreamId

```
public void setAkamaiStreamId(String id)
```

Sets the akamai stream id.

Parameters:

id - the new akamai stream id

setStreamId

```
public void setStreamId(String id)
```

(continued from last page)

setAkamaiUserName

```
public void setAkamaiUserName(String string)
```

Set Akamai user name

setAkamaiPassword

```
public void setAkamaiPassword(String string)
```

Set Akamai password

getStreamAngle

```
public String getStreamAngle()
```

setStreamAngle

```
public void setStreamAngle(String streamAngleIn)
```

getStreamBandWidth

```
public String getStreamBandWidth()
```

setStreamBandWidth

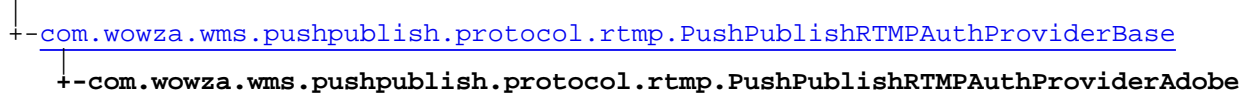
```
public void setStreamBandWidth(String streamBandWidth)
```

createAkamaiStreamName

```
public String createAkamaiStreamName(String dstStreamName)
```

com.wowza.wms.pushpublish.protocol.rtmp Class PushPublishRTMPAuthProviderAdobe

java.lang.Object



All Implemented Interfaces:

[IPushPublishRTMPAuthProvider](#)

public class **PushPublishRTMPAuthProviderAdobe**
extends [PushPublishRTMPAuthProviderBase](#)

Field Summary

protected	authMode
protected	challenge
public static final	CLASS
public static final	CLASSNAME Value: PushPublishRTMPAuthProviderAdobe
protected	md5Digest
protected	opaque
protected	salt

Fields inherited from class

[com.wowza.wms.pushpublish.protocol.rtmp.PushPublishRTMPAuthProviderBase](#)

[lock](#), [PARAM1](#), [password](#), [publisher](#), [userName](#)

Constructor Summary

public	PushPublishRTMPAuthProviderAdobe()
--------	--

Method Summary

String	addAuthQueryString (String queryString)
void	handleConnectResponse (com.wowza.wms.request.RequestFunction function, AMFDataList params)
void	init (PushPublishRTMP publisher)

Methods inherited from class[com.wowza.wms.pushpublish.protocol.rtmp.PushPublishRTMPAuthProviderBase](#)[getParamObj](#), [getPassword](#), [getPublisher](#), [getUserName](#), [init](#), [setPassword](#), [setUserName](#), [splitQueryStr](#)**Methods inherited from class** java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface[com.wowza.wms.pushpublish.protocol.rtmp.IPushPublishRTMPAuthProvider](#)[addAuthQueryString](#), [getPassword](#), [getPublisher](#), [getUserName](#), [handleConnectResponse](#), [init](#), [setPassword](#), [setUserName](#)

Fields

CLASS

```
public static final java.lang.Class CLASS
```

CLASSNAME

```
public static final java.lang.String CLASSNAME
```

Constant value: **PushPublishRTMPAuthProviderAdobe**

authMode

```
protected java.lang.String authMode
```

salt

```
protected java.lang.String salt
```

challenge

```
protected java.lang.String challenge
```

opaque

```
protected java.lang.String opaque
```

md5Digest

protected java.security.MessageDigest **md5Digest**

Constructors

PushPublishRTMPAuthProviderAdobe

```
public PushPublishRTMPAuthProviderAdobe()
```

Methods

init

```
public void init(PushPublishRTMP publisher)
```

addAuthQueryString

```
public String addAuthQueryString(String queryString)
```

handleConnectResponse

```
public void handleConnectResponse(com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

com.wowza.wms.pushpublish.protocol.rtmp Class PushPublishRTMPAuthProviderBase

java.lang.Object

└─com.wowza.wms.pushpublish.protocol.rtmp.PushPublishRTMPAuthProviderBase

All Implemented Interfaces:

[IPushPublishRTMPAuthProvider](#)

Direct Known Subclasses:

[PushPublishRTMPAuthProviderLimelight](#), [PushPublishRTMPAuthProviderAdobe](#)

public abstract class **PushPublishRTMPAuthProviderBase**

extends Object

implements [IPushPublishRTMPAuthProvider](#)

Field Summary

protected	lock
public static final	PARAM1 Value: 3
protected	password
protected	publisher
protected	userName

Constructor Summary

public	PushPublishRTMPAuthProviderBase()
--------	---

Method Summary

AMFDataObj	getParamObj() AMFDataList params, int index)
String	getPassword()
PushPublishRTMP	getPublisher()
String	getUserName()
void	init() (PushPublishRTMP publisher)
void	setPassword() (String password)

void	setUserName (String userName)
java.util.Map	splitQueryStr (String queryStr)

Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Methods inherited from interface

[com.wowza.wms.pushpublish.protocol.rtmp.IPushPublishRTMPAuthProvider](#)

[addAuthQueryString](#), [getPassword](#), [getPublisher](#), [getUserName](#), [handleConnectResponse](#), [init](#), [setPassword](#), [setUserName](#)

Fields

PARAM1

```
public static final int PARAM1
```

Constant value: **3**

publisher

```
protected com.wowza.wms.pushpublish.protocol.rtmp.PushPublishRTMP publisher
```

userName

```
protected java.lang.String userName
```

password

```
protected java.lang.String password
```

lock

```
protected java.lang.Object lock
```

Constructors

PushPublishRTMPAuthProviderBase

```
public PushPublishRTMPAuthProviderBase()
```

Methods

init

```
public void init(PushPublishRTMP publisher)
```

getParamObj

```
protected AMFDataObj getParamObj(AMFDataList params,  
    int index)
```

getUserName

```
public String getUserName()
```

setUserName

```
public void setUserName(String userName)
```

getPassword

```
public String getPassword()
```

setPassword

```
public void setPassword(String password)
```

getPublisher

```
public PushPublishRTMP getPublisher()
```

splitQueryStr

```
public java.util.Map splitQueryStr(String queryStr)
```

com.wowza.wms.pushpublish.protocol.rtmp Class PushPublishRTMPAuthProviderLimelight

java.lang.Object

+ [com.wowza.wms.pushpublish.protocol.rtmp.PushPublishRTMPAuthProviderBase](#)
 + [com.wowza.wms.pushpublish.protocol.rtmp.PushPublishRTMPAuthProviderLimelight](#)

All Implemented Interfaces:

[IPushPublishRTMPAuthProvider](#)

public class **PushPublishRTMPAuthProviderLimelight**
 extends [PushPublishRTMPAuthProviderBase](#)

Field Summary

protected	authMode
public static final	CLASS
public static final	CLASSNAME Value: PushPublishRTMPAuthProviderLimelight
protected	conce
protected	methodValue
protected	ncCounter
protected	nonce
protected	qopValue
protected	realm

Fields inherited from class

[com.wowza.wms.pushpublish.protocol.rtmp.PushPublishRTMPAuthProviderBase](#)

[lock](#), [PARAM1](#), [password](#), [publisher](#), [userName](#)

Constructor Summary

public	PushPublishRTMPAuthProviderLimelight()
--------	--

Method Summary

String	addAuthQueryString (String queryString)
--------	---

String	getAuthMode()
String	getConce()
String	getMethodValue()
int	getNCCounter()
String	getNonce()
PushPublishRTMP	getPublisher()
String	getQOPValue()
String	getRealm()
void	handleConnectResponse (com.wowza.wms.request.RequestFunction function, AMFDataList params)
void	init (PushPublishRTMP publisher)
void	initConce()
void	setAuthMode (String authMode)
void	setConce (String conce)
void	setMethodValue (String methodValue)
void	setNCCounter (int ncCounter)
void	setNonce (String nonce)
void	setQOPValue (String qopValue)
void	setRealm (String realm)

Methods inherited from class[com.wowza.wms.pushpublish.protocol.rtmp.PushPublishRTMPAuthProviderBase](#)

[getParamObj](#), [getPassword](#), [getPublisher](#), [getUserName](#), [init](#), [setPassword](#), [setUserName](#), [splitQueryStr](#)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface[com.wowza.wms.pushpublish.protocol.rtmp.IPushPublishRTMPAuthProvider](#)


```
addAuthQueryString, getPassword, getPublisher, getUserName, handleConnectResponse,  
init, setPassword, setUserName
```

Fields

CLASS

```
public static final java.lang.Class CLASS
```

CLASSNAME

```
public static final java.lang.String CLASSNAME
```

Constant value: **PushPublishRTMPAuthProviderLimelight**

realm

```
protected java.lang.String realm
```

methodValue

```
protected java.lang.String methodValue
```

qopValue

```
protected java.lang.String qopValue
```

authMode

```
protected java.lang.String authMode
```

nonce

```
protected java.lang.String nonce
```

conce

```
protected java.lang.String conce
```

(continued from last page)

ncCounter

```
protected int ncCounter
```

Constructors

PushPublishRTMPAuthProviderLimelight

```
public PushPublishRTMPAuthProviderLimelight()
```

Methods

init

```
public void init(PushPublishRTMP publisher)
```

initOnce

```
protected void initOnce()
```

addAuthQueryString

```
public String addAuthQueryString(String queryString)
```

handleConnectResponse

```
public void handleConnectResponse(com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

getPublisher

```
public PushPublishRTMP getPublisher()
```

getRealm

```
public String getRealm()
```

setRealm

```
public void setRealm(String realm)
```

getMethodValue

```
public String getMethodValue()
```

setMethodValue

```
public void setMethodValue(String methodValue)
```

getQOPValue

```
public String getQOPValue()
```

setQOPValue

```
public void setQOPValue(String qopValue)
```

getAuthMode

```
public String getAuthMode()
```

setAuthMode

```
public void setAuthMode(String authMode)
```

getNonce

```
public String getNonce()
```

setNonce

```
public void setNonce(String nonce)
```

getConce

```
public String getConce()
```

setConce

```
public void setConce(String conce)
```

(continued from last page)

getNCCounter

```
public int getNCCounter()
```

setNCCounter

```
public void setNCCounter(int ncCounter)
```

com.wowza.wms.pushpublish.protocol.rtmp

Class PushPublishRTMPNetConnectionSession

java.lang.Object

└─com.wowza.wms.pushpublish.protocol.rtmp.PushPublishRTMPNetConnectionSession

public class **PushPublishRTMPNetConnectionSession**
extends Object

Field Summary

protected	netConnection
protected	publishStream
protected	session
protected	streamsToPublish

Constructor Summary

public	PushPublishRTMPNetConnectionSession (com.wowza.wms.netconnection.INetConnection netConnection, org.apache.mina.common.Session session)
--------	--

Method Summary

com.wowza.wms.netconnection.INetConnection	getNetConnection ()
IMediaStream	getPublishStream ()
org.apache.mina.common.Session	getSession ()
java.util.List	getStreamsToPublish ()
void	setNetConnection (com.wowza.wms.netconnection.INetConnection netConnection)
void	setPublishStream (IMediaStream publishStream)
void	setSession (org.apache.mina.common.Session session)
void	setStreamsToPublish (java.util.List streamsToPublish)

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

Fields

session

```
protected org.apache.mina.common.Session session
```

netConnection

```
protected com.wowza.wms.netconnection.INetConnection netConnection
```

streamsToPublish

```
protected java.util.List streamsToPublish
```

publishStream

```
protected com.wowza.wms.stream.IMediaStream publishStream
```

Constructors

PushPublishRTMPNetConnectionSession

```
public PushPublishRTMPNetConnectionSession(com.wowza.wms.netconnection.INetConnection netConnection, org.apache.mina.common.Session session)
```

Methods

getSession

```
public org.apache.mina.common.Session getSession()
```

setSession

```
public void setSession(org.apache.mina.common.Session session)
```

(continued from last page)

getNetConnection

```
public com.wowza.wms.netconnection.INetConnection getNetConnection()
```

setNetConnection

```
public void setNetConnection(com.wowza.wms.netconnection.INetConnection netConnection)
```

getStreamsToPublish

```
public java.util.List getStreamsToPublish()
```

setStreamsToPublish

```
public void setStreamsToPublish(java.util.List streamsToPublish)
```

getPublishStream

```
public IMediaStream getPublishStream()
```

setPublishStream

```
public void setPublishStream(IMediaStream publishStream)
```

com.wowza.wms.pushpublish.protocol.rtmp Class PushPublishRTMPNotifyBase

java.lang.Object

└-com.wowza.wms.pushpublish.protocol.rtmp.PushPublishRTMPNotifyBase

All Implemented Interfaces:

[IPushPublishRTMPNotify](#)

public abstract class **PushPublishRTMPNotifyBase**

extends Object

implements [IPushPublishRTMPNotify](#)

Abstract base class for [IPushPublishRTMPNotify](#) interface

See Also:

[PushPublishRTMP.addListener\(IPushPublishRTMPNotify\)](#)

Constructor Summary

public	PushPublishRTMPNotifyBase()
--------	---

Method Summary

void	onAkamaiClientLogin (PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)
void	onAkamaiSetChallenge (PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)
void	onAkamaiSetOriginConnectionInfo (PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)
void	onConnect (PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)
void	onConnectFailure (PushPublishRTMPNetConnectionSession pushPublisherSession)
void	onConnectStart (PushPublishRTMPNetConnectionSession pushPublisherSession)
void	onConnectSuccess (PushPublishRTMPNetConnectionSession pushPublisherSession)
void	onFCAnnounce (PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)
void	onFCPublish (PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)

void	onHandshakeResult (PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)
void	onPublishHandlerPlay (PushPublishRTMPNetConnectionSession pushPublisherSession, java.io.OutputStream out, long[] playSizes)
void	onPushPublisherSessionCreate (PushPublishRTMPNetConnectionSession pushPublisherSession)
void	onPushPublisherSessionDestroy (PushPublishRTMPNetConnectionSession pushPublisherSession)
void	onReleaseStream (PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)
void	onSessionClosed (PushPublishRTMPNetConnectionSession pushPublisherSession)
void	onSessionIdle (PushPublishRTMPNetConnectionSession pushPublisherSession)
void	onSessionOpened (PushPublishRTMPNetConnectionSession pushPublisherSession)
void	onStreamCreate (PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)
void	onStreamOnPlayStatus (PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)
void	onStreamOnStatus (PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)
void	onValidateSession (PushPublishRTMPNetConnectionSession pushPublisherSession)
void	onValidateSessionResult (PushPublishRTMPNetConnectionSession pushPublisherSession, boolean result)

Methods inherited from class [java.lang.Object](#)

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

Methods inherited from interface [com.wowza.wms.pushpublish.protocol.rtmp.IPushPublishRTMPNotify](#)

[onAkamaiClientLogin](#), [onAkamaiSetChallenge](#), [onAkamaiSetOriginConnectionInfo](#), [onConnect](#), [onConnectFailure](#), [onConnectStart](#), [onConnectSuccess](#), [onFCAnnounce](#), [onFCPublish](#), [onHandshakeResult](#), [onPublishHandlerPlay](#), [onPushPublisherSessionCreate](#), [onPushPublisherSessionDestroy](#), [onReleaseStream](#), [onSessionClosed](#), [onSessionIdle](#), [onSessionOpened](#), [onStreamCreate](#), [onStreamOnPlayStatus](#), [onStreamOnStatus](#), [onValidateSession](#), [onValidateSessionResult](#)

Constructors

PushPublishRTMPNotifyBase

```
public PushPublishRTMPNotifyBase()
```

Methods

onConnectStart

```
public void onConnectStart(PushPublishRTMPNetConnectionSession pushPublisherSession)
```

onConnectSuccess

```
public void onConnectSuccess(PushPublishRTMPNetConnectionSession pushPublisherSession)
```

onConnectFailure

```
public void onConnectFailure(PushPublishRTMPNetConnectionSession pushPublisherSession)
```

onHandshakeResult

```
public void onHandshakeResult(PushPublishRTMPNetConnectionSession
pushPublisherSession,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

onAkamaiSetChallenge

```
public void onAkamaiSetChallenge(PushPublishRTMPNetConnectionSession
pushPublisherSession,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

onAkamaiSetOriginConnectionInfo

```
public void onAkamaiSetOriginConnectionInfo(PushPublishRTMPNetConnectionSession
pushPublisherSession,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

(continued from last page)

onAkamaiClientLogin

```
public void onAkamaiClientLogin(PushPublishRTMPNetConnectionSession
pushPublisherSession,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

onFCPublish

```
public void onFCPublish(PushPublishRTMPNetConnectionSession pushPublisherSession,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

onFCAnnounce

```
public void onFCAnnounce(PushPublishRTMPNetConnectionSession pushPublisherSession,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

onReleaseStream

```
public void onReleaseStream(PushPublishRTMPNetConnectionSession pushPublisherSession,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

onConnect

```
public void onConnect(PushPublishRTMPNetConnectionSession pushPublisherSession,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

onStreamCreate

```
public void onStreamCreate(PushPublishRTMPNetConnectionSession pushPublisherSession,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

onStreamOnPlayStatus

```
public void onStreamOnPlayStatus(PushPublishRTMPNetConnectionSession
pushPublisherSession,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

(continued from last page)

onStreamOnStatus

```
public void onStreamOnStatus(PushPublishRTMPNetConnectionSession pushPublisherSession,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

onPublishHandlerPlay

```
public void onPublishHandlerPlay(PushPublishRTMPNetConnectionSession  
pushPublisherSession,  
    java.io.OutputStream out,  
    long[] playSizes)
```

onSessionIdle

```
public void onSessionIdle(PushPublishRTMPNetConnectionSession pushPublisherSession)
```

onSessionOpened

```
public void onSessionOpened(PushPublishRTMPNetConnectionSession pushPublisherSession)
```

onSessionClosed

```
public void onSessionClosed(PushPublishRTMPNetConnectionSession pushPublisherSession)
```

onValidateSession

```
public void onValidateSession(PushPublishRTMPNetConnectionSession  
pushPublisherSession)
```

onValidateSessionResult

```
public void onValidateSessionResult(PushPublishRTMPNetConnectionSession  
pushPublisherSession,  
    boolean result)
```

onPushPublisherSessionCreate

```
public void onPushPublisherSessionCreate(PushPublishRTMPNetConnectionSession  
pushPublisherSession)
```

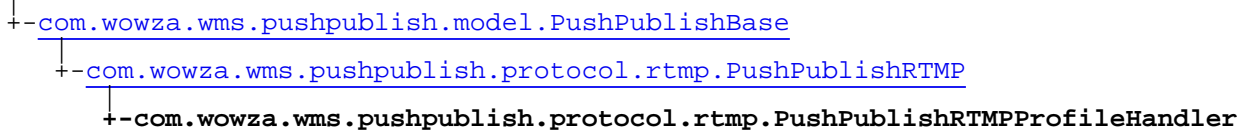
(continued from last page)

onPushPublisherSessionDestroy

```
public void onPushPublisherSessionDestroy(PushPublishRTMPNetConnectionSession  
pushPublisherSession)
```

com.wowza.wms.pushpublish.protocol.rtmp Class PushPublishRTMPProfileHandler

java.lang.Object



All Implemented Interfaces:

[IPushPublish](#)

Direct Known Subclasses:

[PushPublishRTMPUstream](#), [PushPublishRTMPAkamaiHandler](#)

public class **PushPublishRTMPProfileHandler**
extends [PushPublishRTMP](#)

Field Summary

public static final	CLASS
public static final	CLASSNAME Value: PushPublishRTMPProfileHandler
protected	streamId The stream id.

Fields inherited from class [com.wowza.wms.pushpublish.protocol.rtmp.PushPublishRTMP](#)

[akamaiOriginIp](#), [akamaiOriginPort](#), [akamaiPassword](#), [akamaiUserName](#), [bindAddress](#), [bytesAvg](#), [CLASS](#), [CLASSNAME](#), [connected](#), [connectionFlashVersion](#), [connectionPageURL](#), [connectionQueryStr](#), [connectionResetCount](#), [connectionSwfURL](#), [connectionTimeout](#), [connectLastAttempt](#), [connectLastSuccess](#), [connectMetaData](#), [CURRENTFLASHVERSION](#), [CURRENTFMLEVERSION](#), [debugLogLastPacketCount](#), [debugPackets](#), [doSendOnMetaData](#), [dstAppInstanceName](#), [dstApplicationName](#), [dstStreamName](#), [howToPublish](#), [ioSessionToSessionHolder](#), [isReconnectorRunning](#), [isSSL](#), [lastAudioTC](#), [lastDataTC](#), [lastIdleReadTime](#), [lastIdleWriteTime](#), [lastVideoTC](#), [listeners](#), [lock](#), [netconnectToSessionHolder](#), [offlineTime](#), [onlineTime](#), [onMetadataToSetDataFrame](#), [outprint](#), [packetsSentAudio](#), [packetsSentData](#), [packetsSentVideo](#), [removeDefaultAppInstance](#), [resetOnTimecodeOutOfOrder](#), [rtmpAuthProvider](#), [rtmpDebugLog](#), [secureTokenSharedSecret](#), [sendFCAnnounce](#), [sendFCPublish](#), [sendOnMetadata](#), [sendOriginalTimecodes](#), [sendReleaseStream](#), [sendStreamCloseCommands](#), [sendStreamCloseCommandsSendTime](#), [sessionOrder](#), [srcStreamBufferSize](#), [streamCloseWaitTime](#), [syncPacketType](#), [syncTimecode](#), [timecodesOutOfOrderThreshold](#), [totalPacketsSent](#), [tryConnect](#), [validationFrequency](#), [waitOnMetadataAudio](#), [waitOnMetadataAudioDatarate](#), [waitOnMetadataAvailable](#), [waitOnMetadataReady](#), [waitOnMetadataStartime](#), [waitOnMetadataTimeout](#), [waitOnMetadataVideo](#), [waitOnMetadataVideoDatarate](#)

Fields inherited from class [com.wowza.wms.pushpublish.model.PushPublishBase](#)

[appInstance](#), [CLASS](#), [CLASSNAME](#), [dstStreamName](#), [hostname](#), [inetAddr](#), [mapFileIndex](#), [mImplementation](#), [mType](#), [originalTimecodeThreshold](#), [password](#), [port](#), [properties](#), [pushPublishStreamDebug](#), [pushPublishType](#), [reconnectWaitTime](#), [sendingPerformance](#), [sendOriginalTimecodes](#), [srcStream](#), [srcStreamName](#), [userName](#)

Fields inherited from interface [com.wowza.wms.pushpublish.model.IPushPublish](#)

[PUSHPUBLISH_TYPE_RTMP](#), [PUSHPUBLISH_TYPE_RTP](#), [PUSHPUBLISH_TYPE_UNKNOWN](#)

Constructor Summary

public	PushPublishRTMPPProfileHandler() Instantiates a new push publisher rtmp.
--------	---

Method Summary

void	connect_Out() Connect to server.
String	getDstAppInstanceName() Get destination appInstance name.
String	getDstApplicationName() Get destination application name.
String	getLocalBindAddress() Get local bind address for the connection.
void	init(IApplicationInstance appInstance2, String streamName, IMediaStream stream, java.util.HashMap profileData, java.util.HashMap maps, boolean debug)
boolean	isDebugLog() Is debug logging turned on.
boolean	isDebugPackets() Checks if is debug packets.
void	load(java.util.HashMap dataMap)
void	setApplication(String dstApplicationNameIN) Set destination application name.
void	setConnectionFlashVersion(String connectionFlashVersion) Set Flash version used in connection metadata.
void	setConnectionInfoPageURL(String connectionPageURL) Set Page URL used in connection metadata.
void	setConnectionInfoQueryString(String connectionQueryStr) Set the connection query string.
void	setConnectionInfoSwfURL(String connectionSwfURL) Set SWF URL used in connection metadata.
void	setConnectionTimeout(String connectionTimeout)

void	<u>setConnectLastAttempt</u> (String connectLastAttempt)
void	<u>setConnectLastSuccess</u> (String connectLastSuccess)
void	<u>setDebugPackets</u> (boolean debugPackets) Sets the debug packets.
void	<u>setDebugPackets</u> (String debugPacketsIn) Sets the debug packets.
void	<u>setDstAppInstanceName</u> (String dstAppInstanceName)
void	<u>setDstApplicationName</u> (String string) Sets the dst application name.
void	<u>setDstStreamName</u> (String dstStreamNameIn)
void	<u>setlocalBindAddress</u> (String bindAddress) Sets the local bind address for the connection.
void	<u>setOnMetadataToSetDataFrame</u> (String onMetadataToSetDataFrame)
void	<u>setOriginalTimecodeThreshold</u> (String originalTimecodeThreshold) Sets the original timecode threshold.
void	<u>setReconnectWaitTime</u> (String reconnectWaitTime)
void	<u>setRemoveDefaultAppInstance</u> (String removeDefaultAppInstance) Sets the removes the default app instance.
void	<u>setResetOnTimecodeOutOfOrder</u> (String resetOnTimecodeOutOfOrder) Set to true if you wish push to reset if timecodes jump out of order.
void	<u>setSecureTokenSharedSecret</u> (String secureTokenSharedSecret) Set SecureToken shared secret.
void	<u>setSendFCAnnounce</u> (String sendFCAnnounce)
void	<u>setSendFCPublish</u> (String sendFCPublish) Sets the send fc publish.
void	<u>setSendOnMetadata</u> (String sendOnMetadata)
void	<u>setSendOriginalTimecodes</u> (String sendOriginalTimecodes) Sets the send original timecodes.
void	<u>setSendReleaseStream</u> (String sendReleaseStream) Sets the send release stream.
void	<u>setSendStreamCloseCommands</u> (String sendStreamCloseCommands) Sets the send stream close commands.
void	<u>setSSL</u> (String sslStr)
void	<u>setStreamCloseWaitTime</u> (String streamCloseWaitTime)

void	<u>setTimecodesOutOfOrderThreshold</u> (String timecodesOutOfOrderThreshold)
void	<u>setValidationFrequency</u> (String validationFrequency)
void	<u>setWaitOnMetadataAudio</u> (String waitOnMetadataAudio)
void	<u>setWaitOnMetadataAudioDatarate</u> (String waitOnMetadataAudioDatarate)
void	<u>setWaitOnMetadataAvailable</u> (String waitOnMetadataAvailable)
void	<u>setWaitOnMetadataTimeout</u> (String waitOnMetadataTimeout) Timeout (milliseconds) that the wait on onMetadata system will wait looking for video and audio data to be populated.
void	<u>setWaitOnMetadataVideo</u> (String waitOnMetadataVideo)
void	<u>setWaitOnMetadataVideoDatarate</u> (String waitOnMetadataVideoDatarate)

Methods inherited from class [com.wowza.wms.pushpublish.protocol.rtmp.PushPublishRTMP](#)

[addConnectMetaData](#), [addListener](#), [addSession](#), [bumpReconnectWaitTime](#), [callConnect](#), [connect](#), [convertDataEvenToAMFData](#), [convertOnMetadataToSetDataFrame](#), [createPushPublishSession](#), [disconnect](#), [getAkamaiOriginIp](#), [getAkamaiOriginPort](#), [getAkamaiPassword](#), [getAkamaiSessionKey](#), [getAkamaiUserName](#), [getAppInstance](#), [getBindAddress](#), [getConnectionFlashVerion](#), [getConnectionFlashVersion](#), [getConnectionPageURL](#), [getConnectionQueryStr](#), [getConnectionSwfURL](#), [getConnectionTimeout](#), [getConnectLastAttempt](#), [getConnectLastSuccess](#), [getContextStr](#), [getDstAppInstanceName](#), [getDstApplicationName](#), [getDstStreamName](#), [getHostname](#), [getHowToPublish](#), [getLastAudioTC](#), [getLastDataTC](#), [getLastSession](#), [getLastVideoTC](#), [getLocalListeners](#), [getLock](#), [getMessagesAttemptedCount](#), [getMessagesFailedCount](#), [getMessagesRetriedCount](#), [getNetConnection](#), [getOriginalTimecodeThreshold](#), [getPacketsSentAudio](#), [getPacketsSentData](#), [getPacketsSentVideo](#), [getPort](#), [getReconnectDelay](#), [getRTMPAuthProvider](#), [getSecureTokenSharedSecret](#), [getSessionHolderByNetConnection](#), [getSessionHolderBySession](#), [getSessionListDebug](#), [getSrcStreamBufferSize](#), [getSrcStreamName](#), [getStreamCloseWaitTime](#), [getSyncPacketType](#), [getSyncTimecode](#), [getTimecodesOutOfOrderThreshold](#), [getTotalPacketsSent](#), [getValidationFrequency](#), [getWaitOnMetadataTimeout](#), [internalConnect](#), [isConnected](#), [isCurrentSession](#), [isDebugLog](#), [isDebugPackets](#), [isFlashVerionFMLE](#), [isFlashVersionFMLE](#), [isOnMetadataToSetDataFrame](#), [isRemoveDefaultAppInstance](#), [isResetOnTimecodeOutOfOrder](#), [isSendFCAnnounce](#), [isSendFCPublish](#), [isSendOnMetadata](#), [isSendOriginalTimecodes](#), [isSendReleaseStream](#), [isSendStreamCloseCommands](#), [isSSL](#), [isWaitOnMetadataAudio](#), [isWaitOnMetadataAudioDatarate](#), [isWaitOnMetadataAvailable](#), [isWaitOnMetadataVideo](#), [isWaitOnMetadataVideoDatarate](#), [notifyAkamaiClientLogin](#), [notifyAkamaiSetChallenge](#), [notifyAkamaiSetOriginConnectionInfo](#), [notifyConnect](#), [notifyConnectFailure](#), [notifyConnectStart](#), [notifyConnectSuccess](#), [notifyFCAnnounce](#), [notifyFCPublish](#), [notifyHandshakeResult](#), [notifyPublishHandlerPlay](#), [notifyPushPublisherSessionCreate](#), [notifyPushPublisherSessionDestroy](#), [notifySessionClosed](#), [notifySessionIdle](#), [notifySessionOpened](#), [notifyStreamCreate](#), [notifyStreamOnPlayStatus](#), [notifyStreamOnStatus](#), [notifyValidateSession](#), [notifyValidateSessionResult](#), [removeListener](#), [removeSession](#), [resetConnection](#), [resetConnection](#), [resolveHostName](#), [sessionClosed](#), [sessionOpened](#), [setAkamaiOriginIp](#), [setAkamaiOriginPort](#), [setAkamaiPassword](#), [setAkamaiUserName](#), [setAppInstance](#), [setBindAddress](#), [setConnectionFlashVerion](#), [setConnectionFlashVersion](#), [setConnectionPageURL](#), [setConnectionQueryStr](#), [setConnectionSwfURL](#), [setConnectionTimeout](#), [setConnectLastAttempt](#), [setConnectLastSuccess](#), [setDebugLog](#), [setDebugPackets](#), [setDstAppInstanceName](#), [setDstApplicationName](#), [setDstStreamName](#), [setHostname](#), [setHowToPublish](#), [setOnMetadataToSetDataFrame](#), [setOriginalTimecodeThreshold](#), [setPort](#), [setRemoveDefaultAppInstance](#), [setResetOnTimecodeOutOfOrder](#), [setRTMPAuthProvider](#), [setSecureTokenSharedSecret](#), [setSendFCAnnounce](#), [setSendFCPublish](#), [setSendOnMetadata](#), [setSendOriginalTimecodes](#), [setSendReleaseStream](#), [setSendStreamCloseCommands](#), [setSrcStreamBufferSize](#), [setSrcStreamName](#), [setSSL](#), [setStreamCloseWaitTime](#), [setSyncPacketType](#), [setSyncTimecode](#), [setTimecodesOutOfOrderThreshold](#), [setValidationFrequency](#), [setWaitOnMetadataAudio](#), [setWaitOnMetadataAudioDatarate](#), [setWaitOnMetadataAvailable](#), [setWaitOnMetadataTimeout](#), [setWaitOnMetadataVideo](#), [setWaitOnMetadataVideoDatarate](#), [tryToConnect](#)

Methods inherited from class [com.wowza.wms.pushpublish.model.PushPublishBase](#)

[createPushPublishSession](#), [getAppInstance](#), [getBandwidth](#), [getConfigDataOrDefault](#), [getConfigDataOrDefault](#), [getConfigDataOrDefault](#), [getContextStr](#), [getDstStreamName](#), [getHostname](#), [getImplementation](#), [getMapFileIndex](#), [getNumberFromString](#), [getOriginalTimecodeThreshold](#), [getPassword](#), [getPort](#), [getProperties](#), [getPushPublishType](#), [getPushPublishVersionStr](#), [getReconnectWaitTime](#), [getSrcStream](#), [getSrcStreamName](#), [getUserName](#), [init](#), [isAdaptiveStreaming](#), [isSendOriginalTimecodes](#), [load](#), [setAdaptiveStreaming](#), [setAppInstance](#), [setDebugLog](#), [setDstStreamName](#), [setHost](#), [setHostname](#), [setImplementation](#), [setMapFileIndex](#), [setOriginalTimecodeThreshold](#), [setOriginalTimecodeThreshold](#), [setPassword](#), [setPort](#), [setPort](#), [setReconnectWaitTime](#), [setSendOriginalTimecodes](#), [setSrcStream](#), [setSrcStreamName](#), [setStreamName](#), [setUserName](#)

Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Methods inherited from interface [com.wowza.wms.pushpublish.model.IPushPublish](#)

[connect](#), [createPushPublishSession](#), [disconnect](#), [getContextStr](#), [getDstStreamName](#), [getHostname](#), [getMapFileIndex](#), [getMessagesAttemptedCount](#), [getMessagesFailedCount](#), [getMessagesRetriedCount](#), [getPort](#), [getProperties](#), [getPushPublishType](#), [getReconnectWaitTime](#), [getSrcStream](#), [getSrcStreamName](#), [init](#), [isConnected](#), [resetConnection](#), [setDebugLog](#), [setDstStreamName](#), [setHost](#), [setHostname](#), [setMapFileIndex](#), [setPort](#), [setPort](#), [setReconnectWaitTime](#), [setSrcStream](#), [setSrcStreamName](#), [setStreamName](#)

Fields

CLASS

```
public static final java.lang.Class CLASS
```

CLASSNAME

```
public static final java.lang.String CLASSNAME
```

Constant value: **PushPublishRTMPPProfileHandler**

streamId

```
protected java.lang.String streamId
```

The stream id.

Constructors

PushPublishRTMPPProfileHandler

```
public PushPublishRTMPPProfileHandler()
```

Instantiates a new push publisher rtmp.

(continued from last page)

Throws:`LicensingException` - the licensing exception

Methods

setConnectionTimeout

```
public void setConnectionTimeout(String connectionTimeout)
```

setSendFCPublish

```
public void setSendFCPublish(String sendFCPublish)
```

Sets the send fc publish.

Parameters:

`sendFCPublish` - the new send fc publish

setSendReleaseStream

```
public void setSendReleaseStream(String sendReleaseStream)
```

Sets the send release stream.

Parameters:

`sendReleaseStream` - the new send release stream

setConnectionInfoQueryString

```
public void setConnectionInfoQueryString(String connectionQueryStr)
```

Set the connection query string.

Parameters:

`connectionQueryStr` - connection query string

connect_Out

```
public void connect_Out()
```

Connect to server.

setConnectionFlashVersion

```
public void setConnectionFlashVersion(String connectionFlashVersion)
```

Set Flash version used in connection metadata.

Parameters:

`connectionFlashVersion` - Flash version used in connection metadata

setConnectionInfoSwfURL

```
public void setConnectionInfoSwfURL(String connectionSwfURL)
```

Set SWF URL used in connection metadata.

(continued from last page)

Parameters:

connectionSwfURL - SWF URL used in connection metadata

setConnectionInfoPageURL

```
public void setConnectionInfoPageURL(String connectionPageURL)
```

Set Page URL used in connection metadata.

Parameters:

connectionPageURL - Page URL used in connection metadata

setSecureTokenSharedSecret

```
public void setSecureTokenSharedSecret(String secureTokenSharedSecret)
```

Set SecureToken shared secret.

Parameters:

secureTokenSharedSecret - SecureToken shared secret

setSendFCAnnounce

```
public void setSendFCAnnounce(String sendFCAnnounce)
```

setValidationFrequency

```
public void setValidationFrequency(String validationFrequency)
```

setConnectLastAttempt

```
public void setConnectLastAttempt(String connectLastAttempt)
```

setReconnectWaitTime

```
public void setReconnectWaitTime(String reconnectWaitTime)
```

setConnectLastSuccess

```
public void setConnectLastSuccess(String connectLastSuccess)
```

setSendOriginalTimecodes

```
public void setSendOriginalTimecodes(String sendOriginalTimecodes)
```

Sets the send original timecodes.

(continued from last page)

Parameters:

sendOriginalTimecodes - the new send original timecodes

setSendOnMetadata

```
public void setSendOnMetadata(String sendOnMetadata)
```

setRemoveDefaultAppInstance

```
public void setRemoveDefaultAppInstance(String removeDefaultAppInstance)
```

Sets the removes the default app instance.

Parameters:

removeDefaultAppInstance - the new removes the default app instance

setOnMetadataToSetDataFrame

```
public void setOnMetadataToSetDataFrame(String onMetadataToSetDataFrame)
```

setSendStreamCloseCommands

```
public void setSendStreamCloseCommands(String sendStreamCloseCommands)
```

Sets the send stream close commands.

Parameters:

sendStreamCloseCommands - the new send stream close commands

setStreamCloseWaitTime

```
public void setStreamCloseWaitTime(String streamCloseWaitTime)
```

setOriginalTimecodeThreshold

```
public void setOriginalTimecodeThreshold(String originalTimecodeThreshold)
```

Sets the original timecode threshold.

Parameters:

originalTimecodeThreshold - the new original timecode threshold

setResetOnTimecodeOutOfOrder

```
public void setResetOnTimecodeOutOfOrder(String resetOnTimecodeOutOfOrder)
```

Set to true if you wish push to reset if timecodes jump out of order.

Parameters:

resetOnTimecodeOutOfOrder - true if you wish push to reset if timecodes jump out of order

(continued from last page)

setTimecodesOutOfOrderThreshold

```
public void setTimecodesOutOfOrderThreshold(String timecodesOutOfOrderThreshold)
```

setWaitOnMetadataTimeout

```
public void setWaitOnMetadataTimeout(String waitOnMetadataTimeout)
```

Timeout (milliseconds) that the wait on onMetadata system will wait looking for video and audio data to be populated.

Parameters:

waitOnMetadataTimeout - the new timeout value (milliseconds-string)

setWaitOnMetadataAvailable

```
public void setWaitOnMetadataAvailable(String waitOnMetadataAvailable)
```

setWaitOnMetadataVideo

```
public void setWaitOnMetadataVideo(String waitOnMetadataVideo)
```

setWaitOnMetadataAudio

```
public void setWaitOnMetadataAudio(String waitOnMetadataAudio)
```

setWaitOnMetadataVideoDatarate

```
public void setWaitOnMetadataVideoDatarate(String waitOnMetadataVideoDatarate)
```

setWaitOnMetadataAudioDatarate

```
public void setWaitOnMetadataAudioDatarate(String waitOnMetadataAudioDatarate)
```

setSSL

```
public void setSSL(String sslStr)
```

load

```
public void load(java.util.HashMap dataMap)
```

(continued from last page)

isDebugPackets

```
public boolean isDebugPackets()
```

Checks if is debug packets.

Returns:

true, if is debug packets

setDebugPackets

```
public void setDebugPackets(boolean debugPackets)
```

Sets the debug packets.

Parameters:

debugPackets - the new debug packets

setDebugPackets

```
public void setDebugPackets(String debugPacketsIn)
```

Sets the debug packets.

Parameters:

debugPacketsIn - the new debug packets

isDebugLog

```
public boolean isDebugLog()
```

Is debug logging turned on.

Returns:

true if debug logging is turned on

getDstApplicationName

```
public String getDstApplicationName()
```

Get destination application name.

Returns:

destination application name

setApplication

```
public void setApplication(String dstApplicationNameIN)
```

Set destination application name.

Parameters:

dstApplicationNameIN - destination application name

setDstApplicationName

```
public void setDstApplicationName(String string)
```

(continued from last page)

Sets the dst application name.

Parameters:

string - the new dst application name

getDstAppInstanceName

```
public String getDstAppInstanceName()
```

Get destination appInstance name.

Returns:

destination appInstance name

setLocalBindAddress

```
public void setLocalBindAddress(String bindAddress)
```

Sets the local bind address for the connection.

Parameters:

bindAddress - the local bind address for the connection

getLocalBindAddress

```
public String getLocalBindAddress()
```

Get local bind address for the connection.

Returns:

local bind address for the connection

init

```
public void init(IApplicationInstance appInstance2,  
String streamName,  
IMediaStream stream,  
java.util.HashMap profileData,  
java.util.HashMap maps,  
boolean debug)
```

Init's the instance.

setDstStreamName

```
public void setDstStreamName(String dstStreamNameIn)
```

Set destination stream name

setDstAppInstanceName

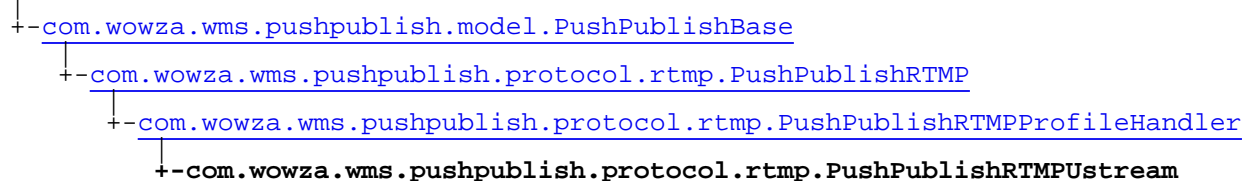
```
public void setDstAppInstanceName(String dstAppInstanceName)
```

Set destination appInstance name

com.wowza.wms.pushpublish.protocol.rtmp

Class PushPublishRTMPUstream

java.lang.Object



All Implemented Interfaces:

[IPushPublish](#)

public class **PushPublishRTMPUstream**
 extends [PushPublishRTMPProfileHandler](#)

Field Summary

public static final	CLASS
public static final	CLASSNAME Value: PushPublishRTMPUstream

Fields inherited from class [com.wowza.wms.pushpublish.protocol.rtmp.PushPublishRTMPProfileHandler](#)

[CLASS](#), [CLASSNAME](#), [streamId](#)

Fields inherited from class [com.wowza.wms.pushpublish.protocol.rtmp.PushPublishRTMP](#)

[akamaiOriginIp](#), [akamaiOriginPort](#), [akamaiPassword](#), [akamaiUserName](#), [bindAddress](#), [bytesAvg](#), [CLASS](#), [CLASSNAME](#), [connected](#), [connectionFlashVersion](#), [connectionPageURL](#), [connectionQueryStr](#), [connectionResetCount](#), [connectionSwfURL](#), [connectionTimeout](#), [connectLastAttempt](#), [connectLastSuccess](#), [connectMetaData](#), [CURRENTFLASHVERSION](#), [CURRENTFMLEVERSION](#), [debugLogLastPacketCount](#), [debugPackets](#), [doSendOnMetaData](#), [dstAppInstanceName](#), [dstApplicationName](#), [dstStreamName](#), [howToPublish](#), [ioSessionToSessionHolder](#), [isReconnectorRunning](#), [isSSL](#), [lastAudioTC](#), [lastDataTC](#), [lastIdleReadTime](#), [lastIdleWriteTime](#), [lastVideoTC](#), [listeners](#), [lock](#), [netconnectToSessionHolder](#), [offlineTime](#), [onlineTime](#), [onMetadataToSetDataFrame](#), [outprint](#), [packetsSentAudio](#), [packetsSentData](#), [packetsSentVideo](#), [removeDefaultAppInstance](#), [resetOnTimecodeOutOfOrder](#), [rtmpAuthProvider](#), [rtmpDebugLog](#), [secureTokenSharedSecret](#), [sendFCAnnounce](#), [sendFCPublish](#), [sendOnMetaData](#), [sendOriginalTimecodes](#), [sendReleaseStream](#), [sendStreamCloseCommands](#), [sendStreamCloseCommandsSendTime](#), [sessionOrder](#), [srcStreamBufferSize](#), [streamCloseWaitTime](#), [syncPacketType](#), [syncTimecode](#), [timecodesOutOfOrderThreshold](#), [totalPacketsSent](#), [tryConnect](#), [validationFrequency](#), [waitOnMetadataAudio](#), [waitOnMetadataAudioDatarate](#), [waitOnMetadataAvailable](#), [waitOnMetadataReady](#), [waitOnMetadataStartime](#), [waitOnMetadataTimeout](#), [waitOnMetadataVideo](#), [waitOnMetadataVideoDatarate](#)

Fields inherited from class [com.wowza.wms.pushpublish.model.PushPublishBase](#)

[appInstance](#), [CLASS](#), [CLASSNAME](#), [dstStreamName](#), [hostname](#), [inetAddr](#), [mapFileIndex](#), [mImplementation](#), [mType](#), [originalTimecodeThreshold](#), [password](#), [port](#), [properties](#), [pushPublishStreamDebug](#), [pushPublishType](#), [reconnectWaitTime](#), [sendingPerformance](#), [sendOriginalTimecodes](#), [srcStream](#), [srcStreamName](#), [userName](#)

Fields inherited from interface [com.wowza.wms.pushpublish.model.IPushPublish](#)

[PUSHPUBLISH_TYPE_RTMP](#), [PUSHPUBLISH_TYPE_RTP](#), [PUSHPUBLISH_TYPE_UNKNOWN](#)

Constructor Summary

public	PushPublishRTMPUstream()
--------	--

Method Summary

void	callConnect (PushPublishRTMPNetConnectionSession pushPublisherSession, String addQueryStr)
void	init (IApplicationInstance appInstance2, String streamName, IMediaStream stream, java.util.HashMap profileData, java.util.HashMap maps, boolean debug)
org.apache.mina.common.io.Session	internalConnect (PushPublishRTMPNetConnectionSession pushPublisherSession)
boolean	isUseUstreamPingPong ()
void	load (java.util.HashMap dataMap)
void	setUseUstreamPingPong (boolean useUstreamPingPong)

Methods inherited from class

[com.wowza.wms.pushpublish.protocol.rtmp.PushPublishRTMPProfileHandler](#)

[connect_Out](#), [getDstAppInstanceName](#), [getDstApplicationName](#), [getLocalBindAddress](#), [init](#), [isDebugLog](#), [isDebugPackets](#), [load](#), [setApplication](#), [setConnectionFlashVersion](#), [setConnectionInfoPageURL](#), [setConnectionInfoQueryString](#), [setConnectionInfoSwfURL](#), [setConnectionTimeout](#), [setConnectLastAttempt](#), [setConnectLastSuccess](#), [setDebugPackets](#), [setDebugPackets](#), [setDstAppInstanceName](#), [setDstApplicationName](#), [setDstStreamName](#), [setLocalBindAddress](#), [setOnMetadataToSetDataFrame](#), [setOriginalTimecodeThreshold](#), [setReconnectWaitTime](#), [setRemoveDefaultAppInstance](#), [setResetOnTimecodeOutOfOrder](#), [setSecureTokenSharedSecret](#), [setSendFCAnnounce](#), [setSendFCPublish](#), [setSendOnMetadata](#), [setSendOriginalTimecodes](#), [setSendReleaseStream](#), [setSendStreamCloseCommands](#), [setSSL](#), [setStreamCloseWaitTime](#), [setTimecodesOutOfOrderThreshold](#), [setValidationFrequency](#), [setWaitOnMetadataAudio](#), [setWaitOnMetadataAudioDataRate](#), [setWaitOnMetadataAvailable](#), [setWaitOnMetadataTimeout](#), [setWaitOnMetadataVideo](#), [setWaitOnMetadataVideoDataRate](#)

Methods inherited from class [com.wowza.wms.pushpublish.protocol.rtmp.PushPublishRTMP](#)

[addConnectMetaData](#), [addListener](#), [addSession](#), [bumpReconnectWaitTime](#), [callConnect](#),
[connect](#), [convertDataEvenToAMFData](#), [convertOnMetadataToSetDataFrame](#),
[createPushPublishSession](#), [disconnect](#), [getAkamaiOriginIp](#), [getAkamaiOriginPort](#),
[getAkamaiPassword](#), [getAkamaiSessionKey](#), [getAkamaiUserName](#), [getAppInstance](#),
[getBindAddress](#), [getConnectionFlashVerion](#), [getConnectionFlashVersion](#),
[getConnectionPageURL](#), [getConnectionQueryStr](#), [getConnectionSwfURL](#),
[getConnectionTimeout](#), [getConnectLastAttempt](#), [getConnectLastSuccess](#), [getContextStr](#),
[getDstAppInstanceName](#), [getDstApplicationName](#), [getDstStreamName](#), [getHostname](#),
[getHowToPublish](#), [getLastAudioTC](#), [getLastDataTC](#), [getLastSession](#), [getLastVideoTC](#),
[getLocalListeners](#), [getLock](#), [getMessagesAttemptedCount](#), [getMessagesFailedCount](#),
[getMessagesRetriedCount](#), [getNetConnection](#), [getOriginalTimecodeThreshold](#),
[getPacketsSentAudio](#), [getPacketsSentData](#), [getPacketsSentVideo](#), [getPort](#),
[getReconnectDelay](#), [getRTMPAuthProvider](#), [getSecureTokenSharedSecret](#),
[getSessionHolderByNetConnection](#), [getSessionHolderBySession](#), [getSessionListDebug](#),
[getSrcStreamBufferSize](#), [getSrcStreamName](#), [getStreamCloseWaitTime](#), [getSyncPacketType](#),
[getSyncTimecode](#), [getTimecodesOutOfOrderThreshold](#), [getTotalPacketsSent](#),
[getValidationFrequency](#), [getWaitOnMetadataTimeout](#), [internalConnect](#), [isConnected](#),
[isCurrentSession](#), [isDebugLog](#), [isDebugPackets](#), [isFlashVerionFMLE](#), [isFlashVersionFMLE](#),
[isOnMetadataToSetDataFrame](#), [isRemoveDefaultAppInstance](#), [isResetOnTimecodeOutOfOrder](#),
[isSendFCAnnounce](#), [isSendFCPublish](#), [isSendOnMetadata](#), [isSendOriginalTimecodes](#),
[isSendReleaseStream](#), [isSendStreamCloseCommands](#), [isSSL](#), [isWaitOnMetadataAudio](#),
[isWaitOnMetadataAudioDatarate](#), [isWaitOnMetadataAvailable](#), [isWaitOnMetadataVideo](#),
[isWaitOnMetadataVideoDatarate](#), [notifyAkamaiClientLogin](#), [notifyAkamaiSetChallenge](#),
[notifyAkamaiSetOriginConnectionInfo](#), [notifyConnect](#), [notifyConnectFailure](#),
[notifyConnectStart](#), [notifyConnectSuccess](#), [notifyFCAnnounce](#), [notifyFCPublish](#),
[notifyHandshakeResult](#), [notifyPublishHandlerPlay](#), [notifyPushPublisherSessionCreate](#),
[notifyPushPublisherSessionDestroy](#), [notifySessionClosed](#), [notifySessionIdle](#),
[notifySessionOpened](#), [notifyStreamCreate](#), [notifyStreamOnPlayStatus](#),
[notifyStreamOnStatus](#), [notifyValidateSession](#), [notifyValidateSessionResult](#),
[removeListener](#), [removeSession](#), [resetConnection](#), [resetConnection](#), [resolveHostName](#),
[sessionClosed](#), [sessionOpened](#), [setAkamaiOriginIp](#), [setAkamaiOriginPort](#),
[setAkamaiPassword](#), [setAkamaiUserName](#), [setAppInstance](#), [setBindAddress](#),
[setConnectionFlashVerion](#), [setConnectionFlashVersion](#), [setConnectionPageURL](#),
[setConnectionQueryStr](#), [setConnectionSwfURL](#), [setConnectionTimeout](#),
[setConnectLastAttempt](#), [setConnectLastSuccess](#), [setDebugLog](#), [setDebugPackets](#),
[setDstAppInstanceName](#), [setDstApplicationName](#), [setDstStreamName](#), [setHostname](#),
[setHowToPublish](#), [setOnMetadataToSetDataFrame](#), [setOriginalTimecodeThreshold](#), [setPort](#),
[setRemoveDefaultAppInstance](#), [setResetOnTimecodeOutOfOrder](#), [setRTMPAuthProvider](#),
[setSecureTokenSharedSecret](#), [setSendFCAnnounce](#), [setSendFCPublish](#), [setSendOnMetadata](#),
[setSendOriginalTimecodes](#), [setSendReleaseStream](#), [setSendStreamCloseCommands](#),
[setSrcStreamBufferSize](#), [setSrcStreamName](#), [setSSL](#), [setStreamCloseWaitTime](#),
[setSyncPacketType](#), [setSyncTimecode](#), [setTimecodesOutOfOrderThreshold](#),
[setValidationFrequency](#), [setWaitOnMetadataAudio](#), [setWaitOnMetadataAudioDatarate](#),
[setWaitOnMetadataAvailable](#), [setWaitOnMetadataTimeout](#), [setWaitOnMetadataVideo](#),
[setWaitOnMetadataVideoDatarate](#), [tryToConnect](#)

Methods inherited from class [com.wowza.wms.pushpublish.model.PushPublishBase](#)

[createPushPublishSession](#), [getAppInstance](#), [getBandwidth](#), [getConfigDataOrDefault](#), [getConfigDataOrDefault](#), [getConfigDataOrDefault](#), [getContextStr](#), [getDstStreamName](#), [getHostname](#), [getImplementation](#), [getMapFileIndex](#), [getNumberFromString](#), [getOriginalTimecodeThreshold](#), [getPassword](#), [getPort](#), [getProperties](#), [getPushPublishType](#), [getPushPublishVersionStr](#), [getReconnectWaitTime](#), [getSrcStream](#), [getSrcStreamName](#), [getUserName](#), [init](#), [isAdaptiveStreaming](#), [isSendOriginalTimecodes](#), [load](#), [setAdaptiveStreaming](#), [setAppInstance](#), [setDebugLog](#), [setDstStreamName](#), [setHost](#), [setHostname](#), [setImplementation](#), [setMapFileIndex](#), [setOriginalTimecodeThreshold](#), [setOriginalTimecodeThreshold](#), [setPassword](#), [setPort](#), [setPort](#), [setReconnectWaitTime](#), [setSendOriginalTimecodes](#), [setSrcStream](#), [setSrcStreamName](#), [setStreamName](#), [setUserName](#)

Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Methods inherited from interface [com.wowza.wms.pushpublish.model.IPushPublish](#)

[connect](#), [createPushPublishSession](#), [disconnect](#), [getContextStr](#), [getDstStreamName](#), [getHostname](#), [getMapFileIndex](#), [getMessagesAttemptedCount](#), [getMessagesFailedCount](#), [getMessagesRetriedCount](#), [getPort](#), [getProperties](#), [getPushPublishType](#), [getReconnectWaitTime](#), [getSrcStream](#), [getSrcStreamName](#), [init](#), [isConnected](#), [resetConnection](#), [setDebugLog](#), [setDstStreamName](#), [setHost](#), [setHostname](#), [setMapFileIndex](#), [setPort](#), [setPort](#), [setReconnectWaitTime](#), [setSrcStream](#), [setSrcStreamName](#), [setStreamName](#)

Fields

CLASS

```
public static final java.lang.Class CLASS
```

CLASSNAME

```
public static final java.lang.String CLASSNAME
```

Constant value: **PushPublishRTMPUstream**

Constructors

PushPublishRTMPUstream

```
public PushPublishRTMPUstream()
```

Methods

isUseUstreamPingPong

```
public boolean isUseUstreamPingPong()
```

(continued from last page)

setUseUstreamPingPong

```
public void setUseUstreamPingPong(boolean useUstreamPingPong)
```

init

```
public void init(IApplicationInstance appInstance2,  
                String streamName,  
                IMediaStream stream,  
                java.util.HashMap profileData,  
                java.util.HashMap maps,  
                boolean debug)
```

Initiates the instance.

load

```
public void load(java.util.HashMap dataMap)
```

internalConnect

```
protected org.apache.mina.common.IoSession  
internalConnect(PushPublishRTMPNetConnectionSession pushPublisherSession)
```

Internal connect method

callConnect

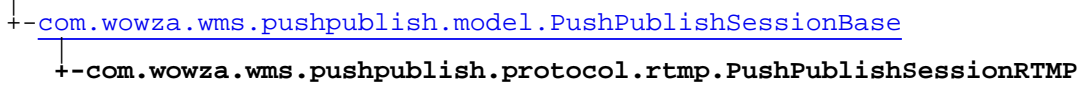
```
protected void callConnect(PushPublishRTMPNetConnectionSession pushPublisherSession,  
                           String addQueryStr)
```

Internal connect

com.wowza.wms.pushpublish.protocol.rtmp

Class PushPublishSessionRTMP

java.lang.Object



All Implemented Interfaces:
[IPushPublishSession](#)

public class **PushPublishSessionRTMP**
 extends [PushPublishSessionBase](#)

Fields inherited from class [com.wowza.wms.pushpublish.model.PushPublishSessionBase](#)

[profile](#), [pushPublishImpl](#), [stream](#), [type](#)

Fields inherited from interface [com.wowza.wms.pushpublish.model.IPushPublishSession](#)

[PUSHPUBLISHSESSION_TYPE_RTMP](#), [PUSHPUBLISHSESSION_TYPE_RTP](#),
[PUSHPUBLISHSESSION_TYPE_UNKNOWN](#)

Constructor Summary

public	PushPublishSessionRTMP()
--------	--

Methods inherited from class [com.wowza.wms.pushpublish.model.PushPublishSessionBase](#)

[getContextStr](#), [getProfile](#), [getPushPublishImpl](#), [getStream](#), [getType](#), [setProfile](#),
[setPushPublishImpl](#), [setStream](#)

Methods inherited from class java.lang.Object

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#),
[wait](#)

Methods inherited from interface [com.wowza.wms.pushpublish.model.IPushPublishSession](#)

[getContextStr](#), [getProfile](#), [getPushPublishImpl](#), [getStream](#), [getType](#), [setProfile](#),
[setPushPublishImpl](#), [setStream](#)

Constructors

PushPublishSessionRTMP

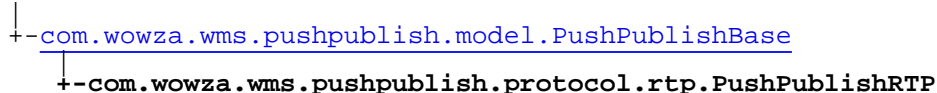
public **PushPublishSessionRTMP**()

Package

com.wowza.wms.pushpublish.protocol.rtp

com.wowza.wms.pushpublish.protocol.rtp Class PushPublishRTP

java.lang.Object



All Implemented Interfaces:

[IPushPublish](#)

public class **PushPublishRTP**
extends [PushPublishBase](#)

The Class PushPublisherRTP.

Field Summary

protected	audioHost The audio host.
protected	audioPort The audio port.
public static final	CLASS
public static final	CLASSNAME Value: PushPublishRTP
protected	isRTPWrapped The is rtp wrapped.
protected	lock
protected	mRtpDestination The m rtp destination.
protected	removeSDPFiles The remove sdp files
protected	rtpPushPublishSession The rtp push publish session.
public static final	SDPFILEEXTENSION The Constant SDPFILEEXTENSION. Value: .sdp
protected	singleStreamAV The stream packetizer.
protected	videoHost The video host.

protected	videoPort The video port.
protected	writeSDPFiles The write sdp files.

Fields inherited from class [com.wowza.wms.pushpublish.model.PushPublishBase](#)

[appInstance](#), [CLASS](#), [CLASSNAME](#), [dstStreamName](#), [hostname](#), [inetAddr](#), [mapFileIndex](#), [mImplementation](#), [mType](#), [originalTimecodeThreshold](#), [password](#), [port](#), [properties](#), [pushPublishStreamDebug](#), [pushPublishType](#), [reconnectWaitTime](#), [sendingPerformance](#), [sendOriginalTimecodes](#), [srcStream](#), [srcStreamName](#), [userName](#)

Fields inherited from interface [com.wowza.wms.pushpublish.model.IPushPublish](#)

[PUSHPUBLISH_TYPE_RTMP](#), [PUSHPUBLISH_TYPE_RTP](#), [PUSHPUBLISH_TYPE_UNKNOWN](#)

Constructor Summary

public	PushPublishRTP()
--------	----------------------------------

Method Summary

void	connect()
IPushPublishSession	createPushPublishSession()
void	disconnect()
IApplicationInstance	getAppInstance()
String	getAudioHost() Gets the audio host.
int	getAudioPort() Gets the audio port.
String	getContextStr()
String	getHostname()
String	getHostType() Gets the host type.
long	getMessagesAttemptedCount()
long	getMessagesFailedCount()
long	getMessagesRetriedCount()
RTPDestination	getmRtpDestination()

<u>RTPPushPublishSession</u>	<u>getRTPPushPublishSession()</u> Gets the rTP push publish session.
long	<u>getRtpStreamWaitTimeout()</u>
String	<u>getSrcStreamName()</u>
int	<u>getTimeToLive()</u> Gets the time to live.
String	<u>getVideoHost()</u> Gets the video host.
int	<u>getVideoPort()</u> Gets the video port.
void	<u>init()</u> (<u>IApplicationInstance</u> appInstance, String streamName, <u>IMediaStream</u> stream, java.util.HashMap profileData, java.util.HashMap maps, boolean debug)
Boolean	<u>isConnected()</u>
boolean	<u>isRTPWrapped()</u> Checks if is rTP wrapped.
boolean	<u>isSingleStreamAV()</u> Checks if is stream is packetized.
void	<u>load()</u> (java.util.HashMap dataMap)
void	<u>resetConnection()</u>
void	<u>resetConnection()</u> (int delay)
void	<u>setAppInstance()</u> (<u>IApplicationInstance</u> appInstance)
void	<u>setAudioHost()</u> (String audioHost) Sets the audio host.
void	<u>setAudioPort()</u> (int audioPort) Sets the audio port.
void	<u>setAudioPort()</u> (String audioPort) Sets the audio port.
void	<u>setConnectionStatus()</u> (Boolean connected)
void	<u>setHost()</u> (String hostname)
void	<u>setHostType()</u> (String hostType) Sets the host type.
void	<u>setPort()</u> (int port) Set destination server port.

void	<u>setPort</u> (String portIn)
void	<u>setRTPPushPublishSession</u> (<u>RTPPushPublishSession</u> rtpPushPublishSession) Sets the rTP push publish session.
void	<u>setRtpStreamWaitTimeout</u> (long rtpStreamWaitTimeout)
void	<u>setRtpStreamWaitTimeout</u> (String rtpStreamWaitTimeout)
void	<u>setRTPWrapped</u> (String string) Sets the rTP wrapped.
void	<u>setSingleStreamAV</u> (boolean inSingleStreamAV) Sets the stream packetizer.
void	<u>setSingleStreamAV</u> (String inSingleStreamAV)
void	<u>setSrcStreamName</u> (String srcStreamName)
void	<u>setTimeToLive</u> (String ttl) Sets the time to live.
void	<u>setVideoHost</u> (String videoHost) Sets the video host.
void	<u>setVideoPort</u> (String videoPort) Sets the video port.

Methods inherited from class [com.wowza.wms.pushpublish.model.PushPublishBase](#)

[createPushPublishSession](#), [getAppInstance](#), [getBandwidth](#), [getConfigDataOrDefault](#), [getConfigDataOrDefault](#), [getContextStr](#), [getDstStreamName](#), [getHostname](#), [getImplementation](#), [getMapFileIndex](#), [getNumberFromString](#), [getOriginalTimecodeThreshold](#), [getPassword](#), [getPort](#), [getProperties](#), [getPushPublishType](#), [getPushPublishVersionStr](#), [getReconnectWaitTime](#), [getSrcStream](#), [getSrcStreamName](#), [getUserName](#), [init](#), [isAdaptiveStreaming](#), [isSendOriginalTimecodes](#), [load](#), [setAdaptiveStreaming](#), [setAppInstance](#), [setDebugLog](#), [setDstStreamName](#), [setHost](#), [setHostname](#), [setImplementation](#), [setMapFileIndex](#), [setOriginalTimecodeThreshold](#), [setOriginalTimecodeThreshold](#), [setPassword](#), [setPort](#), [setPort](#), [setReconnectWaitTime](#), [setSendOriginalTimecodes](#), [setSrcStream](#), [setSrcStreamName](#), [setStreamName](#), [setUserName](#)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface [com.wowza.wms.pushpublish.model.IPushPublish](#)

[connect](#), [createPushPublishSession](#), [disconnect](#), [getContextStr](#), [getDstStreamName](#), [getHostname](#), [getMapFileIndex](#), [getMessagesAttemptedCount](#), [getMessagesFailedCount](#), [getMessagesRetriedCount](#), [getPort](#), [getProperties](#), [getPushPublishType](#), [getReconnectWaitTime](#), [getSrcStream](#), [getSrcStreamName](#), [init](#), [isConnected](#), [resetConnection](#), [setDebugLog](#), [setDstStreamName](#), [setHost](#), [setHostname](#), [setMapFileIndex](#), [setPort](#), [setPort](#), [setReconnectWaitTime](#), [setSrcStream](#), [setSrcStreamName](#), [setStreamName](#)

Fields

CLASS

```
public static final java.lang.Class CLASS
```

CLASSNAME

```
public static final java.lang.String CLASSNAME
```

Constant value: **PushPublishRTP**

SDPFILEEXTENSION

```
public static final java.lang.String SDPFILEEXTENSION
```

The Constant SDPFILEEXTENSION.

Constant value: **.sdp**

rtpPushPublishSession

```
protected com.wowza.wms.rtp.model.RTPPushPublishSession rtpPushPublishSession
```

The rtp push publish session.

singleStreamAV

```
protected boolean singleStreamAV
```

The stream packetizer.

audioPort

```
protected int audioPort
```

The audio port.

videoPort

```
protected int videoPort
```

The video port.

audioHost

```
protected java.lang.String audioHost
```

The audio host.

videoHost

```
protected java.lang.String videoHost
```

(continued from last page)

The video host.

isRTPWrapped

protected boolean **isRTPWrapped**

The is rtp wrapped.

writeSDPFiles

protected boolean **writeSDPFiles**

The write sdp files.

removeSDPFiles

protected boolean **removeSDPFiles**

The remove sdp files

mRtpDestination

protected com.wowza.wms.rtp.model.RTPDestination **mRtpDestination**

The m rtp destination.

lock

protected java.lang.Object **lock**

Constructors

PushPublishRTP

```
public PushPublishRTP()
```

Methods

init

```
public void init(IApplicationInstance appInstance,  
                String streamName,  
                IMediaStream stream,  
                java.util.HashMap profileData,  
                java.util.HashMap maps,  
                boolean debug)
```

Inits the instance.

connect

```
public void connect()
```

disconnect

```
public void disconnect()
```

resetConnection

```
public void resetConnection()
```

resetConnection

```
public void resetConnection(int delay)
```

getRTTPushPublishSession

```
public RTTPushPublishSession getRTTPushPublishSession()
```

Gets the rTP push publish session.

Returns:

the rTP push publish session

setRTTPushPublishSession

```
public void setRTTPushPublishSession(RTTPushPublishSession rtpPushPublishSession)
```

Sets the rTP push publish session.

Parameters:

rtpPushPublishSession - the new rTP push publish session

getAppInstance

```
public IApplicationInstance getAppInstance()
```

Get appInstance interface.

setAppInstance

```
public void setAppInstance(IApplicationInstance appInstance)
```

Set appInstance interface.

getSrcStreamName

```
public String getSrcStreamName()
```

Get source stream name.

setSrcStreamName

```
public void setSrcStreamName(String srcStreamName)
```

(continued from last page)

Set source stream name.

isSingleStreamAV

```
public boolean isSingleStreamAV()
```

Checks if is stream is packetized.

Returns:

true, if is stream packetizer

setSingleStreamAV

```
public void setSingleStreamAV(boolean inSingleStreamAV)
```

Sets the stream packetizer.

Parameters:

inSingleStreamAV - whether stream is packetized

setSingleStreamAV

```
public void setSingleStreamAV(String inSingleStreamAV)
```

getHostname

```
public String getHostname()
```

Get destination server host name.

setHost

```
public void setHost(String hostname)
```

Set destination server host name.

setPort

```
public void setPort(int port)
```

Set destination server port.

Parameters:

port - destination server port

setPort

```
public void setPort(String portIn)
```

getAudioPort

```
public int getAudioPort()
```

Gets the audio port.

(continued from last page)

Returns:

the audio port

setAudioPort

```
public void setAudioPort(int audioPort)
```

Sets the audio port.

Parameters:

audioPort - the new audio port

getAudioHost

```
public String getAudioHost()
```

Gets the audio host.

Returns:

the audio host

setAudioHost

```
public void setAudioHost(String audioHost)
```

Sets the audio host.

Parameters:

audioHost - the new audio host

getVideoHost

```
public String getVideoHost()
```

Gets the video host.

Returns:

the video host

setVideoHost

```
public void setVideoHost(String videoHost)
```

Sets the video host.

Parameters:

videoHost - the new video host

setAudioPort

```
public void setAudioPort(String audioPort)
```

Sets the audio port.

Parameters:

audioPort - the new audio port

getVideoPort

```
public int getVideoPort()
```

Gets the video port.

Returns:

the video port

setVideoPort

```
public void setVideoPort(String videoPort)
```

Sets the video port.

Parameters:

videoPort - the new video port

isRTPWrapped

```
public boolean isRTPWrapped()
```

Checks if is rTP wrapped.

Returns:

true, if is rTP wrapped

setRTPWrapped

```
public void setRTPWrapped(String string)
```

Sets the rTP wrapped.

Parameters:

string - the new rTP wrapped

getHostType

```
public String getHostType()
```

Gets the host type.

Returns:

the host type

setHostType

```
public void setHostType(String hostType)
```

Sets the host type.

Parameters:

hostType - the new host type

setTimeToLive

```
public void setTimeToLive(String ttl)
```

(continued from last page)

Sets the time to live.

Parameters:

ttl - the new time to live

getTimeToLive

```
public int getTimeToLive()
```

Gets the time to live.

Returns:

the time to live

load

```
public void load(java.util.HashMap dataMap)
```

getRtpStreamWaitTimeout

```
public long getRtpStreamWaitTimeout()
```

setRtpStreamWaitTimeout

```
public void setRtpStreamWaitTimeout(String rtpStreamWaitTimeout)
```

setRtpStreamWaitTimeout

```
public void setRtpStreamWaitTimeout(long rtpStreamWaitTimeout)
```

createPushPublishSession

```
public IPushPublishSession createPushPublishSession()
```

getContextStr

```
public String getContextStr()
```

getmRtpDestination

```
public RTPDestination getmRtpDestination()
```

(continued from last page)

isConnected

```
public Boolean isConnected()
```

setConnectionStatus

```
public void setConnectionStatus(Boolean connected)
```

getMessagesAttemptedCount

```
public long getMessagesAttemptedCount()
```

getMessagesRetriedCount

```
public long getMessagesRetriedCount()
```

getMessagesFailedCount

```
public long getMessagesFailedCount()
```

com.wowza.wms.pushpublish.protocol.rtp Class PushPublishRTPRunner

java.lang.Object

└- java.lang.Thread

└- com.wowza.wms.pushpublish.protocol.rtp.PushPublishRTPRunner

All Implemented Interfaces:

Runnable

public class **PushPublishRTPRunner**
extends Thread

The Class RTPStarter. Utility class to starts an RTP session for push publishing

Field Summary

public static final	CLASS
public static final	CLASSNAME Value: PushPublishRTPRunner

Fields inherited from class java.lang.Thread

MAX_PRIORITY, MIN_PRIORITY, NORM_PRIORITY

Constructor Summary

public	PushPublishRTPRunner (PushPublishRTP pushPublisherRTP, RTPDestination rtpDestination) Instantiates a new rTP PushPublishRTPRunner.
--------	--

Method Summary

void	disconnect ()
void	run ()

Methods inherited from class java.lang.Thread

activeCount, checkAccess, clone, countStackFrames, currentThread, destroy, dumpStack, enumerate, getAllStackTraces, getContextClassLoader, getDefaultUncaughtExceptionHandler, getId, getName, getPriority, getStackTrace, getState, getThreadGroup, getUncaughtExceptionHandler, holdsLock, interrupt, interrupted, isAlive, isDaemon, isInterrupted, join, join, join, resume, run, setContextClassLoader, setDaemon, setDefaultUncaughtExceptionHandler, setName, setPriority, setUncaughtExceptionHandler, sleep, sleep, start, stop, stop, suspend, toString, yield

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

Methods inherited from interface `java.lang.Runnable`

```
run
```

Fields

CLASS

```
public static final java.lang.Class CLASS
```

CLASSNAME

```
public static final java.lang.String CLASSNAME
```

Constant value: **PushPublishRTPRunner**

Constructors

PushPublishRTPRunner

```
public PushPublishRTPRunner(PushPublishRTP pushPublisherRTP,  
                             RTPDestination rtpDestination)
```

Instantiates a new rTP PushPublishRTPRunner.

Parameters:

`pushPublisherRTP` - the push publisher rtp

`rtpDestination` - the rtp destination

Methods

disconnect

```
public void disconnect()
```

run

```
public void run()
```

com.wowza.wms.pushpublish.protocol.rtp Class PushPublishRTSPActionNotify

java.lang.Object

└-com.wowza.wms.pushpublish.protocol.rtp.PushPublishRTSPActionNotify

All Implemented Interfaces:

[IRTSPActionNotify](#)

```
public class PushPublishRTSPActionNotify
extends Object
implements IRTSPActionNotify
```

The Class PushPublishRTSPActionNotify.

Constructor Summary

public	PushPublishRTSPActionNotify (com.wowza.wms.pushpublish.module.ModulePushPublish modulePushPublish) Instantiates a new push publish rtsp action notify.
--------	---

Method Summary

void	onAnnounce (RTPSession rtspSession, com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp)
void	onDescribe (RTPSession rtspSession, com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp)
void	onGetParameter (RTPSession rtspSession, com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp)
void	onOptions (RTPSession rtspSession, com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp)
void	onPause (RTPSession rtspSession, com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp)
void	onPlay (RTPSession rtspSession, com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp)
void	onRecord (RTPSession rtspSession, com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp)
void	onRedirect (RTPSession rtspSession, com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp)

void	onSetParameter (RTPSession rtspSession, com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp)
void	onSetup (RTPSession rtspSession, com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp)
void	onTeardown (RTPSession rtspSession, com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp)

Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Methods inherited from interface [com.wowza.wms.rtp.model.IRTSPActionNotify](#)

[onAnnounce](#), [onDescribe](#), [onGetParameter](#), [onOptions](#), [onPause](#), [onPlay](#), [onRecord](#), [onRedirect](#), [onSetParameter](#), [onSetup](#), [onTeardown](#)

Constructors

PushPublishRTSPActionNotify

```
public PushPublishRTSPActionNotify(com.wowza.wms.pushpublish.module.ModulePushPublish modulePushPublish)
```

Instantiates a new push publish rtsp action notify.

Parameters:

`modulePushPublish` - the module push publish

Methods

onDescribe

```
public void onDescribe(RTPSession rtspSession,
    com.wowza.wms.rtsp.RTSPRequestMessage req,
    com.wowza.wms.rtsp.RTSPResponseMessages resp)
```

onAnnounce

```
public void onAnnounce(RTPSession rtspSession,
    com.wowza.wms.rtsp.RTSPRequestMessage req,
    com.wowza.wms.rtsp.RTSPResponseMessages resp)
```

onGetParameter

```
public void onGetParameter(RTPSession rtspSession,
    com.wowza.wms.rtsp.RTSPRequestMessage req,
    com.wowza.wms.rtsp.RTSPResponseMessages resp)
```


(continued from last page)

onSetParameter

```
public void onSetParameter(RTPSession rtspSession,  
    com.wowza.wms.rtsp.RTSPRequestMessage req,  
    com.wowza.wms.rtsp.RTSPResponseMessages resp)
```

onOptions

```
public void onOptions(RTPSession rtspSession,  
    com.wowza.wms.rtsp.RTSPRequestMessage req,  
    com.wowza.wms.rtsp.RTSPResponseMessages resp)
```

onPause

```
public void onPause(RTPSession rtspSession,  
    com.wowza.wms.rtsp.RTSPRequestMessage req,  
    com.wowza.wms.rtsp.RTSPResponseMessages resp)
```

onPlay

```
public void onPlay(RTPSession rtspSession,  
    com.wowza.wms.rtsp.RTSPRequestMessage req,  
    com.wowza.wms.rtsp.RTSPResponseMessages resp)
```

onRecord

```
public void onRecord(RTPSession rtspSession,  
    com.wowza.wms.rtsp.RTSPRequestMessage req,  
    com.wowza.wms.rtsp.RTSPResponseMessages resp)
```

onRedirect

```
public void onRedirect(RTPSession rtspSession,  
    com.wowza.wms.rtsp.RTSPRequestMessage req,  
    com.wowza.wms.rtsp.RTSPResponseMessages resp)
```

onSetup

```
public void onSetup(RTPSession rtspSession,  
    com.wowza.wms.rtsp.RTSPRequestMessage req,  
    com.wowza.wms.rtsp.RTSPResponseMessages resp)
```

(continued from last page)

onTeardown

```
public void onTeardown(RTPSession rtspSession,  
    com.wowza.wms.rtsp.RTSPRequestMessage req,  
    com.wowza.wms.rtsp.RTSPResponseMessages resp)
```

com.wowza.wms.pushpublish.protocol.rtp Class PushPublishSessionRTP

java.lang.Object

```

  +-- com.wowza.wms.pushpublish.model.PushPublishSessionBase
      +-- com.wowza.wms.pushpublish.protocol.rtp.PushPublishSessionRTP
  
```

All Implemented Interfaces:
[IPushPublishSession](#)

```

public class PushPublishSessionRTP
extends PushPublishSessionBase
  
```

Fields inherited from class [com.wowza.wms.pushpublish.model.PushPublishSessionBase](#)

[profile](#), [pushPublishImpl](#), [stream](#), [type](#)

Fields inherited from interface [com.wowza.wms.pushpublish.model.IPushPublishSession](#)

[PUSHPUBLISHSESSION_TYPE_RTMP](#), [PUSHPUBLISHSESSION_TYPE_RTP](#),
[PUSHPUBLISHSESSION_TYPE_UNKNOWN](#)

Constructor Summary

public	PushPublishSessionRTP()
--------	---

Methods inherited from class [com.wowza.wms.pushpublish.model.PushPublishSessionBase](#)

[getContextStr](#), [getProfile](#), [getPushPublishImpl](#), [getStream](#), [getType](#), [setProfile](#),
[setPushPublishImpl](#), [setStream](#)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait,
wait

Methods inherited from interface [com.wowza.wms.pushpublish.model.IPushPublishSession](#)

[getContextStr](#), [getProfile](#), [getPushPublishImpl](#), [getStream](#), [getType](#), [setProfile](#),
[setPushPublishImpl](#), [setStream](#)

Constructors

PushPublishSessionRTP

```
public PushPublishSessionRTP()
```

Package

com.wowza.wms.response

com.wowza.wms.response Class ResponseFunction

java.lang.Object

└─com.wowza.wms.response.ResponseFunction

public class **ResponseFunction**
extends Object

ResponseFunction: class for sending status messages to the client.

Field Summary

public	doBreak
--------	-------------------------

Constructor Summary

public	ResponseFunction (IClient client) Create an empty ResponseFunction
public	ResponseFunction (com.wowza.wms.netconnection.INetConnection netConnection) Create an empty ResponseFunction
public	ResponseFunction (com.wowza.wms.netconnection.INetConnection netConnection, AMFObj amfObj) Create an empty ResponseFunction
public	ResponseFunction (IClient client, AMFObj amfObj) Create an empty ResponseFunction
public	ResponseFunction (IMediaStream stream, AMFObj amfObj) Create an empty ResponseFunction

Method Summary

void	addBody (AMFData body) Add AMFData to the body of the message.
void	addBytes (byte[] inbytes) Add raw AMFData bytes[] to message body
void	createBroadcastMessage (java.nio.ByteBuffer messageBytes) Create broadcast message.
void	createConnectMessage (String inName, double inResultNum) Create net connection connect message.
void	createDefaultMessage (String inName, double inResultNum) Create a default message (onStatus type messages).
void	createEnhancedSeekMessage (byte[] messageBuffer, int src, int tc)

void	<u>createPlayStatusMessage</u> (String inName) Create onPlayStatus message.
void	<u>createSeekMessage</u> (String inName) Create a seek result message.
void	<u>createSOMessage</u> (byte[] messageBuffer, int objectEncoding) Create shared object message.
long	<u>getTimecode</u> () Get function timecode (milliseconds).
int	<u>getType</u> () Get message type.
boolean	<u>isForceAMF0</u> ()
void	<u>setBody</u> (int index, <u>AMFData</u> body) Add AMFData to the body of the message.
void	<u>setForceAMF0</u> (boolean forceAMF0)
void	<u>setMessageBytes</u> (byte[] messageBytes)
void	<u>setRetAMFNumber</u> (int innum) Set the return channel id
void	<u>setSrc</u> (int src) Set the src (stream id) for the message.
void	<u>setTimecode</u> (long timecode) Set function timecode (milliseconds).
void	<u>setType</u> (int type) Set message type.
int	<u>write</u> (java.io.OutputStream out, boolean isAbsTimecode, int chunkSize) Write message directly to OutputStream
int	<u>write</u> (java.io.OutputStream out, int chunkSize) Write message directly to OutputStream.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

doBreak

public boolean **doBreak**

(continued from last page)

Constructors

ResponseFunction

```
public ResponseFunction(IClient client)
```

Create an empty ResponseFunction

Parameters:

client - client

ResponseFunction

```
public ResponseFunction(com.wowza.wms.netconnection.INetConnection netConnection)
```

Create an empty ResponseFunction

Parameters:

netConnection - net connection

ResponseFunction

```
public ResponseFunction(com.wowza.wms.netconnection.INetConnection netConnection,
AMFObj amfObj)
```

Create an empty ResponseFunction

Parameters:

netConnection - net connection

amfObj - amf object

ResponseFunction

```
public ResponseFunction(IClient client,
AMFObj amfObj)
```

Create an empty ResponseFunction

Parameters:

client - client object

amfObj - amf object

ResponseFunction

```
public ResponseFunction(IMediaStream stream,
AMFObj amfObj)
```

Create an empty ResponseFunction

Parameters:

stream - media stream object

amfObj - amf object

Methods

setSrc

```
public void setSrc(int src)
```

(continued from last page)

Set the src (stream id) for the message.

Parameters:

src - src (stream id)

createDefaultMessage

```
public void createDefaultMessage(String inName,  
    double inResultNum)
```

Create a default message (onStatus type messages).

Parameters:

inName - handler name (example: onStatus)

inResultNum - result number of 0 if not result

createConnectMessage

```
public void createConnectMessage(String inName,  
    double inResultNum)
```

Create net connection connect message.

Parameters:

inName - handler name (example: connection)

inResultNum - result number if 0 not a result

createSeekMessage

```
public void createSeekMessage(String inName)
```

Create a seek result message.

Parameters:

inName - handler name (example: onStatus)

createPlayStatusMessage

```
public void createPlayStatusMessage(String inName)
```

Create onPlayStatus message.

Parameters:

inName - handler name (example: onPlayStatus)

setMessageBytes

```
public void setMessageBytes(byte[] messageBytes)
```

createBroadcastMessage

```
public void createBroadcastMessage(java.nio.ByteBuffer messageBytes)
```

Create broadcast message. Used for ByteBuffer handler calls.

Parameters:

(continued from last page)

messageBytes - ByteBuffer with raw AMFData bytes to be sent to client

createSOMessage

```
public void createSOMessage(byte[] messageBuffer,  
    int objectEncoding)
```

Create shared object message. Used for ByteBuffer handler calls.

Parameters:

messageBuffer

createEnhancedSeekMessage

```
public void createEnhancedSeekMessage(byte[] messageBuffer,  
    int src,  
    int tc)
```

addBody

```
public void addBody(AMFData body)
```

Add AMFData to the body of the message.

Parameters:

body - AMFData message

setBody

```
public void setBody(int index,  
    AMFData body)
```

Add AMFData to the body of the message.

Parameters:

index - index in body list

body - AMFData message

addBytes

```
public void addBytes(byte[] inbytes)
```

Add raw AMFData bytes[] to message body

Parameters:

inbytes - raw AMFData bytes[]

setRetAMFNumber

```
public void setRetAMFNumber(int innum)
```

Set the return channel id

Parameters:

innum - return channel id

(continued from last page)

getTimecode

```
public long getTimecode()
```

Get function timecode (milliseconds).

Returns:

function timecode (milliseconds)

setTimecode

```
public void setTimecode(long timecode)
```

Set function timecode (milliseconds).

Parameters:

timecode - function timecode (milliseconds)

setType

```
public void setType(int type)
```

Set message type. See IVHost.CONTENTTYPE_*

Parameters:

type - message type

getType

```
public int getType()
```

Get message type. See IVHost.CONTENTTYPE_*

Returns:

message type

write

```
public int write(java.io.OutputStream out,  
int chunkSize)
```

Write message directly to OutputStream.

Parameters:

out - OutputStream

Returns:

number of bytes written

isForceAMF0

```
public boolean isForceAMF0()
```

setForceAMF0

```
public void setForceAMF0(boolean forceAMF0)
```

(continued from last page)

write

```
public int write(java.io.OutputStream out,  
                boolean isAbsTimecode,  
                int chunkSize)
```

Write message directly to OutputStream

Parameters:

out - OutputStream
isAbsTimecode - is timecode absolute

Returns:

number of bytes written

com.wowza.wms.response Class ResponseFunctions

java.lang.Object

└─com.wowza.wms.response.ResponseFunctions

public class **ResponseFunctions**
extends Object

ResponseFunctions: collection of ResponseFunction objects. This interface is used to asynchronously collect up a set of client responses that will be sent to the client at the next opportunity.

Constructor Summary

public	ResponseFunctions()
--------	-------------------------------------

Method Summary

void	add(ResponseFunction wmsResponseFunction) Add a function.
void	clear()
boolean	isPending() Are there any pending items in the list.
int	output(java.io.OutputStream out, int sendChunkSize) Write all functions (in order added to list) to OutputStream.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

ResponseFunctions

public **ResponseFunctions()**

Methods

add

public void **add**([ResponseFunction wmsResponseFunction](#))

Add a function.

(continued from last page)

Parameters:wmsResponseFunction - function

clear

```
public void clear()
```

isPending

```
public boolean isPending()
```

Are there any pending items in the list.

Returns:

true if items in list

output

```
public int output(java.io.OutputStream out,  
int sendChunkSize)
```

Write all functions (in order added to list) to OutputStream.

Parameters:

out - OutputStream

Returns:

total number of bytes written

Package

com.wowza.wms.rtp.model

com.wowza.wms.rtp.model Interface IRTPMessageHandler

public interface **IRTPMessageHandler**
extends

IRTPMessageHandler: Internal use.

Method Summary

void	handleMessage (java.net.SocketAddress socketAddr, byte[] buffer, int offset, int len)
------	---

Methods

handleMessage

```
public void handleMessage(java.net.SocketAddress socketAddr,  
    byte[] buffer,  
    int offset,  
    int len)
```

com.wowza.wms.rtp.model

Interface IRTPMetadataProvider

public interface **IRTPMetadataProvider**
extends

IRTPMetadataProvider: Internal use.

Method Summary	
byte[]	getMetadataPacket (RTPStream stream)

Methods

getMetadataPacket
public byte[] **getMetadataPacket**([RTPStream](#) stream)

com.wowza.wms.rtp.model

Interface IRTPSessionNotify

public interface **IRTPSessionNotify**
extends

IRTPSessionNotify: listener interface for RTP sessions. See RTPSessions.addSessionListener(IRTPSessionNotify listener)

Method Summary	
void	onRTPSessionCreate (RTPSession rtpSession) Invoked when RTP session is created
void	onRTPSessionDestroy (RTPSession rtpSession) Invoked when RTP session is destroyed

Methods

onRTPSessionCreate

public void **onRTPSessionCreate**([RTPSession](#) rtpSession)

 Invoked when RTP session is created

Parameters:
 rtpSession - RTP session

onRTPSessionDestroy

public void **onRTPSessionDestroy**([RTPSession](#) rtpSession)

 Invoked when RTP session is destroyed

Parameters:
 rtpSession - RTP session

com.wowza.wms.rtp.model Interface IRTSPActionNotify

All Known Implementing Classes:

[PushPublishRTSPActionNotify](#)

public interface **IRTSPActionNotify**
extends

IRTSPActionNotify: listener interface for RTSP actions. See RTPSession.addActionListener(IRTSPActionNotify actionListener)

Method Summary

void	onAnnounce (RTPSession rtspSession, com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp) Invoked by ANNOUNCE command
void	onDescribe (RTPSession rtspSession, com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp) Invoked by DESCRIBE command
void	onGetParameter (RTPSession rtspSession, com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp) Invoked by GET_PARAMETER command
void	onOptions (RTPSession rtspSession, com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp) Invoked by OPTIONS command
void	onPause (RTPSession rtspSession, com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp) Invoked by PAUSE command
void	onPlay (RTPSession rtspSession, com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp) Invoked by PLAY command
void	onRecord (RTPSession rtspSession, com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp) Invoked by RECORD command
void	onRedirect (RTPSession rtspSession, com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp) Invoked by REDIRECT command
void	onSetParameter (RTPSession rtspSession, com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp) Invoked by SET_PARAMETER command
void	onSetup (RTPSession rtspSession, com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp) Invoked by SETUP command

void

```
onTearardown(RTPSession rtspSession,  
com.wowza.wms.rtp.RTSPRequestMessage req,  
com.wowza.wms.rtp.RTSPResponseMessages resp)
```

Invoked by TEARDOWN command

Methods

onDescribe

```
public void onDescribe(RTPSession rtspSession,  
com.wowza.wms.rtp.RTSPRequestMessage req,  
com.wowza.wms.rtp.RTSPResponseMessages resp)
```

Invoked by DESCRIBE command

Parameters:

rtspSession - RTP session
req - RTP request
resp - RTP response

onAnnounce

```
public void onAnnounce(RTPSession rtspSession,  
com.wowza.wms.rtp.RTSPRequestMessage req,  
com.wowza.wms.rtp.RTSPResponseMessages resp)
```

Invoked by ANNOUNCE command

Parameters:

rtspSession - RTP session
req - RTP request
resp - RTP response

onSetParameter

```
public void onSetParameter(RTPSession rtspSession,  
com.wowza.wms.rtp.RTSPRequestMessage req,  
com.wowza.wms.rtp.RTSPResponseMessages resp)
```

Invoked by SET_PARAMETER command

Parameters:

rtspSession - RTP session
req - RTP request
resp - RTP response

onGetParameter

```
public void onGetParameter(RTPSession rtspSession,  
com.wowza.wms.rtp.RTSPRequestMessage req,  
com.wowza.wms.rtp.RTSPResponseMessages resp)
```

Invoked by GET_PARAMETER command

Parameters:

rtspSession - RTP session
req - RTP request
resp - RTP response

onOptions

```
public void onOptions(RTPSession rtspSession,  
    com.wowza.wms.rtp.RTSPRequestMessage req,  
    com.wowza.wms.rtp.RTSPResponseMessages resp)
```

Invoked by OPTIONS command

Parameters:

rtspSession - RTP session
req - RTP request
resp - RTP response

onPause

```
public void onPause(RTPSession rtspSession,  
    com.wowza.wms.rtp.RTSPRequestMessage req,  
    com.wowza.wms.rtp.RTSPResponseMessages resp)
```

Invoked by PAUSE command

Parameters:

rtspSession - RTP session
req - RTP request
resp - RTP response

onPlay

```
public void onPlay(RTPSession rtspSession,  
    com.wowza.wms.rtp.RTSPRequestMessage req,  
    com.wowza.wms.rtp.RTSPResponseMessages resp)
```

Invoked by PLAY command

Parameters:

rtspSession - RTP session
req - RTP request
resp - RTP response

onRecord

```
public void onRecord(RTPSession rtspSession,  
    com.wowza.wms.rtp.RTSPRequestMessage req,  
    com.wowza.wms.rtp.RTSPResponseMessages resp)
```

Invoked by RECORD command

Parameters:

rtspSession
req
resp

onRedirect

```
public void onRedirect(RTPSession rtspSession,  
    com.wowza.wms.rtp.RTSPRequestMessage req,  
    com.wowza.wms.rtp.RTSPResponseMessages resp)
```

Invoked by REDIRECT command

(continued from last page)

Parameters:

rtspSession - RTP session
req - RTP request
resp - RTP response

onSetup

```
public void onSetup(RTPSession rtspSession,  
    com.wowza.wms.rtp.RTSPRequestMessage req,  
    com.wowza.wms.rtp.RTSPResponseMessages resp)
```

Invoked by SETUP command

Parameters:

rtspSession - RTP session
req - RTP request
resp - RTP response

onTeardown

```
public void onTeardown(RTPSession rtspSession,  
    com.wowza.wms.rtp.RTSPRequestMessage req,  
    com.wowza.wms.rtp.RTSPResponseMessages resp)
```

Invoked by TEARDOWN command

Parameters:

rtspSession - RTP session
req - RTP request
resp - RTP response

com.wowza.wms.rtp.model

Class RTPContext

java.lang.Object

└─com.wowza.wms.rtp.model.RTPContext

public class **RTPContext**
extends Object

RTPContext: RTP context

Constructor Summary

public	RTPContext (IVHost vhost) Constructor
--------	---

Method Summary

boolean	acquireSocketAddress (java.net.SocketAddress socketAddress) Acquire UDP socket address
int	acquireUDPPortPair () Acquire next available UDP port pair
int	acquireUDPPortPair (int port) Acquire UDP port pair
void	cacheRTPStream (RTPStream stream) Cache an RTP stream, Internal use.
void	doWatchdog () Idle events for cleanup
boolean	existsRTSPTunnelingSession (String sessionId) Return true if session id is valid RTSP/RTP tunneling session id
int[]	expandToPortPair (int port) Expand a single port to a pair.
RTPPacketizerItem	getAudioPacketizerItem (IApplicationInstance appInstance, int codecId) Get audio packetizer for a given codec id.
com.wowza.wms.rtp.dep acketizer.RTPDePacket izerList	getDePacketizerList () Get a list of the available depacketizers
String	getDePacketizerName (RTPTrack rtpTrack) Get a depacketizer by name
Object	getLock () Get the UDP port manager lock
Object	getRTSPTunnelingLock () Get the RTSP/RTP tunneling lock

com.wowza.wms.rtsp.RTSP TunnelingSession	<u>getRTSPTunnelingSession</u> (String sessionId) Get RTSP/RTP tunneling session by session id
<u>RTPSessions</u>	<u>getSessions</u> () Get RTP sessions
RTPPacketizerItem	<u>getStreamPacketizerItem</u> (<u>IApplicationInstance</u> appInstance, int codecId) Get stream packetizer for a given codec id.
com.wowza.wms.rtp.transport.UDPTransportManager	<u>getUDPTransportManager</u> () Get the UDP transport manager
<u>IVHost</u>	<u>getVHost</u> () Get vhost
RTPPacketizerItem	<u>getVideoPacketizerItem</u> (<u>IApplicationInstance</u> appInstance, int codecId) Get video packetizer for a given codec id.
void	<u>init</u> () Initialize
void	<u>putAudioPacketizerItem</u> (int codecId, RTPPacketizerItem rtpPacketizerInfo) Set the audio packetizer for a given codec id
void	<u>putRTSPTunnelingSession</u> (String sessionId, com.wowza.wms.rtsp.RTSP TunnelingSession rtspTunnelingSession) Remove RTSP/RTP tunneling session by session id
void	<u>putStreamPacketizerItem</u> (int codecId, RTPPacketizerItem rtpPacketizerInfo) Set the stream packetizer for a given codec id
void	<u>putVideoPacketizerItem</u> (int codecId, RTPPacketizerItem rtpPacketizerInfo) Set the video packetizer for a given codec id
boolean	<u>releaseSocketAddress</u> (java.net.SocketAddress socketAddress) Release UDP socket address
void	<u>releaseUDPPortPair</u> (int port) Release port pair
com.wowza.wms.rtsp.RTSP TunnelingSession	<u>removeRTSPTunnelingSession</u> (String sessionId) Remove RTSP/RTP tunneling session by session id
void	<u>shutdown</u> () Shutdown
void	<u>shutdownRTPSession</u> (<u>RTPSession</u> rtpSession) Gracefully and forcefully shutdown and RTP session
<u>RTPStream</u>	<u>uncacheRTPStream</u> (String streamId) UnCache an RTP stream, Internal use.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

RTPContext

```
public RTPContext(IVHost vhost)
```

Constructor

Parameters:

vhost - vhost

Methods

doWatchdog

```
public void doWatchdog()
```

Idle events for cleanup

init

```
public void init()
```

Initialize

shutdown

```
public void shutdown()
```

Shutdown

getAudioPacketizerItem

```
public RTPPacketizerItem getAudioPacketizerItem(IApplicationInstance appInstance,  
int codecId)
```

Get audio packetizer for a given codec id.

Parameters:

appInstance - application instance

codecId - codec id, see IVHost.CODEC_AUDIO_*

Returns:

packetizer info

putAudioPacketizerItem

```
public void putAudioPacketizerItem(int codecId,  
RTPPacketizerItem rtpPacketizerInfo)
```

Set the audio packetizer for a given codec id

Parameters:

codecId - codec id, see IVHost.CODEC_AUDIO_*

rtpPacketizerInfo - packetizer info

getVideoPacketizerItem

```
public RTPPacketizerItem getVideoPacketizerItem(IApplicationInstance appInstance,  
int codecId)
```

Get video packetizer for a given codec id.

Parameters:

appInstance - application instance
codecId - codec id, see IVHost.CODEC_VIDEO_*

Returns:

packetizer info

putVideoPacketizerItem

```
public void putVideoPacketizerItem(int codecId,  
RTPPacketizerItem rtpPacketizerInfo)
```

Set the video packetizer for a given codec id

Parameters:

codecId - codec id, see IVHost.CODEC_VIDEO_*rtpPacketizerInfo - packetizer info

getStreamPacketizerItem

```
public RTPPacketizerItem getStreamPacketizerItem(IApplicationInstance appInstance,  
int codecId)
```

Get stream packetizer for a given codec id.

Parameters:

appInstance - application instance
codecId - codec id, see IVHost.CODEC_STREAM_*

Returns:

packetizer info

putStreamPacketizerItem

```
public void putStreamPacketizerItem(int codecId,  
RTPPacketizerItem rtpPacketizerInfo)
```

Set the stream packetizer for a given codec id

Parameters:

codecId - codec id, see IVHost.CODEC_STREAM_*rtpPacketizerInfo - packetizer info

getSessions

```
public RTPSessions getSessions()
```

Get RTP sessions

Returns:

RTP sessions

getVHost

```
public IVHost getVHost()
```

Get vhost

Returns:
vhost

getUDPTransportManager

```
public com.wowza.wms.rtp.transport.UDPTransportManager getUDPTransportManager()
```

Get the UDP transport manager

Returns:
UDP transport manager

getLock

```
public Object getLock()
```

Get the UDP port manager lock

Returns:
UDP port manager lock

acquireSocketAddress

```
public boolean acquireSocketAddress(java.net.SocketAddress socketAddress)
```

Acquire UDP socket address

Parameters:
socketAddress - UDP socket address

Returns:
true if available

releaseSocketAddress

```
public boolean releaseSocketAddress(java.net.SocketAddress socketAddress)
```

Release UDP socket address

Parameters:
socketAddress - UDP socket address

Returns:
true if available

acquireUDPPortPair

```
public int acquireUDPPortPair(int port)
```

Acquire UDP port pair

(continued from last page)

Parameters:

port - starting port

Returns:

port

expandToPortPair

```
public int[] expandToPortPair(int port)
```

Expand a single port to a pair. Ports are allocated in pairs always starting with even port number.

Parameters:

port - port

Returns:

port

acquireUDPPortPair

```
public int acquireUDPPortPair()
```

Acquire next available UDP port pair

Returns:

port

releaseUDPPortPair

```
public void releaseUDPPortPair(int port)
```

Release port pair

Parameters:

port - first port of pair

cacheRTPStream

```
public void cacheRTPStream(RTPStream stream)
```

Cache an RTP stream, Internal use.

Parameters:

stream - RTP stream

uncacheRTPStream

```
public RTPStream uncacheRTPStream(String streamId)
```

UnCache an RTP stream, Internal use.

Parameters:

streamId - stream id

Returns:

RTP Stream

(continued from last page)

getDePacketizerList

```
public com.wowza.wms.rtp.depaketizer.RTPDePacketizerList getDePacketizerList()
```

Get a list of the available depacketizers

Returns:

list of the available depacketizers

getDePacketizerName

```
public String getDePacketizerName(RTPTrack rtpTrack)
```

Get a depacketizer by name

Parameters:

rtpTrack - track

Returns:

depacketizer

getRTSPTunnelingLock

```
public Object getRTSPTunnelingLock()
```

Get the RTSP/RTP tunneling lock

Returns:

RTSP/RTP tunneling lock

getRTSPTunnelingSession

```
public com.wowza.wms.rtsp.RTSPTunnelingSession getRTSPTunnelingSession(String  
sessionId)
```

Get RTSP/RTP tunneling session by session id

Parameters:

sessionId - session id

Returns:

RTSP/RTP tunneling session

removeRTSPTunnelingSession

```
public com.wowza.wms.rtsp.RTSPTunnelingSession removeRTSPTunnelingSession(String  
sessionId)
```

Remove RTSP/RTP tunneling session by session id

Parameters:

sessionId - session id

Returns:

RTSP/RTP tunneling session if removed

(continued from last page)

existsRTSPTunnelingSession

```
public boolean existsRTSPTunnelingSession(String sessionId)
```

Return true if session id is valid RTSP/RTP tunneling session id

Parameters:

sessionId - session id

Returns:

true if session id is valid RTSP/RTP tunneling session id

putRTSPTunnelingSession

```
public void putRTSPTunnelingSession(String sessionId,  
    com.wowza.wms.rtsp.RTSPTunnelingSession rtspTunnelingSession)
```

Remove RTSP/RTP tunneling session by session id

Parameters:

sessionId - session id

rtspTunnelingSession - RTSP/RTP tunneling session

shutdownRTPSession

```
public void shutdownRTPSession(RTPSession rtpSession)
```

Gracefully and forcefully shutdown and RTP session

Parameters:

rtpSession - RTP session

com.wowza.wms.rtp.model Class RTPDestination

java.lang.Object

└-com.wowza.wms.rtp.model.RTPDestination

public class **RTPDestination**
extends Object

RTPDestination: Fully describes an RTP destination.

Constructor Summary

public	RTPDestination()
--------	----------------------------------

Method Summary

String	getAudioHost() Get audio host
int	getAudioPort() Get audio port
String	getHost() Get the host
String	getHostType() Get host type (default IP4)
String	getName() Get name
int	getStreamPort() Get stream port
int	getTTL() Get time to live
String	getVideoHost() Get video host
int	getVideoPort() Get video port
boolean	isMulticast() Return true if multicast destination
boolean	isRTPWrapped() Is stream wrapped in RTP (MPEG-TS in RTP)
boolean	isStream() Return true if stream destination (and not native RTP destination)

void	<code>setAudioHost</code> (String audioHost) Set audio host
void	<code>setAudioPort</code> (int audioPort) Set audio port
void	<code>setHost</code> (String host) Set host
void	<code>setHostType</code> (String hostType) Set host type (default IP4)
void	<code>setName</code> (String name) Set name
void	<code>setRTPWrapped</code> (boolean isRTPWrapped) Is stream wrapped in RTP (MPEG-TS in RTP)
void	<code>setStreamPort</code> (int streamPort) Set stream port
void	<code>setTTL</code> (int ttl) Set time to live
void	<code>setVideoHost</code> (String videoHost) Set video host
void	<code>setVideoPort</code> (int videoPort) Set video port
String	<code>toString</code> ()

Methods inherited from class java.lang.Object

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Constructors

RTPDestination

```
public RTPDestination()
```

Methods

toString

```
public String toString()
```

(continued from last page)

isMulticast

```
public boolean isMulticast()
```

Return true if multicast destination

Returns:

true if multicast destination

isStream

```
public boolean isStream()
```

Return true if stream destination (and not native RTP destination)

Returns:

true if stream destination

getHost

```
public String getHost()
```

Get the host

Returns:

host

setHost

```
public void setHost(String host)
```

Set host

Parameters:

host - host

getVideoPort

```
public int getVideoPort()
```

Get video port

Returns:

video port

setVideoPort

```
public void setVideoPort(int videoPort)
```

Set video port

Parameters:

videoPort - video port

getAudioPort

```
public int getAudioPort()
```


(continued from last page)

Get audio port

Returns:

audio port

setAudioPort

```
public void setAudioPort(int audioPort)
```

Set audio port

Parameters:

audioPort - audio port

getStreamPort

```
public int getStreamPort()
```

Get stream port

Returns:

stream port

setStreamPort

```
public void setStreamPort(int streamPort)
```

Set stream port

Parameters:

streamPort - stream port

getVideoHost

```
public String getVideoHost()
```

Get video host

Returns:

video host

setVideoHost

```
public void setVideoHost(String videoHost)
```

Set video host

Parameters:

videoHost - video host

getAudioHost

```
public String getAudioHost()
```

Get audio host

Returns:

audio host

setAudioHost

```
public void setAudioHost(String audioHost)
```

Set audio host

Parameters:

audioHost - audio host

getHostType

```
public String getHostType()
```

Get host type (default IP4)

Returns:

host type

setHostType

```
public void setHostType(String hostType)
```

Set host type (default IP4)

Parameters:

hostType - host type

getTTL

```
public int getTTL()
```

Get time to live

Returns:

time to live

setTTL

```
public void setTTL(int ttl)
```

Set time to live

Parameters:

ttl - time to live

isRTPWrapped

```
public boolean isRTPWrapped()
```

Is stream wrapped in RTP (MPEG-TS in RTP)

Returns:

true if stream wrapped in RTP

setRTPWrapped

```
public void setRTPWrapped(boolean isRTPWrapped)
```

(continued from last page)

Is stream wrapped in RTP (MPEG-TS in RTP)

Parameters:

isRTPWrapped - true if stream wrapped in RTP

getName

```
public String getName()
```

Get name

Returns:

name

setName

```
public void setName(String name)
```

Set name

Parameters:

name - name

com.wowza.wms.rtp.model

Class RTPPort

java.lang.Object

└─com.wowza.wms.rtp.model.RTPPort

All Implemented Interfaces:

com.wowza.wms.rtp.transport.IUDPMessageHandler

public class **RTPPort**
 extends Object
 implements com.wowza.wms.rtp.transport.IUDPMessageHandler

Constructor Summary

public	<u>RTPPort</u> (String inIpAddress, int inPort, String outIpAddress, int outPort, boolean isMulticast) Constructor
--------	---

Method Summary

void	<u>bind</u> (int direction) Bind to port
int	<u>getAddressCount</u> () Get address count
<u>I RTPMessageHandler</u>	<u>getHandler</u> () Get the message handler
String	<u>getInIpAddress</u> () Get in IP address
int	<u>getInPort</u> () Get in port
String	<u>getOutIpAddress</u> () Get out IP address
int	<u>getOutPort</u> () Get out port
RTPTrack	<u>getTrack</u> () Get RTP track
int	<u>getTTL</u> () Get time to live (milliseconds)
com.wowza.wms.rtp.transport.IUDPTransport	<u>getUDPTransport</u> () Get the UDP transport
void	<u>handleMessage</u> (java.net.SocketAddress socketAddr, Object message)

boolean	<u>isBlockUDPOut</u> ()
boolean	<u>isInMulticast</u> () Is in stream multicast
boolean	<u>isMulticast</u> () Is multicast
boolean	<u>isOutMulticast</u> () Is out stream multicast
void	<u>sendMessage</u> (byte[] message, int offset, int len) Send a message out
void	<u>sendResponse</u> (byte[] message) Send a response
void	<u>sendResponse</u> (byte[] message, int offset, int len) Send response
void	<u>sendResponse</u> (byte[] message, int offset, int len, java.net.SocketAddress destination) Send response to destination
void	<u>sessionClosed</u> (com.wowza.wms.rtp.transport.IUDPTransportSession session)
void	<u>sessionOpened</u> (com.wowza.wms.rtp.transport.IUDPTransportSession session)
void	<u>setAddressCount</u> (int addressCount) Set address count
void	<u>setBlockUDPOut</u> (boolean blockUDPOut)
void	<u>setHandler</u> (<u>IRTPMessageHandler</u> handler) Set the message handler
void	<u>setTrack</u> (RTPTrack track) Set RTP track
void	<u>setTTL</u> (int ttl) Set time to live (milliseconds)
void	<u>shutdown</u> () Shutdown port
void	<u>unbind</u> () Unbind

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface com.wowza.wms.rtp.transport.IUDPMessageHandler

handleMessage, sessionClosed, sessionOpened

Constructors

RTPPort

```
public RTPPort(String inIpAddress,  
               int inPort,  
               String outIpAddress,  
               int outPort,  
               boolean isMulticast)
```

Constructor

Parameters:

inIpAddress - in ip address
inPort - in port
outIpAddress - out ip address
outPort - out port
isMulticast - true if multicast

Methods

getTrack

```
public RTPTrack getTrack()
```

Get RTP track

Returns:

RTP track

setTrack

```
public void setTrack(RTPTrack track)
```

Set RTP track

Parameters:

track - RTP track

getUDPTransport

```
public com.wowza.wms.rtp.transport.IUDPTransport getUDPTransport()
```

Get the UDP transport

Returns:

UDP transport

getInIpAddress

```
public String getInIpAddress()
```

Get in IP address

Returns:

in IP address

getOutIpAddress

```
public String getOutIpAddress()
```

Get out IP address

Returns:

out IP address

getInPort

```
public int getInPort()
```

Get in port

Returns:

in port

getOutPort

```
public int getOutPort()
```

Get out port

Returns:

out port

bind

```
public void bind(int direction)
```

Bind to port

Parameters:

direction - in or out, see IUDPTTransport.DIRECTION_*

unbind

```
public void unbind()
```

Unbind

shutdown

```
public void shutdown()
```

Shutdown port

getHandler

```
public IRTPMessageHandler getHandler()
```

Get the message handler

Returns:

message handler

setHandler

```
public void setHandler(IRTPMessageHandler handler)
```

Set the message handler

Parameters:

handler - message handler

sendMessage

```
public void sendMessage(byte[] message,  
    int offset,  
    int len)
```

Send a message out

Parameters:

message - message

offset - offset

len - len

sendResponse

```
public void sendResponse(byte[] message)
```

Send a response

Parameters:

message - message

sendResponse

```
public void sendResponse(byte[] message,  
    int offset,  
    int len)
```

Send response

Parameters:

message - message

offset - offset

len - len

sendResponse

```
public void sendResponse(byte[] message,  
    int offset,  
    int len,  
    java.net.SocketAddress destination)
```

Send response to destination

Parameters:

message - message

offset - offset

len - len

destination - destination address

sessionOpened

```
public void sessionOpened(com.wowza.wms.rtp.transport.IUDPTransportSession session)
```

sessionClosed

```
public void sessionClosed(com.wowza.wms.rtp.transport.IUDPTransportSession session)
```

handleMessage

```
public void handleMessage(java.net.SocketAddress socketAddr,  
    Object message)
```

isInMulticast

```
public boolean isInMulticast()
```

Is in stream multicast

Returns:

true if multicast

isOutMulticast

```
public boolean isOutMulticast()
```

Is out stream multicast

Returns:

true if multicast

isMulticast

```
public boolean isMulticast()
```

Is multicast

Returns:

true if multicast

getTTL

```
public int getTTL()
```

Get time to live (milliseconds)

Returns:

time to live (milliseconds)

(continued from last page)

setTTL

```
public void setTTL(int ttl)
```

Set time to live (milliseconds)

Parameters:

ttl - time to live (milliseconds)

getAddressCount

```
public int getAddressCount()
```

Get address count

Returns:

address count

setAddressCount

```
public void setAddressCount(int addressCount)
```

Set address count

Parameters:

addressCount - address count

isBlockUDPOut

```
public boolean isBlockUDPOut()
```

setBlockUDPOut

```
public void setBlockUDPOut(boolean blockUDPOut)
```

com.wowza.wms.rtp.model Class RTPPushPublishSession

java.lang.Object

└--com.wowza.wms.rtp.model.RTPPushPublishSession

public class **RTPPushPublishSession**
extends Object

RTPPushPublishSession: RTP push publishing session

Constructor Summary

public	RTPPushPublishSession()
--------	---

Method Summary

RTPSession	getRTPSession() Get RTP session
String	getSDPData() Get SDP data
void	setRTPSession(RTPSession rtpSession) Set RTP session
void	setSDPData(String sdpData) Set SDP data

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

RTPPushPublishSession

public **RTPPushPublishSession()**

Methods

getSDPData

public String **getSDPData()**

Get SDP data

(continued from last page)

Returns:

SDP data

setSDPData

```
public void setSDPData(String sdpData)
```

Set SDP data

Parameters:

sdpData - SDP data

getRTPSession

```
public RTPSession getRTPSession()
```

Get RTP session

Returns:

RTP session

setRTPSession

```
public void setRTPSession(RTPSession rtpSession)
```

Set RTP session

Parameters:

rtpSession - RTP session

com.wowza.wms.rtp.model

Class RTPSession

java.lang.Object

└-com.wowza.wms.rtp.model.RTPSession

public class **RTPSession**
extends Object

RTPSession: RTP session

Field Summary

protected	actionListeners
protected	appInstance
protected	appInstanceShutdown
protected	audioPacketizers
protected	authenticatePlayHandler
protected	authenticatePublishHandler
public static final	AUTHMETHOD_PLAY Value: 1
public static final	AUTHMETHOD_PUBLISH Value: 2
public static final	AUTHMETHOD_UNKNOWN Value: 0
protected	connectionHolder
protected	cookieStr
protected	debugRTSPSession
protected	elapsedTime
protected	idleFrequency
protected	idleHandler

protected	<u>ioPerformanceCounter</u>
protected	<u>ioSession</u>
protected	<u>ip</u>
protected	<u>isAnnounce</u>
protected	<u>isConnected</u>
protected	<u>isDescribe</u>
protected	<u>isSessionValid</u>
protected	<u>lastAuthenticateMethod</u>
protected	<u>lock</u>
protected	<u>loggedConnect</u>
protected	<u>properties</u>
protected	<u>queryStr</u>
protected	<u>redirectSession</u>
protected	<u>redirectSessionCode</u>
protected	<u>redirectSessionMessage</u>
protected	<u>redirectSessionURL</u>
protected	<u>referrer</u>
protected	<u>rtpWriteListener</u>
protected	<u>rtspPlayRangeStart</u>
protected	<u>rtspPlayRangeStop</u>
protected	<u>rtspTunnelingSessionId</u>
protected	<u>serverIp</u>
protected	<u>serverPort</u>
protected	<u>sessionId</u>

protected	shutdownClient
protected	streamPacketizers
protected	streams
protected	streamsOrder
protected	timeCreated
protected	totalIOPerformance2Last
protected	totalIOPerformanceLast
protected	uri
protected	userAgent
protected	vhost
protected	videoPacketizers

Constructor Summary

public	RTPSession (String sessionId) Constructor
--------	--

Method Summary

void	acceptSession () Accept this session
void	addActionListener (IRTSPActionNotify actionListener) Add action listener
void	addIOPerformance (IOPerformanceCounter totalIOPerformanceResult) Internal use
void	addIOPerformance2 (IOPerformanceCounter totalIOPerformanceResult) Internal use
void	addRTSPStream (RTPStream stream) Add RTP stream
void	clearLoggingValues () Clear logging values, Internal use.
void	doIdle ()
IApplicationInstance	getAppInstance () Get application instance

RTPPacketizerItem	getAudioPacketizerItem(RTPContext rtpContext, IApplicationInstance appInstance, int codecId) Get audio packetizer for a given codec id.
IAuthenticateRTSP	getAuthenticatePlayHandler() Get the authentication play handler
IAuthenticateRTSP	getAuthenticatePublishHandler() Get the RTP authentication handler
ConnectionHolder	getConnectionHolder() Get connection holder, Internal use.
String	getCookieStr() Get cookie string
ElapsedTimer	getElapsedTime() Get the elapsed timer for this RTP session
int	getIdleFrequency() Get idle frequency (milliseconds)
RTPIdleHandler	getIdleHandler() Get idle handler
IOPerformanceCounter	getIOPerformanceCounter() Get IO performance counter
org.apache.mina.common.support.BaseIoSession	getIoSession()
String	getIp() Get remote IP address
int	getLastAuthenticateMethod() Get the last method received
WMSProperties	getProperties() Get properties
String	getQueryStr() Get query string
int	getRedirectSessionCode()
String	getRedirectSessionMessage()
String	getRedirectSessionURL()
String	getReferrer() Get referrer
RTPWriteListener	getRTPWriteListener() Get the RTP write listener for this session
double	getRTSPPlayRangeStart() Get play start range, Internal use.

double	<u>getRTSPPlayRangeStop()</u> Get play stop range, Internal use.
<u>RTPStream</u>	<u>getRTSPStream()</u> Get the default RTP Stream (all RTP sessions have a single RTP Stream)
<u>RTPStream</u>	<u>getRTSPStream(String streamId)</u> Get RTP Stream
String	<u>getRTSPTunnelingSessionId()</u> Get the RTSP/RTP tunneling session id
String	<u>getServerIp()</u> Get the server IP address
int	<u>getServerPort()</u> Get server port
String	<u>getSessionId()</u> Get session id
RTPPacketizerItem	<u>getStreamPacketizerItem(RTPContext rtpContext, IApplicationInstance appInstance, int codecId)</u> Get stream packetizer for a given codec id.
String	<u>getTimeRunning()</u> Get the time running for this RTP session
double	<u>getTimeRunningSeconds()</u> Get the number of second this RTP session has been running
String	<u>getUri()</u> Get URI
String	<u>getUserAgent()</u> Get user agent
<u>IVHost</u>	<u>getVHost()</u> Get vhost
RTPPacketizerItem	<u>getVideoPacketizerItem(RTPContext rtpContext, IApplicationInstance appInstance, int codecId)</u> Get video packetizer for a given codec id.
boolean	<u>isAnnounce()</u> Has ANNOUNCE command been called on this session
boolean	<u>isAnnounceOrDescribe()</u> Has announce or described been called
boolean	<u>isConnected()</u> Is session connection
boolean	<u>isDebugRTSPSession()</u> True if debugging RTSP session
boolean	<u>isDescribe()</u> Has DESCRIBE command been called on this session

boolean	<u>isLoggedConnect</u> () Is connect logged
boolean	<u>isRedirectSession</u> ()
boolean	<u>isSessionValid</u> () Is this session valid
boolean	<u>isShutdownClient</u> () Is RTP session shutdown
void	<u>onAnnounce</u> (com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp) Invoked on ANNOUNCE command
void	<u>onDescribe</u> (com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp) Invoked on DESCRIBE command
void	<u>onGetParameter</u> (com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp) Invoked on GET_PARAMETER command
void	<u>onOptions</u> (com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp) Invoked on OPTIONS command
void	<u>onPause</u> (com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp) Invoked on PAUSE command
void	<u>onPlay</u> (com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp) Invoked on PLAY command
void	<u>onRecord</u> (com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp) Invoked on RECORD command
void	<u>onRedirect</u> (com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp) Invoked on REDIRECT command
void	<u>onSetParameter</u> (com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp) Invoked on SET_PARAMETER command
void	<u>onSetup</u> (com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp) Invoked on SETUP command
void	<u>onTeardown</u> (com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp) Invoked on TEARDOWN command
void	<u>putAudioPacketizerItem</u> (int codecId, String classPath) Set the audio packetizer for a given codec id
void	<u>putStreamPacketizerItem</u> (int codecId, String classPath) Set the stream packetizer for a given codec id

void	<u>putVideoPacketizerItem</u> (int codecId, String classPath) Set the video packetizer for a given codec id
void	<u>redirectSession</u> (String redirectSessionURL)
void	<u>redirectSession</u> (String redirectSessionURL, int redirectSessionCode)
void	<u>rejectSession</u> () Reject this session
void	<u>removeActionListener</u> (<u>IRTSPActionNotify</u> actionListener) Remove action listener
<u>RTPStream</u>	<u>removeRTSPStream</u> (String streamId) Remove RTP stream
void	<u>setAnnounce</u> (boolean isAnnounce) Set ANNOUNCE command been called on this session
void	<u>setAppInstance</u> (<u>IApplicationInstance</u> appInstance) Set application instance
void	<u>setAuthenticatePlayHandler</u> (<u>IAuthenticateRTSP</u> authenticatePlayHandler) Set the authentication play handler
void	<u>setAuthenticatePublishHandler</u> (<u>IAuthenticateRTSP</u> authenticatePublishHandler) Set the RTP authentication handler
void	<u>setConnected</u> (boolean isConnected) Set session is connected
void	<u>setCookieStr</u> (String cookieStr) Set cooking string
void	<u>setDebugRTSPSession</u> (boolean debugRTSPSession) Set debugging RTSP session
void	<u>setDescribe</u> (boolean isDescribe) Set DESCRIBE command been called on this session
void	<u>setIdleFrequency</u> (int idleFrequency) Set idle frequency (milliseconds)
void	<u>setIdleHandler</u> (<u>RTPIdleHandler</u> idleHandler) Set idle handler
void	<u>setIOPerformanceCounter</u> (<u>IOPerformanceCounter</u> ioPerformanceCounter) Set IO performance counter
void	<u>setIoSession</u> (org.apache.mina.common.support.BaseIoSession ioSession)
void	<u>setIp</u> (String ip) Set remote IP address
void	<u>setLastAuthenticateMethod</u> (int lastAuthenticateMethod) Set last method received

void	<u>setLoggedConnect</u> (boolean loggedConnect) Set connect logged
void	<u>setQueryStr</u> (String queryStr) Set query string
void	<u>setRedirectSession</u> (boolean redirectSession)
void	<u>setRedirectSessionCode</u> (int redirectSessionCode)
void	<u>setRedirectSessionMessage</u> (String redirectSessionMessage)
void	<u>setRedirectSessionURL</u> (String redirectSessionURL)
void	<u>setReferrer</u> (String referrer) Set referrer
void	<u>setRTSPPlayRangeStart</u> (double rtspPlayRangeStart) Set play start range, Internal use.
void	<u>setRTSPPlayRangeStop</u> (double rtspPlayRangeStop) Set play stop range, Internal use.
void	<u>setRTSPTunnelingSessionId</u> (String rtspTunnelingSessionId) Set the RTSP/RTP tunneling session id
void	<u>setServerIp</u> (String serverIp) Set the server IP address
void	<u>setServerPort</u> (int serverPort) Set server port
void	<u>setSessionId</u> (String sessionId) Set session id
void	<u>setSessionValid</u> (boolean isSessionValid) Set session valid
void	<u>setShutdownClient</u> (boolean shutdownClient) Set RTP session shutdown
void	<u>setUri</u> (String uri) Set URI
void	<u>setUserAgent</u> (String userAgent) Set user agent
void	<u>setVHost</u> (<u>IVHost</u> vhost) Set vhost
void	<u>shutdown</u> () shutdown RTP session, Internal use.
void	<u>shutdown</u> (RTPRequestStatus status) shutdown RTP session, Internal use.
void	<u>touch</u> () Touch the stream so it doesn't timeout

void

[updateLoggingValues\(\)](#)

Update logging values, Internal use.

Methods inherited from class `java.lang.Object``clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

AUTHMETHOD_UNKNOWN

```
public static final int AUTHMETHOD_UNKNOWN
```

Constant value: **0**

AUTHMETHOD_PLAY

```
public static final int AUTHMETHOD_PLAY
```

Constant value: **1**

AUTHMETHOD_PUBLISH

```
public static final int AUTHMETHOD_PUBLISH
```

Constant value: **2**

sessionId

```
protected java.lang.String sessionId
```

rtspTunnelingSessionId

```
protected java.lang.String rtspTunnelingSessionId
```

streams

```
protected java.util.Map streams
```

streamsOrder

```
protected java.util.List streamsOrder
```

(continued from last page)

vhost

protected com.wowza.wms.vhost.IVHost **vhost**

userAgent

protected java.lang.String **userAgent**

cookieStr

protected java.lang.String **cookieStr**

ip

protected java.lang.String **ip**

isSessionValid

protected boolean **isSessionValid**

authenticatePlayHandler

protected com.wowza.wms.authentication.IAuthenticateRTSP **authenticatePlayHandler**

authenticatePublishHandler

protected com.wowza.wms.authentication.IAuthenticateRTSP **authenticatePublishHandler**

lastAuthenticateMethod

protected int **lastAuthenticateMethod**

appInstance

protected com.wowza.wms.application.IApplicationInstance **appInstance**

appInstanceShutdown

protected com.wowza.wms.application.IApplicationInstance **appInstanceShutdown**

(continued from last page)

loggedConnect

protected boolean **loggedConnect**

properties

protected com.wowza.wms.application.WMSProperties **properties**

isDescribe

protected boolean **isDescribe**

isAnnounce

protected boolean **isAnnounce**

rtpWriteListener

protected com.wowza.wms.rtp.model.RTPWriteListener **rtpWriteListener**

timeCreated

protected long **timeCreated**

ioPerformanceCounter

protected com.wowza.util.IOPerformanceCounter **ioPerformanceCounter**

totalIOPerformanceLast

protected com.wowza.util.IOPerformanceCounter **totalIOPerformanceLast**

totalIOPerformance2Last

protected com.wowza.util.IOPerformanceCounter **totalIOPerformance2Last**

(continued from last page)

connectionHolder

protected com.wowza.wms.client.ConnectionHolder **connectionHolder**

isConnected

protected boolean **isConnected**

idleHandler

protected com.wowza.wms.rtp.model.RTPIdleHandler **idleHandler**

serverIp

protected java.lang.String **serverIp**

serverPort

protected int **serverPort**

uri

protected java.lang.String **uri**

referrer

protected java.lang.String **referrer**

queryStr

protected java.lang.String **queryStr**

actionListeners

protected java.util.List **actionListeners**

elapsedTime

protected com.wowza.util.ElapsedTimer **elapsedTime**

(continued from last page)

idleFrequency

protected int **idleFrequency**

debugRTSPSession

protected boolean **debugRTSPSession**

rtspPlayRangeStart

protected double **rtspPlayRangeStart**

rtspPlayRangeStop

protected double **rtspPlayRangeStop**

shutdownClient

protected boolean **shutdownClient**

audioPacketizers

protected java.util.Map **audioPacketizers**

videoPacketizers

protected java.util.Map **videoPacketizers**

streamPacketizers

protected java.util.Map **streamPacketizers**

redirectSession

protected boolean **redirectSession**

(continued from last page)

redirectSessionCode

protected int **redirectSessionCode**

redirectSessionMessage

protected java.lang.String **redirectSessionMessage**

redirectSessionURL

protected java.lang.String **redirectSessionURL**

lock

protected java.lang.Object **lock**

ioSession

protected org.apache.mina.common.support.BaseIoSession **ioSession**

Constructors

RTPSession

public **RTPSession**(String sessionId)

Constructor

Parameters:

sessionId - session id

Methods

acceptSession

public void **acceptSession**()

Accept this session

rejectSession

public void **rejectSession**()

Reject this session

(continued from last page)

getSessionId

```
public String getSessionId()
```

Get session id

Returns:

session id

setSessionId

```
public void setSessionId(String sessionId)
```

Set session id

Parameters:

sessionId - session id

addRTSPStream

```
public void addRTSPStream(RTSPStream stream)
```

Add RTP stream

Parameters:

stream - RTP stream

removeRTSPStream

```
public RTSPStream removeRTSPStream(String streamId)
```

Remove RTP stream

Parameters:

streamId - stream id

Returns:

RTP stream

getRTSPStream

```
public RTSPStream getRTSPStream(String streamId)
```

Get RTP Stream

Parameters:

streamId - stream id

Returns:

RTP Stream

getRTSPStream

```
public RTSPStream getRTSPStream()
```

Get the default RTP Stream (all RTP sessions have a single RTP Stream)

Returns:

(continued from last page)

RTP stream

getVHost

```
public IVHost getVHost()
```

Get vhost

Returns:

vhost

setVHost

```
public void setVHost(IVHost vhost)
```

Set vhost

Parameters:

vhost - vhost

getUserAgent

```
public String getUserAgent()
```

Get user agent

Returns:

user agent

setUserAgent

```
public void setUserAgent(String userAgent)
```

Set user agent

Parameters:

userAgent - user agent

touch

```
public void touch()
```

Touch the stream so it doesn't timeout

shutdown

```
public void shutdown()
```

shutdown RTP session, Internal use.

shutdown

```
public void shutdown(RTPRequestStatus status)
```

shutdown RTP session, Internal use.

Parameters:

status

isSessionValid

```
public boolean isSessionValid()
```

Is this session valid

Returns:

true if valid

setSessionValid

```
public void setSessionValid(boolean isSessionValid)
```

Set session valid

Parameters:

isSessionValid - true if valid

getAuthenticatePublishHandler

```
public IAuthenticateRTSP getAuthenticatePublishHandler()
```

Get the RTP authentication handler

Returns:

RTP authentication handler

setAuthenticatePublishHandler

```
public void setAuthenticatePublishHandler(IAuthenticateRTSP  
authenticatePublishHandler)
```

Set the RTP authentication handler

Parameters:

authenticatePublishHandler - RTP authentication handler

getAuthenticatePlayHandler

```
public IAuthenticateRTSP getAuthenticatePlayHandler()
```

Get the authentication play handler

Returns:

authentication play handler

setAuthenticatePlayHandler

```
public void setAuthenticatePlayHandler(IAuthenticateRTSP authenticatePlayHandler)
```

Set the authentication play handler

Parameters:

authenticatePlayHandler - authentication play handler

(continued from last page)

getLastAuthenticateMethod

```
public int getLastAuthenticateMethod()
```

Get the last method received

Returns:

last method received

setLastAuthenticateMethod

```
public void setLastAuthenticateMethod(int lastAuthenticateMethod)
```

Set last method received

Parameters:

lastAuthenticateMethod - last method received

getAppInstance

```
public IApplicationInstance getAppInstance()
```

Get application instance

Returns:

application instance

setAppInstance

```
public void setAppInstance(IApplicationInstance appInstance)
```

Set application instance

Parameters:

appInstance - application instance

isLoggedConnect

```
public boolean isLoggedConnect()
```

Is connect logged

Returns:

true if logged

setLoggedConnect

```
public void setLoggedConnect(boolean loggedConnect)
```

Set connect logged

Parameters:

loggedConnect - true if logged

getIp

```
public String getIp()
```

(continued from last page)

Get remote IP address

Returns:

remote IP address

setIp

```
public void setIp(String ip)
```

Set remote IP address

Parameters:

ip - remote IP address

getProperties

```
public WMSProperties getProperties()
```

Get properties

Returns:

properties

isDescribe

```
public boolean isDescribe()
```

Has DESCRIBE command been called on this session

Returns:

true if called

setDescribe

```
public void setDescribe(boolean isDescribe)
```

Set DESCRIBE command been called on this session

Parameters:

isDescribe - true if called

isAnnounce

```
public boolean isAnnounce()
```

Has ANNOUNCE command been called on this session

Returns:

true if called

setAnnounce

```
public void setAnnounce(boolean isAnnounce)
```

Set ANNOUNCE command been called on this session

Parameters:

isAnnounce - true if called

isAnnounceOrDescribe

```
public boolean isAnnounceOrDescribe()
```

Has announce or described been called

Returns:

true if either called

getRTPWriteListener

```
public RTPWriteListener getRTPWriteListener()
```

Get the RTP write listener for this session

Returns:

RTP write listener

addIOPerformance

```
public void addIOPerformance(IOPerformanceCounter totalIOPerformanceResult)
```

Internal use

Parameters:

totalIOPerformanceResult - IO performance counter

addIOPerformance2

```
public void addIOPerformance2(IOPerformanceCounter totalIOPerformanceResult)
```

Internal use

Parameters:

totalIOPerformanceResult - IO performance counter

getConnectionHolder

```
public ConnectionHolder getConnectionHolder()
```

Get connection holder, Internal use.

Returns:

connection holder

isConnected

```
public boolean isConnected()
```

Is session connection

Returns:

true if connected

setConnected

```
public void setConnected(boolean isConnected)
```

(continued from last page)

Set session is connected

Parameters:

isConnected - true if connected

getIdleHandler

```
public RTPIdleHandler getIdleHandler()
```

Get idle handler

Returns:

idle handler

setIdleHandler

```
public void setIdleHandler(RTPIdleHandler idleHandler)
```

Set idle handler

Parameters:

idleHandler - idle handler

getServerIp

```
public String getServerIp()
```

Get the server IP address

Returns:

IP address

setServerIp

```
public void setServerIp(String serverIp)
```

Set the server IP address

Parameters:

serverIp - IP address

getServerPort

```
public int getServerPort()
```

Get server port

Returns:

server port

setServerPort

```
public void setServerPort(int serverPort)
```

Set server port

Parameters:

serverPort - server port

getUri

```
public String getUri()
```

Get URI

Returns:
URI

setUri

```
public void setUri(String uri)
```

Set URI

Parameters:
uri - URI

getReferrer

```
public String getReferrer()
```

Get referrer

Returns:
referrer

setReferrer

```
public void setReferrer(String referrer)
```

Set referrer

Parameters:
referrer - referrer

getQueryStr

```
public String getQueryStr()
```

Get query string

Returns:
query string

setQueryStr

```
public void setQueryStr(String queryStr)
```

Set query string

Parameters:
queryStr - query string

updateLoggingValues

```
public void updateLoggingValues()
```

(continued from last page)

Update logging values, Internal use.

clearLoggingValues

```
public void clearLoggingValues()
```

Clear logging values, Internal use.

addActionListener

```
public void addActionListener(IRTSPActionNotify actionListener)
```

Add action listener

Parameters:

actionListener - action listener

removeActionListener

```
public void removeActionListener(IRTSPActionNotify actionListener)
```

Remove action listener

Parameters:

actionListener - action listener

onDescribe

```
public void onDescribe(com.wowza.wms.rtsp.RTSPRequestMessage req,  
    com.wowza.wms.rtsp.RTSPResponseMessages resp)
```

Invoked on DESCRIBE command

Parameters:

req - RTP request

resp - RTP response

onAnnounce

```
public void onAnnounce(com.wowza.wms.rtsp.RTSPRequestMessage req,  
    com.wowza.wms.rtsp.RTSPResponseMessages resp)
```

Invoked on ANNOUNCE command

Parameters:

req - RTP request

resp - RTP response

onGetParameter

```
public void onGetParameter(com.wowza.wms.rtsp.RTSPRequestMessage req,  
    com.wowza.wms.rtsp.RTSPResponseMessages resp)
```

Invoked on GET_PARAMETER command

Parameters:

req - RTP request

resp - RTP response

onSetParameter

```
public void onSetParameter(com.wowza.wms.rtsp.RTSPRequestMessage req,  
    com.wowza.wms.rtsp.RTSPResponseMessages resp)
```

Invoked on SET_PARAMETER command

Parameters:

req - RTP request
resp - RTP response

onOptions

```
public void onOptions(com.wowza.wms.rtsp.RTSPRequestMessage req,  
    com.wowza.wms.rtsp.RTSPResponseMessages resp)
```

Invoked on OPTIONS command

Parameters:

req - RTP request
resp - RTP response

onPause

```
public void onPause(com.wowza.wms.rtsp.RTSPRequestMessage req,  
    com.wowza.wms.rtsp.RTSPResponseMessages resp)
```

Invoked on PAUSE command

Parameters:

req - RTP request
resp - RTP response

onPlay

```
public void onPlay(com.wowza.wms.rtsp.RTSPRequestMessage req,  
    com.wowza.wms.rtsp.RTSPResponseMessages resp)
```

Invoked on PLAY command

Parameters:

req - RTP request
resp - RTP response

onRecord

```
public void onRecord(com.wowza.wms.rtsp.RTSPRequestMessage req,  
    com.wowza.wms.rtsp.RTSPResponseMessages resp)
```

Invoked on RECORD command

Parameters:

req - RTP request
resp - RTP response

(continued from last page)

onRedirect

```
public void onRedirect(com.wowza.wms.rtsp.RTSPRequestMessage req,  
    com.wowza.wms.rtsp.RTSPResponseMessages resp)
```

Invoked on REDIRECT command

Parameters:

req - RTP request
resp - RTP response

onSetup

```
public void onSetup(com.wowza.wms.rtsp.RTSPRequestMessage req,  
    com.wowza.wms.rtsp.RTSPResponseMessages resp)
```

Invoked on SETUP command

Parameters:

req - RTP request
resp - RTP response

onTeardown

```
public void onTeardown(com.wowza.wms.rtsp.RTSPRequestMessage req,  
    com.wowza.wms.rtsp.RTSPResponseMessages resp)
```

Invoked on TEARDOWN command

Parameters:

req - RTP request
resp - RTP response

getIdleFrequency

```
public int getIdleFrequency()
```

Get idle frequency (milliseconds)

Returns:

idle frequency (milliseconds)

setIdleFrequency

```
public void setIdleFrequency(int idleFrequency)
```

Set idle frequency (milliseconds)

Parameters:

idleFrequency - idle frequency (milliseconds)

isDebugRTSPSession

```
public boolean isDebugRTSPSession()
```

True if debugging RTSP session

Returns:

true if debugging RTSP session

setDebugRTSPSession

```
public void setDebugRTSPSession(boolean debugRTSPSession)
```

Set debugging RTSP session

Parameters:

debugRTSPSession - true if debugging RTSP session

getRTSPTunnelingSessionId

```
public String getRTSPTunnelingSessionId()
```

Get the RTSP/RTP tunneling session id

Returns:

RTSP/RTP tunneling session id

setRTSPTunnelingSessionId

```
public void setRTSPTunnelingSessionId(String rtspTunnelingSessionId)
```

Set the RTSP/RTP tunneling session id

Parameters:

rtspTunnelingSessionId - RTSP/RTP tunneling session id

getRTSPPlayRangeStart

```
public double getRTSPPlayRangeStart()
```

Get play start range, Internal use.

Returns:

start range

setRTSPPlayRangeStart

```
public void setRTSPPlayRangeStart(double rtspPlayRangeStart)
```

Set play start range, Internal use.

Parameters:

rtspPlayRangeStart - start range

getRTSPPlayRangeStop

```
public double getRTSPPlayRangeStop()
```

Get play stop range, Internal use.

Returns:

stop range

setRTSPPlayRangeStop

```
public void setRTSPPlayRangeStop(double rtspPlayRangeStop)
```

(continued from last page)

Set play stop range, Internal use.

Parameters:

rtspPlayRangeStop - stop range

getCookieStr

```
public String getCookieStr()
```

Get cookie string

Returns:

cookie string

setCookieStr

```
public void setCookieStr(String cookieStr)
```

Set cooking string

Parameters:

cookieStr - cooking string

getIOPerformanceCounter

```
public IOPerformanceCounter getIOPerformanceCounter()
```

Get IO performance counter

Returns:

IO performance counter

setIOPerformanceCounter

```
public void setIOPerformanceCounter(IOPerformanceCounter ioPerformanceCounter)
```

Set IO performance counter

Parameters:

ioPerformanceCounter - IO performance counter

isShutdownClient

```
public boolean isShutdownClient()
```

Is RTP session shutdown

Returns:

true if shutdown

setShutdownClient

```
public void setShutdownClient(boolean shutdownClient)
```

Set RTP session shutdown

Parameters:

shutdownClient - true if shutdown

getAudioPacketizerItem

```
public RTPPacketizerItem getAudioPacketizerItem(RTPContext rtpContext,  
        IApplicationInstance appInstance,  
        int codecId)
```

Get audio packetizer for a given codec id.

Parameters:

rtpContext - RTP context
appInstance - application instance
codecId - codec id

Returns:

packetizer info

getVideoPacketizerItem

```
public RTPPacketizerItem getVideoPacketizerItem(RTPContext rtpContext,  
        IApplicationInstance appInstance,  
        int codecId)
```

Get video packetizer for a given codec id.

Parameters:

rtpContext - RTP context
appInstance - application instance
codecId - codec id

Returns:

packetizer info

getStreamPacketizerItem

```
public RTPPacketizerItem getStreamPacketizerItem(RTPContext rtpContext,  
        IApplicationInstance appInstance,  
        int codecId)
```

Get stream packetizer for a given codec id.

Parameters:

rtpContext - RTP context
appInstance - application instance
codecId - codec id

Returns:

packetizer info

putAudioPacketizerItem

```
public void putAudioPacketizerItem(int codecId,  
        String classPath)
```

Set the audio packetizer for a given codec id

Parameters:

codecId - codec id
classPath - class path

putVideoPacketizerItem

```
public void putVideoPacketizerItem(int codecId,  
    String classPath)
```

Set the video packetizer for a given codec id

Parameters:

codecId - codec id
classPath - class path

putStreamPacketizerItem

```
public void putStreamPacketizerItem(int codecId,  
    String classPath)
```

Set the stream packetizer for a given codec id

Parameters:

codecId - codec id
classPath - class path

getElapsedTime

```
public ElapsedTimer getElapsedTime()
```

Get the elapsed timer for this RTP session

Returns:

elapsed timer

getTimeRunning

```
public String getTimeRunning()
```

Get the time running for this RTP session

Returns:

time running as a string

getTimeRunningSeconds

```
public double getTimeRunningSeconds()
```

Get the number of second this RTP session has been running

Returns:

number of second this RTP session has been running

isRedirectSession

```
public boolean isRedirectSession()
```

(continued from last page)

setRedirectSession

```
public void setRedirectSession(boolean redirectSession)
```

getRedirectSessionCode

```
public int getRedirectSessionCode()
```

setRedirectSessionCode

```
public void setRedirectSessionCode(int redirectSessionCode)
```

getRedirectSessionURL

```
public String getRedirectSessionURL()
```

setRedirectSessionURL

```
public void setRedirectSessionURL(String redirectSessionURL)
```

redirectSession

```
public void redirectSession(String redirectSessionURL)
```

redirectSession

```
public void redirectSession(String redirectSessionURL,  
    int redirectSessionCode)
```

getRedirectSessionMessage

```
public String getRedirectSessionMessage()
```

setRedirectSessionMessage

```
public void setRedirectSessionMessage(String redirectSessionMessage)
```

getIoSession

```
public org.apache.mina.common.support.BaseIoSession getIoSession()
```

(continued from last page)

setIoSession

```
public void setIoSession(org.apache.mina.common.support.BaseIoSession ioSession)
```

doIdle

```
public void doIdle()
```

com.wowza.wms.rtp.model

Class RTPSessions

java.lang.Object

└-com.wowza.wms.rtp.model.RTPSessions

public class **RTPSessions**
extends Object

RTPSessions: collection of RTP sessions

Constructor Summary

public	RTPSessions (IVHost vhost) Constructor
--------	--

Method Summary

RTPSession	addSession (RTPSession session) Add RTP session
void	addSessionListener (IRTPSessionNotify listener) Add a RTP session listener
String	getNextSessionId () Get next RTP session id for new session
RTPSession	getSession (String sessionId) Get RTP session by session id
java.util.List	getSessionIds () Get list of current RTP session ids
IVHost	getVHost () Get vhost
void	notifySessionCreate (IApplicationInstance appInstance, RTPSession rtpSession) Notify session create
void	notifySessionCreate (RTPSession rtpSession) Notify session create
void	notifySessionDestroy (IApplicationInstance appInstance, RTPSession rtpSession) Notify session destroy
void	notifySessionDestroy (RTPSession rtpSession) Notify session destroy
void	releaseSessionId (String sessionIdStr) Release an RTP session id

RTPSession	removeSession (RTPSession session) Remove RTP session by object
RTPSession	removeSession (String sessionId) Remove RTP session by session id
void	removeSessionListener (IRTPSessionNotify listener) Remove an RTP session listener

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

RTPSessions

```
public RTPSessions(IVHost vhost)
```

Constructor

Parameters:

vhost - vhost

Methods

getNextSessionId

```
public String getNextSessionId()
```

Get next RTP session id for new session

Returns:

RTP session id

getSessionIds

```
public java.util.List getSessionIds()
```

Get list of current RTP session ids

Returns:

list of current RTP session ids

releaseSessionId

```
public void releaseSessionId(String sessionIdStr)
```

Release an RTP session id

Parameters:

sessionIdStr - RTP session id

(continued from last page)

getSession

```
public RTPSession getSession(String sessionId)
```

Get RTP session by session id

Parameters:

sessionId - session id

Returns:

RTP session

removeSession

```
public RTPSession removeSession(String sessionId)
```

Remove RTP session by session id

Parameters:

sessionId - session id

Returns:

RTP session if removed

removeSession

```
public RTPSession removeSession(RTPSession session)
```

Remove RTP session by object

Parameters:

session - RTP session

Returns:

RTP session if removed

addSession

```
public RTPSession addSession(RTPSession session)
```

Add RTP session

Parameters:

session - RTP session

Returns:

RTP session added

getVHost

```
public IVHost getVHost()
```

Get vhost

Returns:

vhost

(continued from last page)

addSessionListener

```
public void addSessionListener(IRTPSessionNotify listener)
```

Add a RTP session listener

Parameters:

listener - RTP session listener

removeSessionListener

```
public void removeSessionListener(IRTPSessionNotify listener)
```

Remove an RTP session listener

Parameters:

listener - RTP session listener

notifySessionCreate

```
public void notifySessionCreate(IApplicationInstance appInstance,  
    RTPSession rtpSession)
```

Notify session create

Parameters:

appInstance - application instance

rtpSession - RTP session

notifySessionDestroy

```
public void notifySessionDestroy(IApplicationInstance appInstance,  
    RTPSession rtpSession)
```

Notify session destroy

Parameters:

appInstance - application instance

rtpSession - RTP session

notifySessionCreate

```
public void notifySessionCreate(RTPSession rtpSession)
```

Notify session create

Parameters:

rtpSession - RTP session

notifySessionDestroy

```
public void notifySessionDestroy(RTPSession rtpSession)
```

Notify session destroy

Parameters:

rtpSession - RTP session

com.wowza.wms.rtp.model Class RTPStream

java.lang.Object

└--com.wowza.wms.rtp.model.RTPStream

public class **RTPStream**
extends Object

RTPStream: RTP Stream class which encapsulates an IMediaStream class for RTP streaming.

Field Summary

public static final	AVSYNCMETHODS_RTPTIMECODE Value: 3
public static final	AVSYNCMETHODS_SENDERREPORT Value: 1
public static final	AVSYNCMETHODS_SYSTEMCLOCK Value: 2
public static final	AVSYNCMETHODS_UNKNOWN Value: 0
public static final	DEFAULT_RTPIGNOREPROFILELEVELID Value: false
public static final	DEFAULT_RTPIGNORESPROPPARAMETERSETS Value: false
public static final	DEFAULT_RTSPSESSIONNAME Value:
public static final	MODE_PLAY Value: 1
public static final	MODE_PUBLISH Value: 2
public static final	MODE_UNKNOWN Value: 0
public static final	SDPLOCATION_AUDIO Value: 1

public static final	<u>SDPLOCATION_STREAM</u> Value: 0
public static final	<u>SDPLOCATION_VIDEO</u> Value: 2
public static final	<u>STREAMINFO_SESSIONATTRIBUTES</u> Value: attributes
public static final	<u>STREAMINFO_SESSIONBANDWIDTH</u> Value: bandwidth
public static final	<u>STREAMINFO_SESSIONCONNECTIONDATA</u> Value: connectiondata
public static final	<u>STREAMINFO_SESSIONEMAILADDRESS</u> Value: emailaddress
public static final	<u>STREAMINFO_SESSIONINFORMATION</u> Value: information
public static final	<u>STREAMINFO_SESSIONNAME</u> Value: name
public static final	<u>STREAMINFO_SESSIONPHONENUMBER</u> Value: phonenummer
public static final	<u>STREAMINFO_SESSIONPORIGIN</u> Value: origin
public static final	<u>STREAMINFO_SESSIONPROTOCOLVERSION</u> Value: protocolversion
public static final	<u>STREAMINFO_SESSIONREPEATTIMES</u> Value: repeattimes
public static final	<u>STREAMINFO_SESSIONTIMEZONES</u> Value: timezones
public static final	<u>STREAMINFO_SESSIONTIMING</u> Value: timing
public static final	<u>STREAMINFO_SESSIONURI</u> Value: uri

Constructor Summary

public	<u>RTPStream</u> (<u>RTPContext</u> context, String streamId) Constructor
--------	---

Method Summary

void	<u>addStreamAttribute</u> (String key, String value) Add a name value pair to the stream attribute collection
void	<u>addStreamInfo</u> (String key, String value) Add a name value pair to the stream info collection
void	<u>addTrack</u> (RTPTrack track) Add a track
void	<u>addTrackId</u> (String seq, String trackId) Add track
void	<u>addTrackInternal</u> (RTPTrack track) Add RTP track
void	<u>announce</u> (RTPRequestStatus status) Execute announce command
void	<u>attachToWMSSession</u> (RtmpSessionInfo wmsSessionInfo) Attach to WMS session, Internal use
static int	<u>avSyncNameToId</u> (String avSyncName) Get audio/video sync id from name
void	<u>checkSendMetadata</u> (long adjTimecode, RTPTrack rtpTrack) Check to see if we have sent onMetadata event, if not send
void	<u>clearRTSPSessionExtraLines</u> () Clear SDP extra lines
void	<u>clearTracks</u> () Clear all tracks
boolean	<u>createStream</u> (RTPRequestStatus status)
boolean	<u>createStream</u> (RTPRequestStatus status, <u>IApplicationInstance</u> appInstance) Create IMediaStream, Internal use.
String	<u>describe</u> (RTPSession rtspSession, int isStreamPacketizer, RTPRequestStatus status) Execute describe command
String	<u>describe</u> (<u>RTPSession</u> rtspSession, RTPRequestStatus status) Execute describe command
void	<u>detachFromWMSSession</u> (RtmpSessionInfo wmsSessionInfo) Detach from WMS session, Internal use.
void	<u>extractCodecConfigFromTrackInfo</u> () Extract codec config information from SDP data
String	<u>formatRTPInfo</u> (long timecode, int videoSeq, int audioSeq) Format RTP info
<u>IApplicationInstance</u>	<u>getAppInstance</u> () Get application instance

String	<u>getAppInstanceName()</u> Get the application instance name
String	<u>getAppName()</u> Get the application name
RTPTrack	<u>getAudioTrack()</u> Get the most likely audio track
int	<u>getAutoAllocateInterleavePorts()</u>
int	<u>getAVSyncMethod()</u> Get the audio/video sync method.
double	<u>getDuration()</u> Get the duration of the stream if video on demand
String	<u>getHost()</u> Get host
<u>AMFPacket[]</u>	<u>getLastPacketsByType()</u> (<u>IMediaReader</u> localReader, double startTime) Analyzes stream to get information, Internal use.
<u>AMFPacket[]</u>	<u>getLastPacketsByType()</u> (<u>IMediaStream</u> localStream) Analyzes stream to get information, Internal use.
int	<u>getMaxRTCPWaitTime()</u> Get max time to wait for RTCP sender reports (milliseconds)
String	<u>getMediaCasterType()</u> Get the media caster stream type for this stream
<u>IMediaReader</u>	<u>getMediaReader()</u> Get media reader if video on demand stream
byte[]	<u>getMetadataPacket()</u> Get the onMetadata packet for this RTP stream
int	<u>getMode()</u> Get the current play/publish mode
int	<u>getMPEGTSAudioBitrate()</u>
String	<u>getMPEGTSAudioLanguage()</u>
int	<u>getMPEGTSAudioPID()</u> Get the audio PID id if MPEG-TS stream
int	<u>getMPEGTSProgramID()</u>
int	<u>getMPEGTSVideoBitrate()</u>
int	<u>getMPEGTSVideoPID()</u> Get the video PID id if MPEG-TS stream
long	<u>getNormalizedNTPTimecode()</u> (long timecode) Turn a millisecond timcode into an NTP timecode

String	<u>getOutHost()</u> Get the out host
<u>RTPContext</u>	<u>getRTPContext()</u> Get the RTP context
<u>RTPDestination</u>	<u>getRTPDestination()</u> Get RTP destination
RTPStream.RTPInfo	<u>getRTPInfo(double startTime, int videoSeq, int audioSeq)</u> Get the RTP info
RTPStreamContext	<u>getRTPStreamContext()</u>
String	<u>getRTSPBindIpAddress()</u> Get the bind RTSP bind IP address
String	<u>getRTSPConnectionAddressType()</u> Get the connection address type
String	<u>getRTSPConnectionIpAddress()</u> Get the connection IP address
int	<u>getRTSPMaximumPendingWriteBytes()</u> Get the maximum number of waiting bytes allow for this RTSP session
String	<u>getRTSPOriginAddressType()</u> Get the origin address type
String	<u>getRTSPOriginIpAddress()</u> Get the origin IP address
String	<u>getRTSPSessionDescription()</u> Get session description
java.util.List	<u>getRTSPSessionExtraLines()</u> Get extra SDP lines
String	<u>getRTSPSessionName()</u> Get session name
int	<u>getRTSPSessionTimeout()</u> Get RTP session timeout (milliseconds)
String	<u>getSDPLang()</u> Get the SDP language
<u>RTPSession</u>	<u>getSession()</u> Get the RTP session
<u>IMediaStream</u>	<u>getStream()</u> Get the IMediaStream
java.util.Map	<u>getStreamAttributes()</u> Get all name/value pairs in the stream attributes collection
String	<u>getStreamExt()</u> Get the stream extension

String	<u>getStreamId()</u> Get the stream id
java.util.Map	<u>getStreamInfo()</u> Get all name/value pairs in the stream info collection
String	<u>getStreamInfo(String key)</u> Get stream info by name
Object	<u>getStreamLock()</u> Get the synchronization lock for this stream
String	<u>getStreamName()</u> Get stream name
String	<u>getStreamNameLogging()</u> Get the stream name used for logging
String	<u>getStreamQueryStr()</u> Get the stream query string
long	<u>getStreamSessionId()</u> Get the stream session id
String	<u>getStreamSessionIp()</u> Get the stream session ip
long	<u>getStreamSessionVersion()</u> Get the stream session version
RTPTrack	<u>getStreamTrack()</u> Get the most likely stream track
String	<u>getStreamType()</u> Get the stream type
RTPTrack	<u>getTrack(String trackId)</u> Get track by id
String	<u>getTrackId(String seq)</u> Get track by sequence number
java.util.List	<u>getTrackNames()</u> Get a list of track ids
String	<u>getTransportMode()</u> Get the transport mode
int	<u>getUDPManagedDeliveryCount()</u>
int	<u>getUDPManagedDeliveryDelay()</u>
com.wowza.wms.rtp.transport.IUDPTransport	<u>getUDPTransport(boolean isMulticast)</u> Get the UDP transport for this stream
<u>IVHost</u>	<u>getVHost()</u> Get vhost

RTPTrack	<u>getVideoTrack()</u> Get the most likely video track
long	<u>getVODLastTimeTC()</u> Get the last timecode (milliseconds) sent for video on demand
long	<u>getVODPlayLen()</u> Get the video on demand play duration (milliseconds)
long	<u>getVODStartTimeTC()</u> Get the video on demand start time (milliseconds)
void	<u>idle()</u> (org.apache.mina.common.IoSession session, RtmpResponseMessage resp) Process idle event
void	<u>incrementMediaInBytes()</u> (long bytes) Increment the media bytes in, Internal use.
void	<u>initProperties()</u> (<u>IApplicationInstance</u> appInstance)
boolean	<u>isAVSyncNonSR()</u> Is sync method based on RTCP packets (sender report)
boolean	<u>isBlockUDPOut()</u>
boolean	<u>isCheckIpAddr()</u> Are we checking the ip address of each incoming RTP packet
boolean	<u>isCheckSSRC()</u> Are we checking the ssrc values of each incoming RTP packet
boolean	<u>isForceMPEGTSOut()</u>
boolean	<u>isForceRTSPInterleaved()</u> True if forcing RTSP interleaved
boolean	<u>isLive()</u> Is live stream
boolean	<u>isModePlay()</u> Is this a play stream
boolean	<u>isModePublish()</u> Is this a publish stream
boolean	<u>isModeUnknown()</u> Is the stream mode unknown (publish vs play)
boolean	<u>isMPEGTSOut()</u> Is MPEG-TS out
boolean	<u>isPaused()</u> Is stream paused
boolean	<u>isPublishStreamReady()</u> See if a publishing stream has enough data to start playback

boolean	<u>isRefuseIncomingUDPConnections()</u>
boolean	<u>isRefuseOutgoingUDPConnections()</u>
boolean	<u>isResetPlayStream()</u> Is reset stream trigger, Internal use.
boolean	<u>isResyncAudioVideoOnSR()</u> Reset audio/video sync on new RTCP packets (not just first packet)
boolean	<u>isRTPIgnoreProfileLevelId()</u>
boolean	<u>isRTPIgnoreSPropParameterSets()</u>
boolean	<u>isRTSP()</u> Is this RTP Stream managed by RTSP session
boolean	<u>isRTSPAlwaysUseSDPPorts()</u> Force RTSP to use ports in SDP data
boolean	<u>isRTSPPull()</u> Is this RTP Stream managed by RTSP session
boolean	<u>isSendSDESEvents()</u> Send RTCP SDES events
boolean	<u>isStreamStarted()</u> Is stream started
boolean	<u>isTimeout()</u> (long currTime, int timeout) Is the stream timeout out
void	<u>lockRepeaterStreams()</u> (java.util.List streamNames) Lock a list of live repeater stream names, Internal use.
void	<u>pause()</u> (RTPRequestStatus status) Execute pause
RTPStreamPlayResult	<u>play()</u> (RTPRequestStatus status) Execute play
RTPStreamPlayResult	<u>play()</u> (RTPRequestStatus status, double startTime, double stopTime) Execute play
void	<u>putRTSPSessionExtraLine()</u> (int location, String line) Add an extra line to the SDP data
RTPStreamPlayResult	<u>record()</u> (RTPRequestStatus status) Execute record
RTPStreamPlayResult	<u>record()</u> (RTPRequestStatus status, double startTime, double stopTime) Execute record
RTPTrack	<u>removeTrack()</u> (String trackId) Remove a track by id
RTPTrack	<u>removeTrackInternal()</u> (String trackId) Remove track by track id

void	<u>resetSentMetadataFlag()</u> Reset sendMetadata flag
void	<u>setAppInstanceName(String appInstanceName)</u> Set the application instance name
void	<u>setAppName(String appName)</u> Get the application name
void	<u>setAVSyncMethod(int avSyncMethod)</u> Set the audio/video sync method.
void	<u>setBlockUDPOut(boolean blockUDPOut)</u>
void	<u>setCheckIpAddr(boolean checkIpAddr)</u> Are we checking the ip address of each incoming RTP packet
void	<u>setCheckSSRC(boolean checkSSRC)</u> Are we checking the ssrc values of each incoming RTP packet
void	<u>setForceMPEGTSOut(boolean isForceMPEGTSOut)</u>
void	<u>setForceRTSPInterleaved(boolean isForceRTSPInterleaved)</u> True if forcing RTSP interleaved
void	<u>setHost(String host)</u> Set host
void	<u>setLive(boolean isLive)</u> Is live stream
void	<u>setMaxRTCPWaitTime(int maxRTCPWaitTime)</u> Set max time to wait for RTCP sender reports (milliseconds)
void	<u>setMode(int mode)</u> Set the play/publish mode
void	<u>setMPEGTSAudioBitrate(int mpegtsAudioBitrate)</u>
void	<u>setMPEGTSAudioLanguage(String mpegtsAudioLanguage)</u>
void	<u>setMPEGTSAudioPID(int mpegtsAudioPID)</u> Set the audio PID id if MPEG-TS stream
void	<u>setMPEGTSOut(boolean isMPEGTSOut)</u> Is MPEG-TS out
void	<u>setMPEGTSProgramID(int mpegtsProgramId)</u>
void	<u>setMPEGTSVideoBitrate(int mpegtsVideoBitrate)</u>
void	<u>setMPEGTSVideoPID(int mpegtsVideoPID)</u> Set the video PID id if MPEG-TS stream
void	<u>setOutHost(String outHost)</u> Set out host

void	<u>setRefuseIncomingUDPConnections</u> (boolean refuseIncomingUDPConnections) Set refuse incoming UDP connections
void	<u>setRefuseOutgoingUDPConnections</u> (boolean refuseOutgoingUDPConnections) Set refuse outgoing UDP connections
void	<u>setResetPlayStream</u> (boolean doResetPlayStream) Set reset stream trigger
void	<u>setResyncAudioVideoOnSR</u> (boolean resyncAudioVideoOnSR) Reset audio/video sync on new RTCP packets (not just first packet)
void	<u>setRTPDestination</u> (<u>RTPDestination</u> rtpDestination) Set RTP destination
void	<u>setRTPIgnoreProfileLevelId</u> (boolean rtpIgnoreProfileLevelId) Set RTP ignore profile level id
void	<u>setRTPIgnoreSPropParameterSets</u> (boolean rtpIgnoreSPropParameterSets) Set RTP ignore SProp parameter sets
void	<u>setRTPStreamContext</u> (<u>RTPStreamContext</u> rtpStreamContext) Set RTP stream context
void	<u>setRTSP</u> (boolean isRTSP) Is this RTP Stream managed by RTSP session
void	<u>setRTSPAlwaysUseSDPPorts</u> (boolean rtspAlwaysUseSDPPorts) Force RTSP to use ports in SDP data
void	<u>setRTSPBindIpAddress</u> (String rtspBindIpAddress) Set the bind RTSP bind IP address
void	<u>setRTSPConnectionAddressType</u> (String rtspConnectionAddressType) Set the connection address type
void	<u>setRTSPConnectionIpAddress</u> (String rtspConnectionIpAddress) Set the connection IP address
void	<u>setRTSPMaximumPendingWriteBytes</u> (int rtspMaximumPendingWriteBytes) Set the maximum number of waiting bytes allow for this RTSP session
void	<u>setRTSPOriginAddressType</u> (String rtspOriginAddressType) Set the origin address type
void	<u>setRTSPOriginIpAddress</u> (String rtspOriginIpAddress) Set the origin IP address
void	<u>setRTSPPull</u> (boolean isRTSPPull) Is this RTP Stream managed by RTSP session
void	<u>setRTSPSessionDescription</u> (String rtspSessionDescription) Get session description
void	<u>setRTSPSessionName</u> (String rtspSessionName) Set session name
void	<u>setRTSPSessionTimeout</u> (int rtspSessionTimeout) Set RTP session timeout (milliseconds)
void	<u>setSDPLang</u> (String sdpLang) Set the SDP language

void	<u>setSendSDESEvents</u> (boolean sendSDESEvents) Send RTCP SDES events
void	<u>setSession</u> (<u>RTPSession</u> session) Set the RTP session
void	<u>setStreamExt</u> (String streamExt) Set the stream extension
void	<u>setStreamName</u> (String streamName) Set stream name
void	<u>setStreamNameLogging</u> (String streamNameLogging) Set the stream name used for logging
void	<u>setStreamQueryStr</u> (String streamQueryStr) Set the stream query string
void	<u>setStreamSessionId</u> (long streamSessionId) Set the stream session id
void	<u>setStreamSessionIp</u> (String streamSessionIp) Set the stream session ip
void	<u>setStreamSessionVersion</u> (long streamSessionVersion) Set the stream session version
void	<u>setStreamType</u> (String streamType) Set the stream type
void	<u>setTransportMode</u> (String transportMode) Set the transport mode
void	<u>setUDPManagedDeliveryCount</u> (int udpManagedDeliveryCount)
void	<u>setUDPManagedDeliveryDelay</u> (int udpManagedDeliveryDelay)
void	<u>setVODLastTimeTC</u> (long vodLastTimeTC) Set the last timecode (milliseconds) sent for video on demand
void	<u>setVODPlayLen</u> (long vodPlayLen) Set the video on demand play duration (milliseconds)
void	<u>setVODStartTimeTC</u> (long vodStartTimeTC) Set the video on demand start time (milliseconds)
void	<u>shutdown</u> (RTPRequestStatus status) shutdown RTP stream, Internal use.
RTPTrack	<u>sloppyGetTrack</u> (String trackId) Sloppy method for finding track by name, Internal use.
boolean	<u>streamExists</u> () Return true if RTP stream contains a IMediaStream
void	<u>switchSetupToMPEGTS</u> () Switch a stream to MPEG-TS, Internal use.

void	touch() Touch the stream so that it does not timeout
String	transportFindBestMatch (String transport) Based on a trasport string from SETUP command find best match
void	unlockRepeaterStreams () Unlock live repeater streams

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

SDPLOCATION_STREAM

```
public static final int SDPLOCATION_STREAM
```

Constant value: **0**

SDPLOCATION_AUDIO

```
public static final int SDPLOCATION_AUDIO
```

Constant value: **1**

SDPLOCATION_VIDEO

```
public static final int SDPLOCATION_VIDEO
```

Constant value: **2**

MODE_UNKNOWN

```
public static final int MODE_UNKNOWN
```

Constant value: **0**

MODE_PLAY

```
public static final int MODE_PLAY
```

Constant value: **1**

MODE_PUBLISH

```
public static final int MODE_PUBLISH
```

Constant value: **2**

AVSYNCMETHODS_UNKNOWN

```
public static final int AVSYNCMETHODS_UNKNOWN
```

Constant value: **0**

AVSYNCMETHODS_SENDERREPORT

```
public static final int AVSYNCMETHODS_SENDERREPORT
```

Constant value: **1**

AVSYNCMETHODS_SYSTEMCLOCK

```
public static final int AVSYNCMETHODS_SYSTEMCLOCK
```

Constant value: **2**

AVSYNCMETHODS_RTPTIMECODE

```
public static final int AVSYNCMETHODS_RTPTIMECODE
```

Constant value: **3**

STREAMINFO_SESSIONPROTOCOLVERSION

```
public static final java.lang.String STREAMINFO_SESSIONPROTOCOLVERSION
```

Constant value: **protocolversion**

STREAMINFO_SESSIONPORIGIN

```
public static final java.lang.String STREAMINFO_SESSIONPORIGIN
```

Constant value: **origin**

STREAMINFO_SESSIONNAME

```
public static final java.lang.String STREAMINFO_SESSIONNAME
```

Constant value: **name**

STREAMINFO_SESSIONINFORMATION

```
public static final java.lang.String STREAMINFO_SESSIONINFORMATION
```

Constant value: **information**

(continued from last page)

STREAMINFO_SESSIONURI

```
public static final java.lang.String STREAMINFO_SESSIONURI
```

Constant value: **uri**

STREAMINFO_SESSIONEMAILADDRESS

```
public static final java.lang.String STREAMINFO_SESSIONEMAILADDRESS
```

Constant value: **emailaddress**

STREAMINFO_SESSIONPHONENUMBER

```
public static final java.lang.String STREAMINFO_SESSIONPHONENUMBER
```

Constant value: **phonenumber**

STREAMINFO_SESSIONCONNECTIONDATA

```
public static final java.lang.String STREAMINFO_SESSIONCONNECTIONDATA
```

Constant value: **connectiondata**

STREAMINFO_SESSIONBANDWIDTH

```
public static final java.lang.String STREAMINFO_SESSIONBANDWIDTH
```

Constant value: **bandwidth**

STREAMINFO_SESSIONTIMING

```
public static final java.lang.String STREAMINFO_SESSIONTIMING
```

Constant value: **timing**

STREAMINFO_SESSIONREPEATTIMES

```
public static final java.lang.String STREAMINFO_SESSIONREPEATTIMES
```

Constant value: **repeattimes**

STREAMINFO_SESSIONTIMEZONES

```
public static final java.lang.String STREAMINFO_SESSIONTIMEZONES
```

Constant value: **timezones**

STREAMINFO_SESSIONATTRIBUTES

```
public static final java.lang.String STREAMINFO_SESSIONATTRIBUTES
```

(continued from last page)

Constant value: **attributes**

DEFAULT_RTSPSESSIONNAME

```
public static final java.lang.String DEFAULT_RTSPSESSIONNAME
```

Constant value:

DEFAULT_RTPIGNOREPROFILELEVELID

```
public static final boolean DEFAULT_RTPIGNOREPROFILELEVELID
```

Constant value: **false**

DEFAULT_RTPIGNORESPROPPARAMETERSETS

```
public static final boolean DEFAULT_RTPIGNORESPROPPARAMETERSETS
```

Constant value: **false**

Constructors

RTPStream

```
public RTPStream(RTPContext context,  
                 String streamId)
```

Constructor

Parameters:

context - RTP context
streamId - stream id

Methods

getRTPContext

```
public RTPContext getRTPContext()
```

Get the RTP context

Returns:

RTP context

getStreamLock

```
public Object getStreamLock()
```

Get the synchronization lock for this stream

Returns:

synchronization lock for this stream

(continued from last page)

addTrack

```
public void addTrack(RTPTrack track)
```

Add a track

Parameters:

track - RTP track

removeTrack

```
public RTPTrack removeTrack(String trackId)
```

Remove a track by id

Parameters:

trackId - track id

Returns:

RTP track

addTrackInternal

```
public void addTrackInternal(RTPTrack track)
```

Add RTP track

Parameters:

track - RTP track

removeTrackInternal

```
public RTPTrack removeTrackInternal(String trackId)
```

Remove track by track id

Parameters:

trackId - track id

Returns:

RTP track that was removed

clearTracks

```
public void clearTracks()
```

Clear all tracks

incrementMediaInBytes

```
public void incrementMediaInBytes(long bytes)
```

Increment the media bytes in, Internal use.

Parameters:

bytes

(continued from last page)

sloppyGetTrack

```
public RTPTrack sloppyGetTrack(String trackId)
```

Sloppy method for finding track by name, Internal use.

Parameters:

trackId - track name

Returns:

RTP track

getTrack

```
public RTPTrack getTrack(String trackId)
```

Get track by id

Parameters:

trackId - track id

Returns:

RTP track

addTrackId

```
public void addTrackId(String seq,  
    String trackId)
```

Add track

Parameters:

seq - sequence number

trackId - track id

getTrackId

```
public String getTrackId(String seq)
```

Get track by sequence number

Parameters:

seq - sequence number

Returns:

RTP track

getTrackNames

```
public java.util.List getTrackNames()
```

Get a list of track ids

Returns:

list of track ids

(continued from last page)

getSession

```
public RTPSession getSession()
```

Get the RTP session

Returns:

RTP session

setSession

```
public void setSession(RTPSession session)
```

Set the RTP session

Parameters:

session - RTP session

getStreamId

```
public String getStreamId()
```

Get the stream id

Returns:

stream id

getStream

```
public IMediaStream getStream()
```

Get the IMediaStream

Returns:

stream (IMediaStream) interface

getAudioTrack

```
public RTPTrack getAudioTrack()
```

Get the most likely audio track

Returns:

RTP track

getVideoTrack

```
public RTPTrack getVideoTrack()
```

Get the most likely video track

Returns:

RTP track

getStreamTrack

```
public RTPTrack getStreamTrack()
```

(continued from last page)

Get the most likely stream track

Returns:

RTP track

addStreamInfo

```
public void addStreamInfo(String key,  
    String value)
```

Add a name value pair to the stream info collection

Parameters:

key - name

value - value

getStreamInfo

```
public String getStreamInfo(String key)
```

Get stream info by name

Parameters:

key - name

Returns:

value

addStreamAttribute

```
public void addStreamAttribute(String key,  
    String value)
```

Add a name value pair to the stream attribute collection

Parameters:

key - name

value - value

getVHost

```
public IVHost getVHost()
```

Get vhost

Returns:

vhost

getHost

```
public String getHost()
```

Get host

Returns:

host

(continued from last page)

setHost

```
public void setHost(String host)
```

Set host

Parameters:

host - host

streamExists

```
public boolean streamExists()
```

Return true if RTP stream contains a IMediaStream

Returns:

true if RTP stream contains a IMediaStream

isPublishStreamReady

```
public boolean isPublishStreamReady()
```

See if a publishing stream has enough data to start playback

Returns:

true, if the stream is ready for playback

lockRepeaterStreams

```
public void lockRepeaterStreams(java.util.List streamNames)
```

Lock a list of live repeater stream names, Internal use.

Parameters:

streamNames - stream names

unlockRepeaterStreams

```
public void unlockRepeaterStreams()
```

Unlock live repeater streams

getMediaCasterType

```
public String getMediaCasterType()
```

Get the media caster stream type for this stream

Returns:

media caster stream type

createStream

```
public boolean createStream(RTPRequestStatus status)
```

(continued from last page)

createStream

```
public boolean createStream(RTPRequestStatus status,  
    IApplicationInstance appInstance)
```

Create IMediaStream, Internal use.

Parameters:

status - status

Returns:

true if successful

shutdown

```
public void shutdown(RTPRequestStatus status)
```

shutdown RTP stream, Internal use.

Parameters:

status - status

initProperties

```
public void initProperties(IApplicationInstance appInstance)
```

getAppInstance

```
public IApplicationInstance getAppInstance()
```

Get application instance

Returns:

application instance

isPaused

```
public boolean isPaused()
```

Is stream paused

Returns:

true if stream is paused

getDuration

```
public double getDuration()
```

Get the duration of the stream if video on demand

Returns:

duration in seconds

isResetPlayStream

```
public boolean isResetPlayStream()
```

(continued from last page)

Is reset stream trigger, Internal use.

Returns:

true if reset trigger

setResetPlayStream

```
public void setResetPlayStream(boolean doResetPlayStream)
```

Set reset stream trigger

Parameters:

doResetPlayStream - true if reset trigger

record

```
public RTPStreamPlayResult record(RTPRequestStatus status)
```

Execute record

Parameters:

status - RTP status

Returns:

status

record

```
public RTPStreamPlayResult record(RTPRequestStatus status,  
    double startTime,  
    double stopTime)
```

Execute record

Parameters:

status - RTP status

startTime - playback start time

stopTime - playback stop time

Returns:

status

play

```
public RTPStreamPlayResult play(RTPRequestStatus status)
```

Execute play

Parameters:

status - RTP status

Returns:

status

(continued from last page)

play

```
public RTPStreamPlayResult play(RTPRequestStatus status,  
    double startTime,  
    double stopTime)
```

Execute play

Parameters:

status - RTP status
startTime - playback start time
stopTime - playback stop time

Returns:

status

attachToWMSession

```
public void attachToWMSession(RtmpSessionInfo wmsSessionInfo)
```

Attach to WMS session, Internal use

Parameters:

wmsSessionInfo - session info

detachFromWMSession

```
public void detachFromWMSession(RtmpSessionInfo wmsSessionInfo)
```

Detach from WMS session, Internal use.

Parameters:

wmsSessionInfo - session info

pause

```
public void pause(RTPRequestStatus status)
```

Execute pause

Parameters:

status - RTP status

announce

```
public void announce(RTPRequestStatus status)
```

Execute announce command

Parameters:

status - RTP status

isResyncAudioVideoOnSR

```
public boolean isResyncAudioVideoOnSR()
```

Reset audio/video sync on new RTCP packets (not just first packet)

Returns:

(continued from last page)

true if resetting audio/video on new RTCP packets

setResyncAudioVideoOnSR

```
public void setResyncAudioVideoOnSR(boolean resyncAudioVideoOnSR)
```

Reset audio/video sync on new RTCP packets (not just first packet)

Parameters:

resyncAudioVideoOnSR - true if resetting audio/video on new RTCP packets

getMetadataPacket

```
public byte[] getMetadataPacket()
```

Get the onMetadata packet for this RTP stream

Returns:

onMetadata packet

resetSentMetadataFlag

```
public void resetSentMetadataFlag()
```

Reset sendMetadata flag

checkSendMetadata

```
public void checkSendMetadata(long adjTimecode,  
    RTPTrack rtpTrack)
```

Check to see if we have sent onMetadata event, if not send

Parameters:

adjTimecode - timecode (milliseconds)

extractCodecConfigFromTrackInfo

```
public void extractCodecConfigFromTrackInfo()
```

Extract codec config information from SDP data

getStreamType

```
public String getStreamType()
```

Get the stream type

Returns:

stream type

setStreamType

```
public void setStreamType(String streamType)
```

Set the stream type

Parameters:

(continued from last page)

streamType - stream type

getAppName

```
public String getAppName()
```

Get the application name

Returns:

application name

setAppName

```
public void setAppName(String appName)
```

Get the application name

Parameters:

appName - application name

getAppInstanceName

```
public String getAppInstanceName()
```

Get the application instance name

Returns:

application instance name

setAppInstanceName

```
public void setAppInstanceName(String appInstanceName)
```

Set the application instance name

Parameters:

appInstanceName - application instance name

getStreamExt

```
public String getStreamExt()
```

Get the stream extension

Returns:

stream extension

setStreamExt

```
public void setStreamExt(String streamExt)
```

Set the stream extension

Parameters:

streamExt - stream extension

(continued from last page)

getStreamName

```
public String getStreamName()
```

Get stream name

Returns:

stream name

setStreamName

```
public void setStreamName(String streamName)
```

Set stream name

Parameters:

streamName - stream name

getStreamNameLogging

```
public String getStreamNameLogging()
```

Get the stream name used for logging

Returns:

stream name used for logging

setStreamNameLogging

```
public void setStreamNameLogging(String streamNameLogging)
```

Set the stream name used for logging

Parameters:

streamNameLogging - stream name used for logging

getStreamSessionId

```
public long getStreamSessionId()
```

Get the stream session id

Returns:

stream session id

setStreamSessionId

```
public void setStreamSessionId(long streamSessionId)
```

Set the stream session id

Parameters:

streamSessionId - stream session id

getStreamSessionVersion

```
public long getStreamSessionVersion()
```

(continued from last page)

Get the stream session version

Returns:

stream session version

setStreamSessionVersion

```
public void setStreamSessionVersion(long streamSessionVersion)
```

Set the stream session version

Parameters:

streamSessionVersion - stream session version

getStreamSessionIp

```
public String getStreamSessionIp()
```

Get the stream session ip

Returns:

stream session ip

setStreamSessionIp

```
public void setStreamSessionIp(String streamSessionIp)
```

Set the stream session ip

Parameters:

streamSessionIp - stream session ip

isCheckSSRC

```
public boolean isCheckSSRC()
```

Are we checking the ssrc values of each incoming RTP packet

Returns:

true if checking

setCheckSSRC

```
public void setCheckSSRC(boolean checkSSRC)
```

Are we checking the ssrc values of each incoming RTP packet

Parameters:

checkSSRC - true if checking

isCheckIpAddr

```
public boolean isCheckIpAddr()
```

Are we checking the ip address of each incoming RTP packet

Returns:

true if checking ip address of each incoming RTP packet

setCheckIpAddr

```
public void setCheckIpAddr(boolean checkIpAddr)
```

Are we checking the ip address of each incoming RTP packet

Parameters:

checkIpAddr - true if checking ip address of each incoming RTP packet

getUDPTransport

```
public com.wowza.wms.rtp.transport.IUDPTransport getUDPTransport(boolean isMulticast)
```

Get the UDP transport for this stream

Parameters:

isMulticast - true if multicast

Returns:

UDP transport

getAVSyncMethod

```
public int getAVSyncMethod()
```

Get the audio/video sync method. See AVSYNCMETHODS_*

Returns:

audio/video sync method. See AVSYNCMETHODS_*

setAVSyncMethod

```
public void setAVSyncMethod(int avSyncMethod)
```

Set the audio/video sync method. See AVSYNCMETHODS_*

Parameters:

avSyncMethod - audio/video sync method. See AVSYNCMETHODS_*

isAVSyncNonSR

```
public boolean isAVSyncNonSR()
```

Is sync method based on RTCP packets (sender report)

Returns:

true if based in RTCP sender reports

getMaxRTCPWaitTime

```
public int getMaxRTCPWaitTime()
```

Get max time to wait for RTCP sender reports (milliseconds)

Returns:

max time to wait for RTCP sender reports (milliseconds)

setMaxRTCPWaitTime

```
public void setMaxRTCPWaitTime(int maxRTCPWaitTime)
```

Set max time to wait for RTCP sender reports (milliseconds)

Parameters:

maxRTCPWaitTime - max time to wait for RTCP sender reports (milliseconds)

getStreamInfo

```
public java.util.Map getStreamInfo()
```

Get all name/value pairs in the stream info collection

Returns:

map of name/value pairs

getStreamAttributes

```
public java.util.Map getStreamAttributes()
```

Get all name/value pairs in the stream attributes collection

Returns:

map of name/value pairs

avSyncNameToId

```
public static int avSyncNameToId(String avSyncName)
```

Get audio/video sync id from name

Parameters:

avSyncName - audio/video sync name

Returns:

audio/video sync id, see AVSYNCMETHODS_*

getMPEGTSAudioLanguage

```
public String getMPEGTSAudioLanguage()
```

setMPEGTSAudioLanguage

```
public void setMPEGTSAudioLanguage(String mpegtsAudioLanguage)
```

getMPEGTSProgramID

```
public int getMPEGTSProgramID()
```

setMPEGTSProgramID

```
public void setMPEGTSProgramID(int mpegtsProgramId)
```

getMPEGTSVideoPID

```
public int getMPEGTSVideoPID()
```

Get the video PID id if MPEG-TS stream

Returns:

video PID id if MPEG-TS stream

setMPEGTSVideoPID

```
public void setMPEGTSVideoPID(int mpegtsVideoPID)
```

Set the video PID id if MPEG-TS stream

Parameters:

mpegtsVideoPID - video PID id if MPEG-TS stream

getMPEGTSAudioPID

```
public int getMPEGTSAudioPID()
```

Get the audio PID id if MPEG-TS stream

Returns:

audio PID id if MPEG-TS stream

setMPEGTSAudioPID

```
public void setMPEGTSAudioPID(int mpegtsAudioPID)
```

Set the audio PID id if MPEG-TS stream

Parameters:

mpegtsAudioPID - audio PID id if MPEG-TS stream

isRTSP

```
public boolean isRTSP()
```

Is this RTP Stream managed by RTSP session

Returns:

true if managed by RTSP session

setRTSP

```
public void setRTSP(boolean isRTSP)
```

Is this RTP Stream managed by RTSP session

(continued from last page)

Parameters:`isRTSP` - true if managed by RTSP session

isRTSPPull

```
public boolean isRTSPPull()
```

Is this RTP Stream managed by RTSP session

Returns:

true if managed by RTSP session

setRTSPPull

```
public void setRTSPPull(boolean isRTSPPull)
```

Is this RTP Stream managed by RTSP session

Parameters:`isRTSPPull` - true if managed by RTSP session

getStreamQueryStr

```
public String getStreamQueryStr()
```

Get the stream query string

Returns:

stream query string

setStreamQueryStr

```
public void setStreamQueryStr(String streamQueryStr)
```

Set the stream query string

Parameters:`streamQueryStr` - stream query string

isModePublish

```
public boolean isModePublish()
```

Is this a publish stream

Returns:

true if publish stream

isModePlay

```
public boolean isModePlay()
```

Is this a play stream

Returns:

true if play stream

(continued from last page)

isModeUnknown

```
public boolean isModeUnknown()
```

Is the stream mode unknown (publish vs play)

Returns:

true if stream mode is unknown

getOutHost

```
public String getOutHost()
```

Get the out host

Returns:

out host

setOutHost

```
public void setOutHost(String outHost)
```

Set out host

Parameters:

outHost - out host

idle

```
public void idle(org.apache.mina.common.Session session,  
    RtmpResponseMessage resp)
```

Process idle event

Parameters:

session - io session

resp - idle request

getLastPacketsByType

```
public AMFPacket\[\] getLastPacketsByType(IMediaStream localStream)
```

Analyzes stream to get information, Internal use.

Parameters:

localStream - local stream

Returns:

important packets

getLastPacketsByType

```
public AMFPacket\[\] getLastPacketsByType(IMediaReader localReader,  
    double startTime)
```

Analyzes stream to get information, Internal use.

Parameters:

localReader - media reader

(continued from last page)

startTime - start time

Returns:

important packets

describe

```
public String describe(RTPSession rtspSession,  
    RTPRequestStatus status)
```

Execute describe command

Parameters:

rtspSession - RTP session
status - RTP status

Returns:

describe response

switchSetupToMPEGTS

```
public void switchSetupToMPEGTS()
```

Switch a stream to MPEG-TS, Internal use.

describe

```
public String describe(RTPSession rtspSession,  
    int isStreamPacketizer,  
    RTPRequestStatus status)
```

Execute describe command

Parameters:

rtspSession - RTP session
isStreamPacketizer - true is stream
status - RTP status

Returns:

describe string

getMode

```
public int getMode()
```

Get the current play/publish mode

Returns:

play/publish mode

setMode

```
public void setMode(int mode)
```

Set the play/publish mode

Parameters:

mode - play/publish mode

getTransportMode

```
public String getTransportMode()
```

Get the transport mode

Returns:

transport mode

setTransportMode

```
public void setTransportMode(String transportMode)
```

Set the transport mode

Parameters:

transportMode - transport mode

getNormalizedNTPTimecode

```
public long getNormalizedNTPTimecode(long timecode)
```

Turn a millisecond timecode into an NTP timecode

Parameters:

timecode - timecode (milliseconds)

Returns:

NTP timecode

isForceRTSPInterleaved

```
public boolean isForceRTSPInterleaved()
```

True if forcing RTSP interleaved

Returns:

true if forcing RTSP interleaved

setForceRTSPInterleaved

```
public void setForceRTSPInterleaved(boolean isForceRTSPInterleaved)
```

True if forcing RTSP interleaved

Parameters:

isForceRTSPInterleaved - true if forcing RTSP interleaved

getRTPInfo

```
public RTPStream.RTPInfo getRTPInfo(double startTime,  
    int videoSeq,  
    int audioSeq)
```

Get the RTP info

Parameters:

startTime - start time

(continued from last page)

videoSeq - video sequence number
audioSeq - audio sequence number

Returns:

RTP info string

formatRTPInfo

```
public String formatRTPInfo(long timecode,  
    int videoSeq,  
    int audioSeq)
```

Format RTP info

Parameters:

timecode - timecode
videoSeq - video sequence number
audioSeq - audio sequence number

Returns:

RTP Info string

getRTSPSessionName

```
public String getRTSPSessionName()
```

Get session name

Returns:

session name

setRTSPSessionName

```
public void setRTSPSessionName(String rtspSessionName)
```

Set session name

Parameters:

rtspSessionName - session name

getRTSPSessionDescription

```
public String getRTSPSessionDescription()
```

Get session description

Returns:

session description

setRTSPSessionDescription

```
public void setRTSPSessionDescription(String rtspSessionDescription)
```

Get session description

Parameters:

rtspSessionDescription - session description

(continued from last page)

isSendSDESEvents

```
public boolean isSendSDESEvents()
```

Send RTCP SDES events

Returns:

true if sending RTCP SDES events

setSendSDESEvents

```
public void setSendSDESEvents(boolean sendSDESEvents)
```

Send RTCP SDES events

Parameters:

sendSDESEvents - true if sending RTCP SDES events

isMPEGTSOut

```
public boolean isMPEGTSOut()
```

Is MPEG-TS out

Returns:

true if MPEG-TS out

setMPEGTSOut

```
public void setMPEGTSOut(boolean isMPEGTSOut)
```

Is MPEG-TS out

Parameters:

isMPEGTSOut - true if MPEG-TS out

transportFindBestMatch

```
public String transportFindBestMatch(String transport)
```

Based on a transport string from SETUP command find best match

Parameters:

transport - transport string

Returns:

best match

getAutoAllocateInterleavePorts

```
public int getAutoAllocateInterleavePorts()
```

isLive

```
public boolean isLive()
```

(continued from last page)

Is live stream

Returns:

true if live

setLive

```
public void setLive(boolean isLive)
```

Is live stream

Parameters:

isLive - true if live

getMediaReader

```
public IMediaReader getMediaReader()
```

Get media reader if video on demand stream

Returns:

media reader

getRTSPSessionTimeout

```
public int getRTSPSessionTimeout()
```

Get RTP session timeout (milliseconds)

Returns:

RTP session timeout (milliseconds)

setRTSPSessionTimeout

```
public void setRTSPSessionTimeout(int rtspSessionTimeout)
```

Set RTP session timeout (milliseconds)

Parameters:

rtspSessionTimeout - RTP session timeout (milliseconds)

getRTSPMaximumPendingWriteBytes

```
public int getRTSPMaximumPendingWriteBytes()
```

Get the maximum number of waiting bytes allow for this RTSP session

Returns:

maximum number of waiting bytes allow for this RTSP session

setRTSPMaximumPendingWriteBytes

```
public void setRTSPMaximumPendingWriteBytes(int rtspMaximumPendingWriteBytes)
```

Set the maximum number of waiting bytes allow for this RTSP session

Parameters:

rtspMaximumPendingWriteBytes - maximum number of waiting bytes allow for this RTSP session

isTimeout

```
public boolean isTimeout(long currTime,  
                          int timeout)
```

Is the stream timeout out

Parameters:

currTime - current time
timeout - timeout value

Returns:

true if timed out

touch

```
public void touch()
```

Touch the stream so that it does not timeout

getRTSPBindIpAddress

```
public String getRTSPBindIpAddress()
```

Get the bind RTSP bind IP address

Returns:

bind RTSP bind IP address

setRTSPBindIpAddress

```
public void setRTSPBindIpAddress(String rtspBindIpAddress)
```

Set the bind RTSP bind IP address

Parameters:

rtspBindIpAddress - bind RTSP bind IP address

getRTSPConnectionIpAddress

```
public String getRTSPConnectionIpAddress()
```

Get the connection IP address

Returns:

connection IP address

setRTSPConnectionIpAddress

```
public void setRTSPConnectionIpAddress(String rtspConnectionIpAddress)
```

Set the connection IP address

Parameters:

rtspConnectionIpAddress - connection IP address

(continued from last page)

getRTSPConnectionAddressType

```
public String getRTSPConnectionAddressType()
```

Get the connection address type

Returns:

connection address type

setRTSPConnectionAddressType

```
public void setRTSPConnectionAddressType(String rtspConnectionAddressType)
```

Set the connection address type

Parameters:

rtspConnectionAddressType - connection address type

getRTSPOriginIpAddress

```
public String getRTSPOriginIpAddress()
```

Get the origin IP address

Returns:

origin IP address

setRTSPOriginIpAddress

```
public void setRTSPOriginIpAddress(String rtspOriginIpAddress)
```

Set the origin IP address

Parameters:

rtspOriginIpAddress - origin IP address

getRTSPOriginAddressType

```
public String getRTSPOriginAddressType()
```

Get the origin address type

Returns:

origin address type

setRTSPOriginAddressType

```
public void setRTSPOriginAddressType(String rtspOriginAddressType)
```

Set the origin address type

Parameters:

rtspOriginAddressType - origin address type

getVODStartTimeTC

```
public long getVODStartTimeTC()
```

(continued from last page)

Get the video on demand start time (milliseconds)

Returns:

video on demand start time (milliseconds)

setVODStartTimeTC

```
public void setVODStartTimeTC(long vodStartTimeTC)
```

Set the video on demand start time (milliseconds)

Parameters:

vodStartTimeTC - video on demand start time (milliseconds)

getVODLastTimeTC

```
public long getVODLastTimeTC( )
```

Get the last timecode (milliseconds) sent for video on demand

Returns:

last timecode (milliseconds) sent for video on demand

setVODLastTimeTC

```
public void setVODLastTimeTC(long vodLastTimeTC)
```

Set the last timecode (milliseconds) sent for video on demand

Parameters:

vodLastTimeTC - last timecode (milliseconds) sent for video on demand

getVODPlayLen

```
public long getVODPlayLen( )
```

Get the video on demand play duration (milliseconds)

Returns:

video on demand play duration (milliseconds)

setVODPlayLen

```
public void setVODPlayLen(long vodPlayLen)
```

Set the video on demand play duration (milliseconds)

Parameters:

vodPlayLen - video on demand play duration (milliseconds)

isStreamStarted

```
public boolean isStreamStarted( )
```

Is stream started

Returns:

true if stream has started playback

isRTSPAlwaysUseSDPPorts

```
public boolean isRTSPAlwaysUseSDPPorts()
```

Force RTSP to use ports in SDP data

Returns:

true if using ports in SDP data

setRTSPAlwaysUseSDPPorts

```
public void setRTSPAlwaysUseSDPPorts(boolean rtspAlwaysUseSDPPorts)
```

Force RTSP to use ports in SDP data

Parameters:

rtspAlwaysUseSDPPorts - true if using ports in SDP data

putRTSPSessionExtraLine

```
public void putRTSPSessionExtraLine(int location,  
String line)
```

Add an extra line to the SDP data

Parameters:

location - location of line, see SDPLOCATION_
line - line to add

clearRTSPSessionExtraLines

```
public void clearRTSPSessionExtraLines()
```

Clear SDP extra lines

getRTSPSessionExtraLines

```
public java.util.List getRTSPSessionExtraLines()
```

Get extra SDP lines

Returns:

extra SDP lines

getRTPDestination

```
public RTPDestination getRTPDestination()
```

Get RTP destination

Returns:

RTP destination

setRTPDestination

```
public void setRTPDestination(RTPDestination rtpDestination)
```

(continued from last page)

Set RTP destination

Parameters:

rtpDestination - RTP destination

getSDPLang

```
public String getSDPLang()
```

Get the SDP language

Returns:

SDP language

setSDPLang

```
public void setSDPLang(String sdpLang)
```

Set the SDP language

Parameters:

sdpLang - SDP language

isRTPIgnoreProfileLevelId

```
public boolean isRTPIgnoreProfileLevelId()
```

setRTPIgnoreProfileLevelId

```
public void setRTPIgnoreProfileLevelId(boolean rtpIgnoreProfileLevelId)
```

getUDPMangedDeliveryDelay

```
public int getUDPMangedDeliveryDelay()
```

setUDPMangedDeliveryDelay

```
public void setUDPMangedDeliveryDelay(int udpManagedDeliveryDelay)
```

getUDPMangedDeliveryCount

```
public int getUDPMangedDeliveryCount()
```

setUDPMangedDeliveryCount

```
public void setUDPMangedDeliveryCount(int udpManagedDeliveryCount)
```

isForceMPEGTSOut

```
public boolean isForceMPEGTSOut()
```

setForceMPEGTSOut

```
public void setForceMPEGTSOut(boolean isForceMPEGTSOut)
```

isBlockUDPOut

```
public boolean isBlockUDPOut()
```

setBlockUDPOut

```
public void setBlockUDPOut(boolean blockUDPOut)
```

getMPEGTSAudioBitrate

```
public int getMPEGTSAudioBitrate()
```

setMPEGTSAudioBitrate

```
public void setMPEGTSAudioBitrate(int mpegtsAudioBitrate)
```

getMPEGTSVideoBitrate

```
public int getMPEGTSVideoBitrate()
```

setMPEGTSVideoBitrate

```
public void setMPEGTSVideoBitrate(int mpegtsVideoBitrate)
```

isRTPIgnoreSPropParameterSets

```
public boolean isRTPIgnoreSPropParameterSets()
```

setRTPIgnoreSPropParameterSets

```
public void setRTPIgnoreSPropParameterSets(boolean rtpIgnoreSPropParameterSets)
```

(continued from last page)

getRTPStreamContext

```
public RTPStreamContext getRTPStreamContext()
```

setRTPStreamContext

```
public void setRTPStreamContext(RTPStreamContext rtpStreamContext)
```

isRefuseIncomingUDPConnections

```
public boolean isRefuseIncomingUDPConnections()
```

setRefuseIncomingUDPConnections

```
public void setRefuseIncomingUDPConnections(boolean refuseIncomingUDPConnections)
```

isRefuseOutgoingUDPConnections

```
public boolean isRefuseOutgoingUDPConnections()
```

setRefuseOutgoingUDPConnections

```
public void setRefuseOutgoingUDPConnections(boolean refuseOutgoingUDPConnections)
```

Package

com.wowza.wms.server

com.wowza.wms.server

Interface ICommandInterfaceCommand

public interface **ICommandInterfaceCommand**
extends

Method Summary	
boolean	canHandle (CommandInterfaceRequestMessage req)
void	invoke (CommandInterfaceRequestMessage req, CommandInterfaceResponseMessage resp)

Methods

invoke

public void **invoke**(CommandInterfaceRequestMessage req,
CommandInterfaceResponseMessage resp)

canHandle

public boolean **canHandle**(CommandInterfaceRequestMessage req)

com.wowza.wms.server Interface ICompilerDirectives

public interface **ICompilerDirectives**
extends

Field Summary

public static final

[RTMPENABLED](#)

Value: **true**

Fields

RTMPENABLED

public static final boolean **RTMPENABLED**

Constant value: **true**

com.wowza.wms.server Interface IResponseListener

public interface **IResponseListener**
extends

IResponseListener: Internal use.

Method Summary

void	onResponseWriteStart (RtmpResponseMessage response)
void	onResponseWriteStop (RtmpResponseMessage response)

Methods

onResponseWriteStart

public void **onResponseWriteStart**(RtmpResponseMessage response)

onResponseWriteStop

public void **onResponseWriteStop**(RtmpResponseMessage response)

com.wowza.wms.server Interface IServer

All Known Implementing Classes:
[Server](#)

public interface **IServer**
extends

IServer: public interface to Server object.

Field Summary

public static final	SERVER_DEFAULT Value: _defaultServer_
---------------------	---

Method Summary

boolean	addAuthorizedRESTAPI (Class classType) Returns true if the class has been added to the REST API.
boolean	addAuthorizedRESTAPI (Class classType, String url) Returns true if the class has been added to the REST API at the specified URL
void	addServerListener (IServerNotify serverListener) Add server listener
java.util.List	getAdminInterfaceObjectList () Get the list of objects exposed through JMX interface
RandomIdGenerator	getClientIdGenerator () Get the client id generator for the server
CommandInterfaceCommandHandler	getCommandInterfaceCommandHandler () Get the command interface command handler
HostPort	getCommandInterfaceHostPort () Get the definition of the command interface
ConnectionCounter	getConnectionCounter () Get the server connection counter.
ConnectionCounterSimple	getConnectionCounter (int counterIndex) Get the server connection counter for a specific technology (see IVHost.COUNTER_*)
long	getConnectionsMaximum () Returns maximum number of connections allowed to the server (-1 means unlimited)
int	getCoreHandlerPoolSize () Get the handler core thread pool size.
int	getCoreTransportPoolSize () Get the transport core thread pool size.

String	<u>getDateStarted()</u> Get the date and time the server was started.
String	<u>getDefaultStreamPrefix()</u> Get the default stream prefix used if the stream name does not include a prefix (default is mp4)
java.util.Properties	<u>getDynamicLogProperties()</u> Get the dynamic log properties defined at the server level in conf/log4j.properties
<u>ThreadPool</u>	<u>getHandlerThreadPool()</u> Get the server handler thread pool.
String	<u>getHTTPHeaderServer()</u> Get the Server header used for HTTP responses
<u>IOPerformanceCounter</u>	<u>getIoPerformanceCounter()</u> Get the server performance counter.
<u>IOPerformanceCounter</u>	<u>getIoPerformanceCounter(int counterIndex)</u> Get the server performance counter for a specific technology (see IVHost.COUNTER_*)
long	<u>getLiveStreamTranscoderSessionCount()</u> Get the current number of live stream transcoder sessions.
<u>WMSProperties</u>	<u>getManagerProperties()</u> Get server's manager's properties collection
<u>WMSProperties</u>	<u>getProperties()</u> Get server level properties collection
long	<u>getPublishersMaximum()</u> Get the maximum number of streams that can be published to the server
String	<u>getRTMPTHeaderServer()</u> Get the Server header used for RTMPT responses
<u>ThreadPool</u>	<u>getThreadPool()</u> Get the server handler thread pool.
String	<u>getTimeRunning()</u> Get a formatted String of how long the server has been running.
double	<u>getTimeRunningSeconds()</u> Get time running in seconds
long	<u>getTranscoderLicenseInUse()</u> Get the total number of transcoder licenses in use.
long	<u>getTranscoderLicenseTotal()</u> Get the total number of transcoder licenses (-1 means unlimited - subscription)
<u>ThreadPool</u>	<u>getTransportThreadPool()</u> Get the server transport thread pool.
ServerTuningHandler	<u>getTuningHandler()</u> Get the handler that control automatic server tuning
com.wowza.wms.transpo rt.udp.UDPPortManager	<u>getUDPPortManager()</u> Get the UDP port manager which manages the allocation of incoming UDP port binding to be sure there are not port conflicts

com.wowza.wms.transp rt.udp.UDPPortSharing Manager	<u>getUDPPortSharingManager()</u> Get the UDP port sharing manager.
String[]	<u>getUserAgents()</u> Get a pipe " " delimited list of user agents that the server recognizes as RTMPT client.
String	<u>getVersion()</u> Get server version number.
<u>VHostList</u>	<u>getVHostList()</u> Returns the interface to the VHostList for the server
boolean	<u>isAcceptWOWZConnections()</u> If true, WOWZ connections will use the WOWZ protocol.
boolean	<u>isDRMAvailable()</u> Returns true if DRM AddOn is properly licensed.
boolean	<u>isDRMInUse()</u> Returns true if DRM AddOn is in-use.
boolean	<u>isDVRAvailable()</u> Returns true if nDVR AddOn is properly licensed.
boolean	<u>isDVRInUse()</u> Returns true if nDVR AddOn is in-use.
boolean	<u>isDynamicLogContextLoaded()</u> (String logContext) Returns true if the given dynamic log context is already loaded.
boolean	<u>isHTTPOriginAvailable()</u> Returns true if HTTP origin is properly licensed.
boolean	<u>isInitiateWOWZConnections()</u> If true, NetConnection attempts will use the WOWZ protocol when possible.
boolean	<u>isPushPublishAvailable()</u> Returns true if Push Publish is properly licensed.
boolean	<u>isRESTAPIAvailable()</u> Returns true if REST API is properly licensed.
boolean	<u>isRestServerEnabled()</u> Returns true if REST API is enabled
boolean	<u>isSuspended()</u> Is the server current suspended
boolean	<u>isTranscoderWatermark()</u> Return true if transcoder has a watermark
String	<u>readConfig()</u> (String sName) Method to read xml config file..
void	<u>reloadVHostConfig()</u> Reload the VHosts.xml file.
void	<u>removeServerListener()</u> (<u>IServerNotify</u> serverListener) Remove server listener

void	<u>setAcceptWOWZConnections</u> (boolean acceptWOWZConnections) If true, WOWZ connections will use the WOWZ protocol.
void	<u>setCommandInterfaceHostPort</u> (<u>HostPort</u> commandInterfaceHostPort) Set the definition for the command interface.
void	<u>setCoreHandlerPoolSize</u> (int corePoolSize) Set the handler core thread pool size.
void	<u>setCoreTransportPoolSize</u> (int corePoolSize) Set the transport core thread pool size.
void	<u>setDefaultStreamPrefix</u> (String defaultStreamPrefix) Set the default stream prefix used if the stream name does not include a prefix (default is mp4)
void	<u>setDynamicLogProperties</u> (java.util.Properties dynamicLogProperties) Set the dynamic log properties set at the server level
void	<u>setHTTPHeaderServer</u> (String httpHeaderServer) Set the Server header used for HTTP responses
void	<u>setInitiateWOWZConnections</u> (boolean initiateWOWZConnections) If true, NetConnection attempts will use the WOWZ protocol when possible.
void	<u>setRestServerEnabled</u> (boolean restServerEnabled) Sets if REST API server will be enabled
void	<u>setRTMPTHeaderServer</u> (String rtmptHeaderServer) Set the Server header used for RTMPT responses
void	<u>setUserAgents</u> (String[] userAgents) Set a pipe " " delimited list of user agents that the server recognizes as RTMPT client.
void	<u>startCommandInterface</u> () Start the command interface as defined in Server.xml.
void	<u>startVHost</u> (String vhostName) Start a vHost by name.
void	<u>startVHosts</u> () Start all vHosts
void	<u>stopAdminAgent</u> () Stop the JMX interface
void	<u>stopCommandInterface</u> () Stop the command interface as defined in Server.xml.
void	<u>stopVHost</u> (String vhostName) Stop a vHost by name.
void	<u>stopVHosts</u> () Stop all vHosts
void	<u>suspendAllVHosts</u> () Suspend all virtual hosts (Calls IVHost.suspendAllHostPorts for each vhost)
void	<u>suspendServer</u> () Suspend all virtual hosts and the command interface

void	<code>unbindAllVHosts()</code> Unbind all virtual hosts (Calls IVHost.unbindAllHostPorts for each vhost)
boolean	<code>writeConfig(String sName, String data)</code> Method to write xml config file..

Fields

SERVER_DEFAULT

```
public static final java.lang.String SERVER_DEFAULT
```

Constant value: `_defaultServer_`

Methods

startCommandInterface

```
public void startCommandInterface()
```

Start the command interface as defined in Server.xml. The command interface is used by shutdown.sh script to stop the server. It is also used by ant task to stop and start the server on build events.

stopCommandInterface

```
public void stopCommandInterface()
```

Stop the command interface as defined in Server.xml. The command interface is used by shutdown.sh script to stop the server. It is also used by ant task to stop and start the server on build events.

getVersion

```
public String getVersion()
```

Get server version number.

Returns:

server version number

reloadVHostConfig

```
public void reloadVHostConfig()
```

Reload the VHosts.xml file. This method can be invoked through the JMX interface to manage vHosts while the server is running.

stopVHost

```
public void stopVHost(String vhostName)
```

Stop a vHost by name.

Parameters:

vhostName - vHost name

stopVHosts

```
public void stopVHosts()
```

Stop all vHosts

startVHost

```
public void startVHost(String vhostName)
```

Start a vHost by name.

Parameters:

vhostName - vHost name

startVHosts

```
public void startVHosts()
```

Start all vHosts

getIoPerformanceCounter

```
public IoPerformanceCounter getIoPerformanceCounter()
```

Get the server performance counter.

Returns:

io performance counter

getIoPerformanceCounter

```
public IoPerformanceCounter getIoPerformanceCounter(int counterIndex)
```

Get the server performance counter for a specific technology (see IVHost.COUNTER_*)

Parameters:

counterIndex - counter index (see IVHost.COUNTER_*)

Returns:

io performance counter

getConnectionCounter

```
public ConnectionCounter getConnectionCounter()
```

Get the server connection counter.

Returns:

connection counter

getPublishersMaximum

```
public long getPublishersMaximum()
```

Get the maximum number of streams that can be published to the server

(continued from last page)

Returns:

maximum number of streams that can be published to the server

getConnectionCounter

```
public ConnectionCounterSimple getConnectionCounter(int counterIndex)
```

Get the server connection counter for a specific technology (see IVHost.COUNTER_*)

Parameters:

counterIndex - counter index (see IVHost.COUNTER_*)

Returns:

connection counter

getDateStarted

```
public String getDateStarted()
```

Get the date and time the server was started.

Returns:

date and time the server was started

getTimeRunning

```
public String getTimeRunning()
```

Get a formatted String of how long the server has been running.

Returns:

formatted String of how long the server has been running

getTimeRunningSeconds

```
public double getTimeRunningSeconds()
```

Get time running in seconds

Returns:

time running in seconds

getCommandInterfaceHostPort

```
public HostPort getCommandInterfaceHostPort()
```

Get the definition of the command interface

Returns:

host port definition of command interface

setCommandInterfaceHostPort

```
public void setCommandInterfaceHostPort(HostPort commandInterfaceHostPort)
```

Set the definition for the command interface.

Parameters:

(continued from last page)

commandInterfaceHostPort - host port definition of command interface

getUserAgents

```
public String[] getUserAgents()
```

Get a pipe "|" delimited list of user agents that the server recognizes as RTMPT client.

Returns:

pipe "|" delimited list of user agents that the server recognizes as RTMPT client

setUserAgents

```
public void setUserAgents(String[] userAgents)
```

Set a pipe "|" delimited list of user agents that the server recognizes as RTMPT client.

Parameters:

userAgents - pipe "|" delimited list of user agents that the server recognizes as RTMPT client

addServerListener

```
public void addServerListener(IServerNotify serverListener)
```

Add server listener

Parameters:

serverListener - server listener

removeServerListener

```
public void removeServerListener(IServerNotify serverListener)
```

Remove server listener

Parameters:

serverListener - server listener

getAdminInterfaceObjectList

```
public java.util.List getAdminInterfaceObjectList()
```

Get the list of objects exposed through JMX interface

Returns:

list of objects exposed through JMX interface

getCoreTransportPoolSize

```
public int getCoreTransportPoolSize()
```

Get the transport core thread pool size.

Returns:

default core thread pool size

(continued from last page)

setCoreTransportPoolSize

```
public void setCoreTransportPoolSize(int corePoolSize)
```

Set the transport core thread pool size.

Parameters:

corePoolSize - core thread pool size

getCoreHandlerPoolSize

```
public int getCoreHandlerPoolSize()
```

Get the handler core thread pool size.

Returns:

default core thread pool size

setCoreHandlerPoolSize

```
public void setCoreHandlerPoolSize(int corePoolSize)
```

Set the handler core thread pool size.

Parameters:

corePoolSize - core thread pool size

getThreadPool

```
public ThreadPool getThreadPool()
```

Get the server handler thread pool. Same as getHandlerThreadPool.

Returns:

server handler thread pool

getTransportThreadPool

```
public ThreadPool getTransportThreadPool()
```

Get the server transport thread pool. This thread pool is used to read/write data from the transports sockets.

Returns:

server transport thread pool

getHandlerThreadPool

```
public ThreadPool getHandlerThreadPool()
```

Get the server handler thread pool. This thread pool is used to process the incoming events.

Returns:

server handler thread pool

getProperties

```
public WMSProperties getProperties()
```

(continued from last page)

Get server level properties collection

Returns:

server level properties collection

getManagerProperties

```
public WMSProperties getManagerProperties()
```

Get server's manager's properties collection

Returns:

manager's property collection

getClientIdGenerator

```
public RandomIdGenerator getClientIdGenerator()
```

Get the client id generator for the server

Returns:

client id generator for the server

getDynamicLogProperties

```
public java.util.Properties getDynamicLogProperties()
```

Get the dynamic log properties defined at the server level in conf/log4j.properties

Returns:

dynamic log properties defined at the server level

setDynamicLogProperties

```
public void setDynamicLogProperties(java.util.Properties dynamicLogProperties)
```

Set the dynamic log properties set at the server level

Parameters:

dynamicLogProperties - dynamic log properties defined at the server level

isDynamicLogContextLoaded

```
public boolean isDynamicLogContextLoaded(String logContext)
```

Returns true if the given dynamic log context is already loaded. If not loaded it will return false and add it to the a Set of loaded log context. Log context is [VHost].[Application].[AppInstance]

getVHostList

```
public VHostList getVHostList()
```

Returns the interface to the VHostList for the server

Returns:

vhostList

(continued from last page)

suspendAllVHosts

```
public void suspendAllVHosts()
```

Suspend all virtual hosts (Calls IVHost.suspendAllHostPorts for each vhost)

unbindAllVHosts

```
public void unbindAllVHosts()
```

Unbind all virtual hosts (Calls IVHost.unbindAllHostPorts for each vhost)

suspendServer

```
public void suspendServer()
```

Suspend all virtual hosts and the command interface

isSuspended

```
public boolean isSuspended()
```

Is the server current suspended

stopAdminAgent

```
public void stopAdminAgent()
```

Stop the JMX interface

getUDPPortManager

```
public com.wowza.wms.transport.udp.UDPPortManager getUDPPortManager()
```

Get the UDP port manager which manages the allocation of incoming UDP port binding to be sure there are not port conflicts

Returns:

UDP port manager

getUDPPortSharingManager

```
public com.wowza.wms.transport.udp.UDPPortSharingManager getUDPPortSharingManager()
```

Get the UDP port sharing manager.

Returns:

UDP port sharing manager

readConfig

```
public String readConfig(String sName)
```

Method to read xml config file..

writeConfig

```
public boolean writeConfig(String sName,  
                           String data)
```

(continued from last page)

Method to write xml config file..

getCommandInterfaceCommandHandler

```
public CommandInterfaceCommandHandler getCommandInterfaceCommandHandler()
```

Get the command interface command handler

Returns:

command interface command handler

isAcceptWOWZConnections

```
public boolean isAcceptWOWZConnections()
```

If true, WOWZ connections will use the WOWZ protocol. If false, WOWZ connection attempts will fall back to RTMP.

Returns:

true, WOWZ connections will use the WOWZ protocol

setAcceptWOWZConnections

```
public void setAcceptWOWZConnections(boolean acceptWOWZConnections)
```

If true, WOWZ connections will use the WOWZ protocol. If false, WOWZ connection attempts will fall back to RTMP.

Parameters:

acceptWOWZConnections - true, WOWZ connections will use the WOWZ protocol

isInitiateWOWZConnections

```
public boolean isInitiateWOWZConnections()
```

If true, NetConnection attempts will use the WOWZ protocol when possible.

Returns:

true, NetConnection attempts will use the WOWZ protocol when possible.

setInitiateWOWZConnections

```
public void setInitiateWOWZConnections(boolean initiateWOWZConnections)
```

If true, NetConnection attempts will use the WOWZ protocol when possible.

Parameters:

initiateWOWZConnections - true, NetConnection attempts will use the WOWZ protocol when possible.

getLiveStreamTranscoderSessionCount

```
public long getLiveStreamTranscoderSessionCount()
```

Get the current number of live stream transcoder sessions.

Returns:

current number of live stream transcoder sessions

(continued from last page)

getTuningHandler

```
public ServerTuningHandler getTuningHandler()
```

Get the handler that control automatic server tuning

Returns:

handler that control automatic server tuning

getTranscoderLicenseInUse

```
public long getTranscoderLicenseInUse()
```

Get the total number of transcoder licenses in use.

Returns:

total number of transcoder licenses in use.

getTranscoderLicenseTotal

```
public long getTranscoderLicenseTotal()
```

Get the total number of transcoder licenses (-1 means unlimited - subscription)

Returns:

total number of transcoder licenses

isTranscoderWatermark

```
public boolean isTranscoderWatermark()
```

Return true if transcoder has a watermark

Returns:

true if transcoder has a watermark

isDVRAvailable

```
public boolean isDVRAvailable()
```

Returns true if nDVR AddOn is properly licensed.

Returns:

true if nDVR AddOn is properly licensed

isDVRIInUse

```
public boolean isDVRIInUse()
```

Returns true if nDVR AddOn is in-use.

Returns:

true if nDVR AddOn is in-use

isDRMAvailable

```
public boolean isDRMAvailable()
```

(continued from last page)

Returns true if DRM AddOn is properly licensed.

Returns:

true if DRM AddOn is properly licensed

isDRMInUse

```
public boolean isDRMInUse()
```

Returns true if DRM AddOn is in-use.

Returns:

true if DRM AddOn is in-use

isHTTPOriginAvailable

```
public boolean isHTTPOriginAvailable()
```

Returns true if HTTP origin is properly licensed.

Returns:

true if HTTP origin is properly licensed

isRESTAPIAvailable

```
public boolean isRESTAPIAvailable()
```

Returns true if REST API is properly licensed.

Returns:

true if REST API is properly licensed

isPushPublishAvailable

```
public boolean isPushPublishAvailable()
```

Returns true if Push Publish is properly licensed.

Returns:

true if Push Publish is properly licensed.

setRestServerEnabled

```
public void setRestServerEnabled(boolean restServerEnabled)
```

Sets if REST API server will be enabled

Parameters:

restServerEnabled - if the REST API server will be enabled

isRestServerEnabled

```
public boolean isRestServerEnabled()
```

Returns true if REST API is enabled

Returns:

true if REST API is enabled

addAuthorizedRESTAPI

```
public boolean addAuthorizedRESTAPI(Class classType)
```

Returns true if the class has been added to the REST API.

Returns:

true if the class has been added to the REST API

addAuthorizedRESTAPI

```
public boolean addAuthorizedRESTAPI(Class classType,  
String url)
```

Returns true if the class has been added to the REST API at the specified URL

Returns:

true if the class has been added to the REST API at the specified URL

getConnectionsMaximum

```
public long getConnectionsMaximum()
```

Returns maximum number of connections allowed to the server (-1 means unlimited)

Returns:

maximum number of connections allowed to the server (-1 means unlimited)

getDefaultStreamPrefix

```
public String getDefaultStreamPrefix()
```

Get the default stream prefix used if the stream name does not include a prefix (default is mp4)

Returns:

default stream prefix

setDefaultStreamPrefix

```
public void setDefaultStreamPrefix(String defaultStreamPrefix)
```

Set the default stream prefix used if the stream name does not include a prefix (default is mp4)

Parameters:

defaultStreamPrefix - default stream prefix

getRTMPTHeaderServer

```
public String getRTMPTHeaderServer()
```

Get the Server header used for RTMPT responses

Returns:

Server header used for RTMPT responses

(continued from last page)

setRTMPHeaderServer

```
public void setRTMPHeaderServer(String rtmpHeaderServer)
```

Set the Server header used for RTMPT responses

Parameters:

rtmpHeaderServer - Server header used for RTMPT responses

getHTTPHeaderServer

```
public String getHTTPHeaderServer()
```

Get the Server header used for HTTP responses

Returns:

the Server header used for HTTP responses

setHTTPHeaderServer

```
public void setHTTPHeaderServer(String httpHeaderServer)
```

Set the Server header used for HTTP responses

Parameters:

httpHeaderServer

com.wowza.wms.server Interface IServerNotify

All Subinterfaces:

[IServerNotify2](#), [IServerNotify3](#)

public interface **IServerNotify**
extends

IServerNotify: listener interface. Configured by adding class entries definitions to Server.xml.Startup order is: [constructor]; onServerConfigLoaded, onServerCreate, onServerInit Shutdown order is: onServerShutdownStart, onServerShutdownComplete, [exit]

Method Summary

void	onServerCreate (IServer server) Triggered when server object is first created.
void	onServerInit (IServer server) Triggered when server initialization is complete and all VHosts have been started
void	onServerShutdownComplete (IServer server) Triggered at the end of server shutdown
void	onServerShutdownStart (IServer server) Triggered at the beginning of server shutdown

Methods

onServerCreate

public void **onServerCreate**([IServer](#) server)

Triggered when server object is first created.

Parameters:

server - server object

onServerInit

public void **onServerInit**([IServer](#) server)

Triggered when server initialization is complete and all VHosts have been started

Parameters:

server - server object

onServerShutdownStart

public void **onServerShutdownStart**([IServer](#) server)

Triggered at the beginning of server shutdown

(continued from last page)

Parameters:

server - server object

onServerShutdownComplete

```
public void onServerShutdownComplete(IServer server)
```

Triggered at the end of server shutdown

Parameters:

server - server object

com.wowza.wms.server Interface IServerNotify2

All Superinterfaces:

[IServerNotify](#)

All Subinterfaces:

[IServerNotify3](#)

public interface **IServerNotify2**
extends [IServerNotify](#)

IServerNotify2: listener interface. Configured by adding class entries definitions to Server.xml.Startup order is: [constructor]; onServerConfigLoaded, onServerCreate, onServerInit Shutdown order is: onServerShutdownStart, onServerShutdownComplete, [exit]

Method Summary

void	onServerConfigLoaded (IServer server) Triggered when server configuration is loaded
------	---

Methods inherited from interface [com.wowza.wms.server.IServerNotify](#)

[onServerCreate](#), [onServerInit](#), [onServerShutdownComplete](#), [onServerShutdownStart](#)

Methods

onServerConfigLoaded

public void **onServerConfigLoaded**([IServer](#) server)

Triggered when server configuration is loaded

Parameters:

server - server object

com.wowza.wms.server Interface IServerNotify3

All Superinterfaces:

[IServerNotify2](#), [IServerNotify](#)

public interface **IServerNotify3**
extends [IServerNotify2](#)

Method Summary

void	onServerAfterVHostStartup (IServer server) Triggered just after virtual hosts are started.
void	onServerBeforeVHostStartup (IServer server) Triggered just before virtual hosts are started.

Methods inherited from interface [com.wowza.wms.server.IServerNotify2](#)

[onServerConfigLoaded](#)

Methods inherited from interface [com.wowza.wms.server.IServerNotify](#)

[onServerCreate](#), [onServerInit](#), [onServerShutdownComplete](#), [onServerShutdownStart](#)

Methods

onServerBeforeVHostStartup

public void **onServerBeforeVHostStartup**([IServer](#) server)

Triggered just before virtual hosts are started.

Parameters:

server - server object

onServerAfterVHostStartup

public void **onServerAfterVHostStartup**([IServer](#) server)

Triggered just after virtual hosts are started.

Parameters:

server - server object

com.wowza.wms.server Class Server

java.lang.Object

└-com.wowza.wms.server.Server

All Implemented Interfaces:

[IServer](#)

public class **Server**
extends Object
implements [IServer](#)

Nested Class Summary

class	Server.TranscoderPollingTracker Server.TranscoderPollingTracker
-------	--

Field Summary

public	debugWOWZProtocol
public static	logNotifier

Fields inherited from interface [com.wowza.wms.server.IServer](#)

[SERVER_DEFAULT](#)

Constructor Summary

public	Server ()
--------	---------------------------

Method Summary

boolean	addAuthorizedRESTAPI (Class classType)
boolean	addAuthorizedRESTAPI (Class classType, String url)
void	addServerListener (IServerNotify serverListener)
static void	decodeS (String[] ins) Deprecated.
static boolean	decodeSS (String in) Deprecated.
String	decodeStorageDir (IVHost vhost, String storageDir)

void	doWatchdog()
static String	fS (byte[] kiIn, String p) Deprecated.
com.wowza.wms.admin.AdminAgent	getAdminAgent()
String	getAdminGUID()
java.util.List	getAdminInterfaceObjectList()
RandomIdGenerator	getClientIdGenerator()
int	getClientIdGeneratorRecycleDelaySize()
int	getClientIdGeneratorRecycleSize()
long	getClientIdGeneratorTimeout()
Object	getCommandInterface()
CommandInterfaceCommandHandler	getCommandInterfaceCommandHandler()
HostPort	getCommandInterfaceHostPort()
long	getCommittedVirtualMemory()
ConnectionCounter	getConnectionCounter()
ConnectionCounterSimple	getConnectionCounter (int counterIndex)
long	getConnectionsMaximum()
IConnectionValidator	getConnectionValidator()
int	getCoreHandlerPoolSize()
int	getCoreTransportPoolSize()
int	getCryptoPoolActiveCount()
int	getCryptoPoolMaxSize()
long	getCurrentHeapSize()
String	getDateStarted()
String	getDefaultStreamPrefix()

java.util.Properties	getDynamicLogProperties()
String	getGUID()
ThreadPool	getHandlerThreadPool()
String	getHTTPHeaderServer()
static Server	getInstance()
IOPerformanceCounter	getIoPerformanceCounter()
IOPerformanceCounter	getIoPerformanceCounter(int counterIndex)
JMXRemoteConfig	getJmxRemoteConfig()
LicenseCounter	getLicenseCounter(int index)
Server.LicenseSessionTracker	getLicenseTracker(int index)
ILicenseValidator	getLicenseValidator()
long	getLiveStreamTranscoderSessionCount()
long	getLiveThreads()
WMSProperties	getManagerProperties()
long	getMaxHeapSize()
long	getPeakThreads()
WMSProperties	getProperties()
boolean[]	getProtocolUsage()
Server.ProtocolUsageSessionTracker	getProtocolUsageTracker()
long	getPublishersMaximum()
String	getRTMPHeaderServer()
String	getServerGUID()
String	getSessionGUID()
ThreadPool	getThreadPool()

String	getTimeRunning()
double	getTimeRunningSeconds()
long	getTranscoderLicenseInUse()
long	getTranscoderLicenseTotal()
ThreadPool	getTransportThreadPool()
ServerTuningHandler	getTuningHandler()
com.wowza.wms.transpo rt.udp.UDPPortManager	getUDPPortManager()
com.wowza.wms.transpo rt.udp.UDPPortSharing Manager	getUDPPortSharingManager()
String[]	getUserAgents()
String	getVersion()
VHostList	getVHostList()
boolean	isAcceptWOWZConnections()
boolean	isDRMAvailable()
boolean	isDRMInUse()
boolean	isDVRAvailable()
boolean	isDVRInUse()
boolean	isDynamicLogContextLoaded() (String logContext)
boolean	isHTTPOriginAvailable()
boolean	isInitiateWOWZConnections()
boolean	isPushPublishAvailable()
boolean	isRESTAPIAvailable()
boolean	isRestDocumentationServerEnabled()
boolean	isRestServerEnabled()
boolean	isSuspended()

boolean	<u>isTranscoderWatermark()</u>
boolean	<u>isVHostRunning</u> (String name)
static void	<u>main</u> (String[] args)
void	<u>onNewVHost</u> (<u>IVHost</u> vhost)
String	<u>readConfig</u> (String sName)
static String	<u>readXMLConfig</u> (String sPath)
void	<u>registerLiveStreamTranscoder</u> (<u>ILiveStreamTranscoder</u> liveStreamTranscoder, byte[] license)
void	<u>reloadVHostConfig</u> ()
void	<u>removeServerListener</u> (<u>IServerNotify</u> serverListener)
void	<u>setAcceptWOWZConnections</u> (boolean acceptWOWZConnections)
void	<u>setCommandInterface</u> (Object commandInterface)
void	<u>setCommandInterfaceHostPort</u> (<u>HostPort</u> commandInterfaceHostPort)
void	<u>setCoreHandlerPoolSize</u> (int corePoolSize)
void	<u>setCoreTransportPoolSize</u> (int corePoolSize)
void	<u>setDefaultStreamPrefix</u> (String defaultStreamPrefix)
void	<u>setDynamicLogProperties</u> (java.util.Properties dynamicLogProperties)
void	<u>setHTTPHeaderServer</u> (String httpHeaderServer)
void	<u>setInitiateWOWZConnections</u> (boolean initiateWOWZConnections)
void	<u>setIoPerformanceCounter</u> (<u>IOPerformanceCounter</u> ioPerformanceCounter)
void	<u>setRestDocumentationServerEnabled</u> (boolean restDocumentationServerEnabled)
void	<u>setRestServerEnabled</u> (boolean restServerEnabled)
void	<u>setRTMPHeaderServer</u> (String rtmtptHeaderServer)
void	<u>setUserAgents</u> (String[] userAgents)

static void	start()
void	startCommandInterface()
void	startServer()
void	startStats()
void	startVHost(String vhostName)
void	startVHosts()
void	stopAdminAgent()
void	stopCommandInterface()
void	stopServer()
void	stopStats()
void	stopVHost(String vhostName)
void	stopVHosts()
void	suspendAllVHosts()
void	suspendCommandInterface()
void	suspendServer()
void	unbindAllVHosts()
void	unregisterLiveStreamTranscoder(ILiveStreamTranscoder liveStreamTranscoder, byte[] license)
IOPerformanceCounter	updateIOPerformance()
void	updateLoggingDuration()
boolean	writeConfig(String sName, String data)
static boolean	writeXMLConfig(String sPath, String data)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface [com.wowza.wms.server.IServer](#)

```
addAuthorizedRESTAPI, addAuthorizedRESTAPI, addServerListener,  
getAdminInterfaceObjectList, getClientIdGenerator, getCommandInterfaceCommandHandler,  
getCommandInterfaceHostPort, getConnectionCounter, getConnectionCounter,  
getConnectionsMaximum, getCoreHandlerPoolSize, getCoreTransportPoolSize,  
getDateStarted, getDefaultStreamPrefix, getDynamicLogProperties,  
getHandlerThreadPool, getHTTPHeaderServer, getIoPerformanceCounter,  
getIoPerformanceCounter, getLiveStreamTranscoderSessionCount, getManagerProperties,  
getProperties, getPublishersMaximum, getRTMPHeaderServer, getThreadPool,  
getTimeRunning, getTimeRunningSeconds, getTranscoderLicenseInUse,  
getTranscoderLicenseTotal, getTransportThreadPool, getTuningHandler,  
getUDPPortManager, getUDPPortSharingManager, getUserAgents, getVersion, getVHostList,  
isAcceptWOWZConnections, isDRMAvailable, isDRMInUse, isDVRAvailable, isDVRInUse,  
isDynamicLogContextLoaded, isHTTPOriginAvailable, isInitiateWOWZConnections,  
isPushPublishAvailable, isRESTAPIAvailable, isRestServerEnabled, isSuspended,  
isTranscoderWatermark, readConfig, reloadVHostConfig, removeServerListener,  
setAcceptWOWZConnections, setCommandInterfaceHostPort, setCoreHandlerPoolSize,  
setCoreTransportPoolSize, setDefaultStreamPrefix, setDynamicLogProperties,  
setHTTPHeaderServer, setInitiateWOWZConnections, setRestServerEnabled,  
setRTMPHeaderServer, setUserAgents, startCommandInterface, startVHost, startVHosts,  
stopAdminAgent, stopCommandInterface, stopVHost, stopVHosts, suspendAllVHosts,  
suspendServer, unbindAllVHosts, writeConfig
```

Fields

logNotifier

```
public static com.wowza.wms.logging.ILogNotify logNotifier
```

debugWOWZProtocol

```
public boolean debugWOWZProtocol
```

Constructors

Server

```
public Server()
```

Methods

getInstance

```
public static Server getInstance()
```

(continued from last page)

getAdminAgent

```
public com.wowza.wms.admin.AdminAgent getAdminAgent()
```

main

```
public static void main(String[] args)
```

start

```
public static void start()
```

stopServer

```
public void stopServer()
```

stopAdminAgent

```
public void stopAdminAgent()
```

isVHostRunning

```
public boolean isVHostRunning(String name)
```

suspendCommandInterface

```
public void suspendCommandInterface()
```

startCommandInterface

```
public void startCommandInterface()
```

stopCommandInterface

```
public void stopCommandInterface()
```

startServer

```
public void startServer()
```

(continued from last page)

getVersion

```
public String getVersion()
```

reloadVHostConfig

```
public void reloadVHostConfig()
```

stopVHost

```
public void stopVHost(String vhostName)
```

stopVHosts

```
public void stopVHosts()
```

decodeStorageDir

```
public String decodeStorageDir(IVHost vhost,  
                                String storageDir)
```

startVHost

```
public void startVHost(String vhostName)
```

isSuspended

```
public boolean isSuspended()
```

suspendServer

```
public void suspendServer()
```

suspendAllVHosts

```
public void suspendAllVHosts()
```

(continued from last page)

unbindAllVHosts

```
public void unbindAllVHosts()
```

startVHosts

```
public void startVHosts()
```

doWatchdog

```
public void doWatchdog()
```

updateIOPerformance

```
public IOPerformanceCounter updateIOPerformance()
```

getIoPerformanceCounter

```
public IOPerformanceCounter getIoPerformanceCounter()
```

getIoPerformanceCounter

```
public IOPerformanceCounter getIoPerformanceCounter(int counterIndex)
```

setIoPerformanceCounter

```
public void setIoPerformanceCounter(IOPerformanceCounter ioPerformanceCounter)
```

getConnectionCounter

```
public ConnectionCounter getConnectionCounter()
```

getConnectionCounter

```
public ConnectionCounterSimple getConnectionCounter(int counterIndex)
```

getConnectionValidator

```
public IConnectionValidator getConnectionValidator()
```

(continued from last page)

getLicenseValidator

```
public ILicenseValidator getLicenseValidator()
```

onNewVHost

```
public void onNewVHost(IVHost vhost)
```

getDateStarted

```
public String getDateStarted()
```

getTimeRunning

```
public String getTimeRunning()
```

getTimeRunningSeconds

```
public double getTimeRunningSeconds()
```

getCommandInterfaceHostPort

```
public HostPort getCommandInterfaceHostPort()
```

setCommandInterfaceHostPort

```
public void setCommandInterfaceHostPort(HostPort commandInterfaceHostPort)
```

getCommandInterface

```
public Object getCommandInterface()
```

setCommandInterface

```
public void setCommandInterface(Object commandInterface)
```

(continued from last page)

getUserAgents

```
public String[] getUserAgents()
```

setUserAgents

```
public void setUserAgents(String[] userAgents)
```

updateLoggingDuration

```
public void updateLoggingDuration()
```

addServerListener

```
public void addServerListener(IServerNotify serverListener)
```

removeServerListener

```
public void removeServerListener(IServerNotify serverListener)
```

getJmxRemoteConfig

```
public JMXRemoteConfig getJmxRemoteConfig()
```

getAdminInterfaceObjectList

```
public java.util.List getAdminInterfaceObjectList()
```

getCoreTransportPoolSize

```
public int getCoreTransportPoolSize()
```

setCoreTransportPoolSize

```
public void setCoreTransportPoolSize(int corePoolSize)
```

getCoreHandlerPoolSize

```
public int getCoreHandlerPoolSize()
```

(continued from last page)

setCoreHandlerPoolSize

```
public void setCoreHandlerPoolSize(int corePoolSize)
```

getThreadPool

```
public ThreadPool getThreadPool()
```

getTransportThreadPool

```
public ThreadPool getTransportThreadPool()
```

getHandlerThreadPool

```
public ThreadPool getHandlerThreadPool()
```

getProperties

```
public WMSProperties getProperties()
```

getManagerProperties

```
public WMSProperties getManagerProperties()
```

getClientIdGenerator

```
public RandomIdGenerator getClientIdGenerator()
```

getDynamicLogProperties

```
public java.util.Properties getDynamicLogProperties()
```

setDynamicLogProperties

```
public void setDynamicLogProperties(java.util.Properties dynamicLogProperties)
```

(continued from last page)

isDynamicLogContextLoaded

```
public boolean isDynamicLogContextLoaded(String logContext)
```

getVHostList

```
public VHostList getVHostList()
```

getSessionGUID

```
public String getSessionGUID()
```

getServerGUID

```
public String getServerGUID()
```

getGUID

```
public String getGUID()
```

getAdminGUID

```
public String getAdminGUID()
```

getUDPPortSharingManager

```
public com.wowza.wms.transport.udp.UDPPortSharingManager getUDPPortSharingManager()
```

getUDPPortManager

```
public com.wowza.wms.transport.udp.UDPPortManager getUDPPortManager()
```

getCryptoPoolMaxSize

```
public int getCryptoPoolMaxSize()
```

getCryptoPoolActiveCount

```
public int getCryptoPoolActiveCount()
```

(continued from last page)

getLiveThreads

```
public long getLiveThreads()
```

getPeakThreads

```
public long getPeakThreads()
```

getCurrentHeapSize

```
public long getCurrentHeapSize()
```

getMaxHeapSize

```
public long getMaxHeapSize()
```

getCommittedVirtualMemory

```
public long getCommittedVirtualMemory()
```

readConfig

```
public String readConfig(String sName)
```

writeConfig

```
public boolean writeConfig(String sName,  
    String data)
```

readXMLConfig

```
public static String readXMLConfig(String sPath)
```

writeXMLConfig

```
public static boolean writeXMLConfig(String sPath,  
    String data)
```

getClientIdGeneratorTimeout

```
public long getClientIdGeneratorTimeout()
```

getClientIdGeneratorRecycleSize

```
public int getClientIdGeneratorRecycleSize()
```

getClientIdGeneratorRecycleDelaySize

```
public int getClientIdGeneratorRecycleDelaySize()
```

getCommandInterfaceCommandHandler

```
public CommandInterfaceCommandHandler getCommandInterfaceCommandHandler()
```

getLicenseCounter

```
public LicenseCounter getLicenseCounter(int index)
```

getLicenseTracker

```
public Server.LicenseSessionTracker getLicenseTracker(int index)
```

decodeS

```
public static void decodeS(String[] ins)
```

Deprecated.

decodeSS

```
public static boolean decodeSS(String in)
```

Deprecated.

fS

```
public static String fS(byte[] kiIn,  
                        String p)
```

Deprecated.

(continued from last page)

registerLiveStreamTranscoder

```
public void registerLiveStreamTranscoder(ILiveStreamTranscoder liveStreamTranscoder,  
byte[] license)
```

unregisterLiveStreamTranscoder

```
public void unregisterLiveStreamTranscoder(ILiveStreamTranscoder liveStreamTranscoder,  
byte[] license)
```

getProtocolUsage

```
public boolean[] getProtocolUsage()
```

getProtocolUsageTracker

```
public Server.ProtocolUsageSessionTracker getProtocolUsageTracker()
```

isAcceptWOWZConnections

```
public boolean isAcceptWOWZConnections()
```

setAcceptWOWZConnections

```
public void setAcceptWOWZConnections(boolean acceptWOWZConnections)
```

isInitiateWOWZConnections

```
public boolean isInitiateWOWZConnections()
```

setInitiateWOWZConnections

```
public void setInitiateWOWZConnections(boolean initiateWOWZConnections)
```

setRestServerEnabled

```
public void setRestServerEnabled(boolean restServerEnabled)
```

isRestServerEnabled

```
public boolean isRestServerEnabled()
```

setRestDocumentationServerEnabled

```
public void setRestDocumentationServerEnabled(boolean restDocumentationServerEnabled)
```

isRestDocumentationServerEnabled

```
public boolean isRestDocumentationServerEnabled()
```

addAuthorizedRESTAPI

```
public boolean addAuthorizedRESTAPI(Class classType)
```

addAuthorizedRESTAPI

```
public boolean addAuthorizedRESTAPI(Class classType,  
    String url)
```

getLiveStreamTranscoderSessionCount

```
public long getLiveStreamTranscoderSessionCount()
```

getTuningHandler

```
public ServerTuningHandler getTuningHandler()
```

getTranscoderLicenseInUse

```
public long getTranscoderLicenseInUse()
```

getTranscoderLicenseTotal

```
public long getTranscoderLicenseTotal()
```

isTranscoderWatermark

```
public boolean isTranscoderWatermark()
```

(continued from last page)

isDVRAvailable

```
public boolean isDVRAvailable()
```

isDVRInUse

```
public boolean isDVRInUse()
```

isDRMAvailable

```
public boolean isDRMAvailable()
```

isHTTPOriginAvailable

```
public boolean isHTTPOriginAvailable()
```

isPushPublishAvailable

```
public boolean isPushPublishAvailable()
```

isRESTAPIAvailable

```
public boolean isRESTAPIAvailable()
```

isDRMInUse

```
public boolean isDRMInUse()
```

getConnectionsMaximum

```
public long getConnectionsMaximum()
```

getPublishersMaximum

```
public long getPublishersMaximum()
```

(continued from last page)

startStats

```
public void startStats()
```

stopStats

```
public void stopStats()
```

getDefaultStreamPrefix

```
public String getDefaultStreamPrefix()
```

setDefaultStreamPrefix

```
public void setDefaultStreamPrefix(String defaultStreamPrefix)
```

getRTMPHeaderServer

```
public String getRTMPHeaderServer()
```

setRTMPHeaderServer

```
public void setRTMPHeaderServer(String rtmpHeaderServer)
```

getHTTPHeaderServer

```
public String getHTTPHeaderServer()
```

setHTTPHeaderServer

```
public void setHTTPHeaderServer(String httpHeaderServer)
```

com.wowza.wms.server

Class Server.TranscoderPollingTracker

java.lang.Object

└-com.wowza.wms.server.Server.TranscoderPollingTracker

public class **Server.TranscoderPollingTracker**

extends Object

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
--

Package

com.wowza.wms.sharedobject

com.wowza.wms.sharedobject Interface ISharedObject

All Known Implementing Classes:

[SharedObject](#)

public interface **ISharedObject**
extends

ISharedObject: public interface to SharedObject class.

Field Summary

public static final	FILEEXTENSION Value: rso
public static final	SHARED_OBJECT_CMD_CONNECT shared object command: connect Value: 1
public static final	SHARED_OBJECT_CMD_CONNECTSUCCESS shared object command: clear Value: 11
public static final	SHARED_OBJECT_CMD_DELETE shared object command: delete Value: 10
public static final	SHARED_OBJECT_CMD_DISCONNECT shared object command: disconnect Value: 2
public static final	SHARED_OBJECT_CMD_ERROR shared object command: error Value: 7
public static final	SHARED_OBJECT_CMD_SEND shared object command: send Value: 6
public static final	SHARED_OBJECT_CMD_SETVALUE shared object command: setvalue Value: 3
public static final	SHARED_OBJECT_STATUS_CHANGE shared object status: change Value: 4
public static final	SHARED_OBJECT_STATUS_CLEAR shared object status: clear Value: 8

public static final	<u>SHARED_OBJECT_STATUS_DELETE</u> shared object status: delete Value: 9
public static final	<u>SHARED_OBJECT_STATUS_SUCCESS</u> shared object status: success Value: 5

Method Summary

void	<u>acquire()</u> Increment the reference count to this shared object.
void	<u>addClient(IClient client)</u> Add a client to this shared object.
void	<u>addSlotListener(ISharedObjectSlotNotify slotListener)</u> Add a slot listener.
void	<u>clear()</u> Clear all properties from a shared object
void	<u>close()</u> Force close this shared object (not implemented)
boolean	<u>containsProperty(String slotName)</u> Returns true is slot/property name exists
boolean	<u>containsSlot(String slotName)</u> Returns true is slot/property name exists
void	<u>deleteSlot(IClient client, String slotName)</u> Remove a slot (property)
void	<u>deleteSlot(String slotName)</u> Remove slot (property)
void	<u>disconnect(IClient client)</u> Disconnect client from shared object.
void	<u>flush()</u> Flush (write to disk) shared object
java.util.List	<u>getClients()</u> Get a list of client that are connected to this shared object.
String	<u>getName()</u> Get shared object name
<u>ISharedObjects</u>	<u>getParent()</u> Get the shared object container to which this shared object belongs.
<u>AMFData</u>	<u>getProperty(String slotName)</u> Get slot (property) value.
int	<u>getRefCount()</u> Get the current reference (clients) connected to this shared object.

<u>ISharedObjectSlot</u>	<u>getSlot</u> (String name) Get ISharedObjectSlot interface to a slot (property) by name
java.util.List	<u>getSlotNames</u> () Get a list of slot (property) names
java.util.List	<u>getSlots</u> () Get a list of active slots
String	<u>getStorageDir</u> () Get path used to store shared object.
int	<u>getVersion</u> () Get the interval version number.
boolean	<u>isClient</u> (<u>IClient</u> client) Is this client connected to shared object
boolean	<u>isPersistent</u> () Is this shared object being persisted.
void	<u>lock</u> () Lock a shared object for write access
int	<u>purge</u> (int version) Purge all deleted properties older than the version number
void	<u>putSlot</u> (String name, <u>ISharedObjectSlot</u> slot) Add a new slot (property) to a shared object.
void	<u>release</u> () Decrement the reference count to this shared object.
void	<u>removeClient</u> (<u>IClient</u> client) Remove a client from this shared object.
void	<u>removeSlotListener</u> (<u>ISharedObjectSlotNotify</u> slotListener) Remove slot listener
void	<u>send</u> (String handlerName) Call client side handler attached to shared object (no parameters).
void	<u>send</u> (String handlerName, Object[] params) Call client side handler attached to shared object.
void	<u>setName</u> (String name) Set shared object name
void	<u>setPersistent</u> (boolean isPersistent) Set is shared object persisted.
void	<u>setProperty</u> (String slotName, <u>AMFData</u> data) Set slot (property) value as AMFData object.
void	<u>setProperty</u> (String slotName, boolean value) Set slot (property) value as a boolean value (will be wrapped in an AMFDataItem object)
void	<u>setProperty</u> (String slotName, java.util.Date value) Set slot (property) value as a date value (will be wrapped in an AMFDataItem object)

void	<code>setProperty</code> (String slotName, double value) Set slot (property) value as a double value (will be wrapped in an AMFDataItem object)
void	<code>setProperty</code> (String slotName, int value) Set slot (property) value as a int value (will be wrapped in an AMFDataItem object)
void	<code>setProperty</code> (String slotName, long value) Set slot (property) value as a long value (will be wrapped in an AMFDataItem object)
void	<code>setProperty</code> (String slotName, String value) Set slot (property) value as a string value (will be wrapped in an AMFDataItem object)
void	<code>setStorageDir</code> (String storageDir) Set path used to store shared object.
void	<code>setVersion</code> (int version) Set the internal version number.
int	<code>size</code> () Get the number of active slot (properties).
void	<code>unlock</code> () Unlock a shared object for write access
void	<code>writeDeleteError</code> (IClient client, String soName, boolean isPersistent, String slotName, String errorMsg) Write an delete error message back to the client
void	<code>writeSetValueError</code> (IClient client, String soName, boolean isPersistent, String slotName, String errorMsg) Write an set value error message back to the client

Fields

FILEEXTENSION

```
public static final java.lang.String FILEEXTENSION
```

Constant value: **rs0**

SHARED_OBJECT_CMD_CONNECT

```
public static final byte SHARED_OBJECT_CMD_CONNECT
```

shared object command: connect

Constant value: **1**

SHARED_OBJECT_CMD_DISCONNECT

```
public static final byte SHARED_OBJECT_CMD_DISCONNECT
```

shared object command: disconnect

Constant value: **2**

(continued from last page)

SHARED_OBJECT_CMD_SETVALUE

```
public static final byte SHARED_OBJECT_CMD_SETVALUE
```

shared object command: setvalue
Constant value: **3**

SHARED_OBJECT_CMD_SEND

```
public static final byte SHARED_OBJECT_CMD_SEND
```

shared object command: send
Constant value: **6**

SHARED_OBJECT_CMD_ERROR

```
public static final byte SHARED_OBJECT_CMD_ERROR
```

shared object command: error
Constant value: **7**

SHARED_OBJECT_CMD_DELETE

```
public static final byte SHARED_OBJECT_CMD_DELETE
```

shared object command: delete
Constant value: **10**

SHARED_OBJECT_CMD_CONNECTSUCCESS

```
public static final byte SHARED_OBJECT_CMD_CONNECTSUCCESS
```

shared object command: clear
Constant value: **11**

SHARED_OBJECT_STATUS_CHANGE

```
public static final byte SHARED_OBJECT_STATUS_CHANGE
```

shared object status: change
Constant value: **4**

SHARED_OBJECT_STATUS_SUCCESS

```
public static final byte SHARED_OBJECT_STATUS_SUCCESS
```

shared object status: success
Constant value: **5**

SHARED_OBJECT_STATUS_CLEAR

```
public static final byte SHARED_OBJECT_STATUS_CLEAR
```

shared object status: clear
Constant value: **8**

SHARED_OBJECT_STATUS_DELETE

```
public static final byte SHARED_OBJECT_STATUS_DELETE
```

(continued from last page)

shared object status: delete
Constant value: **9**

Methods

getSlotNames

```
public java.util.List getSlotNames()
```

Get a list of slot (property) names

Returns:

list of slot (property) names

getSlot

```
public ISharedObjectSlot getSlot(String name)
```

Get ISharedObjectSlot interface to a slot (property) by name

Parameters:

name - slot (property) name

Returns:

ISharedObjectSlot interface

putSlot

```
public void putSlot(String name,  
    ISharedObjectSlot slot)
```

Add a new slot (property) to a shared object.

Parameters:

name - slot (property) name

slot - new slot

getClients

```
public java.util.List getClients()
```

Get a list of client that are connected to this shared object.

Returns:

list of client that are connected to this shared object

isClient

```
public boolean isClient(IClient client)
```

Is this client connected to shared object

Parameters:

client - client

Returns:

true if client is conencted to shared object

addClient

```
public void addClient(IClient client)
```

Add a client to this shared object.

Parameters:

client - client

removeClient

```
public void removeClient(IClient client)
```

Remove a client from this shared object.

Parameters:

client - client

size

```
public int size()
```

Get the number of active slot (properties). Deleted slots are not counted.

Returns:

number of active slot (properties). Deleted slots are not counted

acquire

```
public void acquire()
```

Increment the reference count to this shared object.

release

```
public void release()
```

Descrement the reference count to this shared object.

close

```
public void close()
```

Force close this shared object (not implemented)

getRefCount

```
public int getRefCount()
```

Get the current reference (clients) connected to this shared object. Includes unbalanced count of calls to acquire.

Returns:

the current reference (clients) connected to this shared object

isPersistent

```
public boolean isPersistent()
```

(continued from last page)

Is this shared object being persisted.

Returns:

true if shared object is being persisted

setPersistent

```
public void setPersistent(boolean isPersistent)
```

Set is shared object persisted.

Parameters:

isPersistent - true if shared object is being persisted

getVersion

```
public int getVersion()
```

Get the interval version number.

Returns:

interval version number

setVersion

```
public void setVersion(int version)
```

Set the internal version number.

Parameters:

version - internal version number

flush

```
public void flush()
```

Flush (write to disk) shared object

deleteSlot

```
public void deleteSlot(IClient client,  
String slotName)
```

Remove a slot (property)

Parameters:

client - client removing slot or null if server side code
slotName - slot (property) name

deleteSlot

```
public void deleteSlot(String slotName)
```

Remove slot (property)

Parameters:

slotName - slot (property) name

getProperty

```
public AMFData getProperty(String slotName)
```

Get slot (property) value.

Parameters:

slotName - slot (property) name

Returns:

slot (property) value as AMFData object

setProperty

```
public void setProperty(String slotName,  
                        String value)
```

Set slot (property) value as a string value (will be wrapped in an AMFDataItem object)

Parameters:

slotName - slot (property) name

value - string value

setProperty

```
public void setProperty(String slotName,  
                        double value)
```

Set slot (property) value as a double value (will be wrapped in an AMFDataItem object)

Parameters:

slotName - slot (property) name

value - double value

setProperty

```
public void setProperty(String slotName,  
                        int value)
```

Set slot (property) value as a int value (will be wrapped in an AMFDataItem object)

Parameters:

slotName - slot (property) name

value - int value

setProperty

```
public void setProperty(String slotName,  
                        long value)
```

Set slot (property) value as a long value (will be wrapped in an AMFDataItem object)

Parameters:

slotName - slot (property) name

value - long value

(continued from last page)

setProperty

```
public void setProperty(String slotName,  
    java.util.Date value)
```

Set slot (property) value as a date value (will be wrapped in an AMFDataItem object)

Parameters:

slotName - slot (property) name
value - date value

setProperty

```
public void setProperty(String slotName,  
    boolean value)
```

Set slot (property) value as a boolean value (will be wrapped in an AMFDataItem object)

Parameters:

slotName - slot (property) name
value - boolean value

setProperty

```
public void setProperty(String slotName,  
    AMFData data)
```

Set slot (property) value as AMFData object.

Parameters:

slotName - slot (property) name
data - slot value as AMFData object. Example: new AMFDataItem((double)1.234) or new AMFDataItem()

disconnect

```
public void disconnect(IClient client)
```

Disconnect client from shared object.

Parameters:

client - client to disconnect

getName

```
public String getName()
```

Get shared object name

Returns:

shared object name

setName

```
public void setName(String name)
```

Set shared object name

Parameters:

name - shared object name

send

```
public void send(String handlerName,  
                Object[] params)
```

Call client side handler attached to shared object.

Parameters:

handlerName - handler name

params - variable argument list of parameters

send

```
public void send(String handlerName)
```

Call client side handler attached to shared object (no parameters).

Parameters:

handlerName - handler name

addSlotListener

```
public void addSlotListener(ISharedObjectSlotNotify slotListener)
```

Add a slot listener. Will receive the following events: onSlotSetValue and onSlotDelete

Parameters:

slotListener - slot listener

removeSlotListener

```
public void removeSlotListener(ISharedObjectSlotNotify slotListener)
```

Remove slot listener

Parameters:

slotListener - slot listener

getStorageDir

```
public String getStorageDir()
```

Get path used to store shared object.

Returns:

path used to store shared object

setStorageDir

```
public void setStorageDir(String storageDir)
```

Set path used to store shared object.

Parameters:

storageDir - path used to store shared object

(continued from last page)

writeSetValueError

```
public void writeSetValueError(IClient client,  
    String soName,  
    boolean isPersistent,  
    String slotName,  
    String errorMsg)
```

Write an set value error message back to the client

Parameters:

client - client removing slot or null if server side code
soName - sharedObject name
isPersistent - is persistent
slotName - slot name
errorMsg - error message

writeDeleteError

```
public void writeDeleteError(IClient client,  
    String soName,  
    boolean isPersistent,  
    String slotName,  
    String errorMsg)
```

Write an delete error message back to the client

Parameters:

client - client removing slot or null if server side code
soName - sharedObject name
isPersistent - is persistent
slotName - slot name
errorMsg - error message

clear

```
public void clear()
```

Clear all properties from a shared object

purge

```
public int purge(int version)
```

Purge all deleted properties older than the version number

Returns:

number of slots purged

getSlots

```
public java.util.List getSlots()
```

Get a list of active slots

Returns:

list of active slots

(continued from last page)

containsProperty

```
public boolean containsProperty(String slotName)
```

Returns true is slot/property name exists

Parameters:

slotName - slot/property name

Returns:

true is slot/property name exists

containsSlot

```
public boolean containsSlot(String slotName)
```

Returns true is slot/property name exists

Parameters:

slotName - slot/property name

Returns:

true is slot/property name exists

lock

```
public void lock()
```

Lock a shared object for write access

unlock

```
public void unlock()
```

Unlock a shared object for write access

getParent

```
public ISharedObjects getParent()
```

Get the shared object container to which this shared object belongs.

Returns:

shared object container

com.wowza.wms.sharedobject Interface ISharedObjectNotify

public interface **ISharedObjectNotify**
extends

ISharedObjectNotify: listener interface used by IApplicationInstance addSharedObjectListener

Method Summary

void	onSharedObjectConnect (ISharedObject sharedObject, IClient client) Triggered when client connects to sharedObject
void	onSharedObjectCreate (ISharedObject sharedObject) Triggered when sharedObject created
void	onSharedObjectDestroy (ISharedObject sharedObject) Triggered when sharedObject destroyed
void	onSharedObjectDisconnect (ISharedObject sharedObject, IClient client) Triggered when client disconnects from sharedObject

Methods

onSharedObjectCreate

public void **onSharedObjectCreate**([ISharedObject](#) sharedObject)

Triggered when sharedObject created

Parameters:

sharedObject - sharedObject

onSharedObjectDestroy

public void **onSharedObjectDestroy**([ISharedObject](#) sharedObject)

Triggered when sharedObject destroyed

Parameters:

sharedObject - sharedObject

onSharedObjectConnect

public void **onSharedObjectConnect**([ISharedObject](#) sharedObject, [IClient](#) client)

Triggered when client connects to sharedObject

Parameters:

sharedObject - sharedObject

client - client

onSharedObjectDisconnect

```
public void onSharedObjectDisconnect(ISharedObject sharedObject,  
    IClient client)
```

Triggered when client disconnects from sharedObject

Parameters:

sharedObject - sharedObject

client - client

com.wowza.wms.sharedobject Interface ISharedObjects

public interface **ISharedObjects**
extends

ISharedObjects: public interface to SharedObjects. Represent the list of shared objects managed by IApplicationInstance.

Method Summary

void	<u>addSharedObjectListener</u> (<u>ISharedObjectNotify</u> sharedObjectListener)
	Add a shared object listener.
void	<u>disconnect</u> (<u>IClient</u> client)
	Disconnect client from all shared objects in list.
boolean	<u>exists</u> (<u>ISharedObject</u> sharedObject)
	Is sharedObject in this list (by shared object reference).
boolean	<u>exists</u> (String objectName)
	Is sharedObject in this list (by name).
void	<u>flush</u> ()
	Flush all persistent shared objects to disk.
<u>ISharedObject</u>	<u>get</u> (String name)
	Get shared object by name.
java.util.List	<u>getObjectNames</u> ()
	Get a list of shared object names.
<u>ISharedObject</u>	<u>getOrCreate</u> (String name)
	Get shared object by name if it does not exist create a new shared object with the given name.
String	<u>getStorageDir</u> ()
	Get the storage directory for all shared objects in list.
boolean	<u>isPersistent</u> ()
	Are shared objects in list persistent.
void	<u>load</u> ()
	Load persistent shared objects from file system.
void	<u>put</u> (String objectName, <u>ISharedObject</u> sharedObject)
	Add or replace a shared object.
void	<u>remove</u> (String objectName)
	Remove a shared object.
void	<u>removeClient</u> (<u>IClient</u> client)
	Remove a client from any shared object that it is connected to in this list.
void	<u>removeSharedObjectListener</u> (<u>ISharedObjectNotify</u> sharedObjectListener)
	Remove a shared object listener.

void	<code>setPersistent</code> (boolean isPersistent) Set is shared object in list persistent
void	<code>setStorageDir</code> (String storageDir) Set the storage directory for all shared objects in list.
int	<code>size</code> () Get number of shared objects.

Methods

size

```
public int size()
```

Get number of shared objects.

Returns:

number of shared objects

get

```
public ISharedObject get(String name)
```

Get shared object by name. If the shared object does not exist null will be returned. (see ISharedObjects.getOrCreate)

Parameters:

name - shared object name

Returns:

shared object

getOrCreate

```
public ISharedObject getOrCreate(String name)
```

Get shared object by name if it does not exist create a new shared object with the given name.

Parameters:

name - shared object name

Returns:

shared object

getObjectNames

```
public java.util.List getObjectNames()
```

Get a list of shared object names.

Returns:

list of shared object names

(continued from last page)

put

```
public void put(String objectName,  
    ISharedObject sharedObject)
```

Add or replace a shared object.

Parameters:

objectName - shared object name
sharedObject - shared object

remove

```
public void remove(String objectName)
```

Remove a shared object.

Parameters:

objectName - shared object name

exists

```
public boolean exists(String objectName)
```

Is sharedObject in this list (by name).

Parameters:

objectName - shared object name

Returns:

true if shared object in list

exists

```
public boolean exists(ISharedObject sharedObject)
```

Is sharedObject in this list (by shared object reference).

Parameters:

sharedObject - shared object

Returns:

true if shared object in list

isPersistent

```
public boolean isPersistent()
```

Are shared objects in list persistent.

Returns:

true if shared objects in list are persistent

setPersistent

```
public void setPersistent(boolean isPersistent)
```

Set if shared object in list persistent

(continued from last page)

Parameters:

isPersistent - true is shared objects in list are persistent

removeClient

```
public void removeClient(IClient client)
```

Remove a client from any shared object that it is connected to in this list.

Parameters:

client - client

flush

```
public void flush()
```

Flush all persistent shared objects to disk.

disconnect

```
public void disconnect(IClient client)
```

Disconnect client from all shared objects in list.

Parameters:

client - client

addSharedObjectListener

```
public void addSharedObjectListener(ISharedObjectNotify sharedObjectListener)
```

Add a shared object listener. The listener will receive the following events: onSharedObjectCreate, onSharedObjectDestroy, onSharedObjectConnect, onSharedObjectDisconnect.

Parameters:

sharedObjectListener

removeSharedObjectListener

```
public void removeSharedObjectListener(ISharedObjectNotify sharedObjectListener)
```

Remove a shared object listener.

Parameters:

sharedObjectListener

getStorageDir

```
public String getStorageDir()
```

Get the storage directory for all shared objects in list.

Returns:

storage dir

setStorageDir

```
public void setStorageDir(String storageDir)
```

(continued from last page)

Set the storage directory for all shared objects in list.

Parameters:

storageDir

load

```
public void load()
```

Load persistent shared objects from file system.

com.wowza.wms.sharedobject Interface ISharedObjectSlot

public interface **ISharedObjectSlot**
extends

ISharedObjectSlot: public interface to SharedObjectSlot class.

Method Summary

AMFData	getData() Get slot data as AMFData object.
int	getLastClientId() Get client id of client that performed last operation on slot.
int	getLastOperation() Get last slot (property) operation.
String	getName() Get slot (property) name.
int	getSlotVersion() Get slot version
int	getSoVersion() Get parent shared object version
void	incSlotVersion() Increment slot version by 1.
void	init (String name, AMFData data, int slotVersion) Initialize shared object slot
void	setData (AMFData data) Set slot data as AMFData object.
void	setData (byte[] data) Set slot data as byte[].
void	setData (byte[] data, AMFDataContextDeserialize context) Set slot data as byte[].
void	setLastClientId (int lastClientId) Set client id of client that performed last operation on slot.
void	setLastOperation (int lastOperation) Set last slot (property) operation.
void	setName (String name) Set slot (property) name
void	setSlotVersion (int slotVersion) Set slot version

void	setSoVersion (int soVersion) Set parent shared object version
------	--

Methods

init

```
public void init(String name,  
    AMFData data,  
    int slotVersion)
```

Initialize shared object slot

Parameters:

name - slot (property) name
data - data
slotVersion - version

getData

```
public AMFData getData()
```

Get slot data as AMFData object.

Returns:

slot data as AMFData object

setData

```
public void setData(byte[] data,  
    AMFDataContextDeserialize context)
```

Set slot data as byte[]. Data will be deserialized as AMFData object.

Parameters:

data - data as byte[]
context - deserialization context

setData

```
public void setData(byte[] data)
```

Set slot data as byte[]. Data will be deserialized as AMFData object.

Parameters:

data - data as byte[]

setData

```
public void setData(AMFData data)
```

Set slot data as AMFData object.

Parameters:

data - data as AMFData object

getName

```
public String getName()
```

Get slot (property) name.

Returns:

slot (property) name

setName

```
public void setName(String name)
```

Set slot (property) name

Parameters:

name - slot (property) name

getSlotVersion

```
public int getSlotVersion()
```

Get slot version

Returns:

slot version

setSlotVersion

```
public void setSlotVersion(int slotVersion)
```

Set slot version

Parameters:

slotVersion - slot version

incSlotVersion

```
public void incSlotVersion()
```

Increment slot version by 1.

getSoVersion

```
public int getSoVersion()
```

Get parent shared object version

Returns:

parent shared object version

setSoVersion

```
public void setSoVersion(int soVersion)
```

Set parent shared object version

(continued from last page)

Parameters:

soVersion - parent shared object version

getLastOperation

```
public int getLastOperation()
```

Get last slot (property) operation. See ISharedObject.SHAREDOBJECT_CMD_*.

Returns:

last slot (property) operation (ISharedObject.SHAREDOBJECT_CMD_*)

setLastOperation

```
public void setLastOperation(int lastOperation)
```

Set last slot (property) operation. See ISharedObject.SHAREDOBJECT_CMD_*.

Parameters:

lastOperation - last slot (property) operation (ISharedObject.SHAREDOBJECT_CMD_*)

getLastClientId

```
public int getLastClientId()
```

Get client id of client that performed last operation on slot.

Returns:

client id of client that performed last operation on slot

setLastClientId

```
public void setLastClientId(int lastClientId)
```

Set client id of client that performed last operation on slot.

Parameters:

lastClientId - client id of client that performed last operation on slot

com.wowza.wms.sharedobject Interface ISharedObjectSlotNotify

public interface **ISharedObjectSlotNotify**
extends

ISharedObjectNotify: listener interface used by ISharedObject addSlotListener

Method Summary

void	onSlotDelete (ISharedObject sharedObject, ISharedObjectSlot slot) Triggered when sharedObject slot value deleted
void	onSlotSetValue (ISharedObject sharedObject, ISharedObjectSlot slot) Triggered when sharedObject slot value set

Methods

onSlotSetValue

public void **onSlotSetValue**([ISharedObject](#) sharedObject, [ISharedObjectSlot](#) slot)

Triggered when sharedObject slot value set

Parameters:

sharedObject
slot

onSlotDelete

public void **onSlotDelete**([ISharedObject](#) sharedObject, [ISharedObjectSlot](#) slot)

Triggered when sharedObject slot value deleted

Parameters:

sharedObject
slot

com.wowza.wms.sharedobject Class SharedObject

java.lang.Object

└─com.wowza.wms.sharedobject.SharedObject

All Implemented Interfaces:

[ISharedObject](#)

public class **SharedObject**
extends Object
implements [ISharedObject](#)

Field Summary

public static final	READACCESS Value: 0
public static final	WRITEACCESS Value: 1

Fields inherited from interface [com.wowza.wms.sharedobject.ISharedObject](#)

[FILEEXTENSION](#), [SHARED_OBJECT_CMD_CONNECT](#), [SHARED_OBJECT_CMD_CONNECTSUCCESS](#),
[SHARED_OBJECT_CMD_DELETE](#), [SHARED_OBJECT_CMD_DISCONNECT](#), [SHARED_OBJECT_CMD_ERROR](#),
[SHARED_OBJECT_CMD_SEND](#), [SHARED_OBJECT_CMD_SETVALUE](#), [SHARED_OBJECT_STATUS_CHANGE](#),
[SHARED_OBJECT_STATUS_CLEAR](#), [SHARED_OBJECT_STATUS_DELETE](#), [SHARED_OBJECT_STATUS_SUCCESS](#)

Constructor Summary

public	SharedObject (String name) Create new shared object
public	SharedObject (String name, boolean isPersistent, String storageDir) Create new shared object

Method Summary

void	acquire ()
void	addClient (IClient client)
void	addSlotListener (ISharedObjectSlotNotify slotListener)
void	clear ()
void	close ()

boolean	<u>containsProperty</u> (String slotName)
boolean	<u>containsSlot</u> (String slotName)
void	<u>deleteSlot</u> (<u>IClient</u> client, String slotName)
void	<u>deleteSlot</u> (String slotName)
void	<u>disconnect</u> (<u>IClient</u> client)
void	<u>flush</u> ()
static boolean[]	<u>getAccess</u> (<u>IClient</u> client, String soName)
java.util.List	<u>getClients</u> ()
void	<u>getClientUpdates</u> (<u>IClient</u> client)
String	<u>getName</u> ()
<u>ISharedObjects</u>	<u>getParent</u> ()
<u>AMFData</u>	<u>getProperty</u> (String slotName)
int	<u>getRefCount</u> ()
<u>ISharedObjectSlot</u>	<u>getSlot</u> (String name)
java.util.List	<u>getSlotNames</u> ()
java.util.List	<u>getSlots</u> ()
String	<u>getStorageDir</u> ()
int	<u>getVersion</u> ()
boolean	<u>isClient</u> (<u>IClient</u> client)
boolean	<u>isPersistent</u> ()
void	<u>load</u> ()
void	<u>lock</u> ()
void	<u>notifySlotDelete</u> (<u>ISharedObjectSlot</u> slot)
void	<u>notifySlotSetValue</u> (<u>ISharedObjectSlot</u> slot)

int	purge (int version)
void	putSlot (String name, ISharedObjectSlot slot)
void	release ()
void	removeClient (IClient client)
void	removeSlotListener (ISharedObjectSlotNotify slotListener)
void	send (String handlerName)
void	send (String handlerName, Object[] params)
void	sendInternal (IClient client, String handlerName, byte[] msgBytes)
void	sendInternal (IClient client, String handlerName, byte[] msgBytes, AMFDataContextDeserialize context)
void	sendInternal (String handlerName, AMFData[] params)
void	setName (String name)
void	setParent (ISharedObjects parent)
void	setPersistent (boolean isPersistent)
void	setProperty (String slotName, AMFData data)
void	setProperty (String slotName, boolean value)
void	setProperty (String slotName, java.util.Date value)
void	setProperty (String slotName, double value)
void	setProperty (String slotName, int value)
void	setProperty (String slotName, long value)
void	setProperty (String slotName, String value)
void	setSlotValue (IClient client, String slotName, AMFData amfData)
void	setSlotValue (IClient client, String slotName, byte[] byteData)
void	setSlotValue (IClient client, String slotName, byte[] byteData, AMFData amfData)

void	setSlotValue(IClient client, String slotName, byte[] byteData, AMFData amfData, AMFDataContextDeserialize context)
void	setSlotValue(IClient client, String slotName, byte[] byteData, AMFDataContextDeserialize context)
void	setStorageDir (String storageDir)
void	setVersion (int version)
int	size ()
void	unlock ()
void	writeDeleteError(IClient client, String soName, boolean isPersistent, String slotName, String errorMsg)
static void	writeError(IClient client, String soName, boolean isPersistent, String errorMsg, boolean isConnect)
void	writeSetValueError(IClient client, String soName, boolean isPersistent, String slotName, String errorMsg)

Methods inherited from class [java.lang.Object](#)

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

Methods inherited from interface [com.wowza.wms.sharedobject.ISharedObject](#)

[acquire](#), [addClient](#), [addSlotListener](#), [clear](#), [close](#), [containsProperty](#), [containsSlot](#), [deleteSlot](#), [deleteSlot](#), [disconnect](#), [flush](#), [getClients](#), [getName](#), [getParent](#), [getProperty](#), [getRefCount](#), [getSlot](#), [getSlotNames](#), [getSlots](#), [getStorageDir](#), [getVersion](#), [isClient](#), [isPersistent](#), [lock](#), [purge](#), [putSlot](#), [release](#), [removeClient](#), [removeSlotListener](#), [send](#), [send](#), [setName](#), [setPersistent](#), [setProperty](#), [setProperty](#), [setProperty](#), [setProperty](#), [setProperty](#), [setProperty](#), [setStorageDir](#), [setVersion](#), [size](#), [unlock](#), [writeDeleteError](#), [writeSetValueError](#)

Fields

READACCESS

```
public static final int READACCESS
```

Constant value: **0**

WRITEACCESS

```
public static final int WRITEACCESS
```

Constant value: **1**

(continued from last page)

Constructors

SharedObject

```
public SharedObject(String name)
```

Create new shared object

Parameters:

name - shared object name

SharedObject

```
public SharedObject(String name,  
                    boolean isPersistent,  
                    String storageDir)
```

Create new shared object

Parameters:

name - shared object name

isPersistent - is persistent

storageDir - storage directory for persistent shared object

Methods

size

```
public int size()
```

purge

```
public int purge(int version)
```

clear

```
public void clear()
```

getSlots

```
public java.util.List getSlots()
```

getSlotNames

```
public java.util.List getSlotNames()
```

(continued from last page)

getSlot

```
public ISharedObjectSlot getSlot(String name)
```

putSlot

```
public void putSlot(String name,  
    ISharedObjectSlot slot)
```

getClients

```
public java.util.List getClients()
```

isClient

```
public boolean isClient(IClient client)
```

addClient

```
public void addClient(IClient client)
```

removeClient

```
public void removeClient(IClient client)
```

acquire

```
public void acquire()
```

release

```
public void release()
```

getRefCount

```
public int getRefCount()
```

isPersistent

```
public boolean isPersistent()
```

(continued from last page)

setPersistent

```
public void setPersistent(boolean isPersistent)
```

getVersion

```
public int getVersion()
```

setVersion

```
public void setVersion(int version)
```

deleteSlot

```
public void deleteSlot(String slotName)
```

containsProperty

```
public boolean containsProperty(String slotName)
```

containsSlot

```
public boolean containsSlot(String slotName)
```

deleteSlot

```
public void deleteSlot(IClient client,  
                        String slotName)
```

getProperty

```
public AMFData getProperty(String slotName)
```

setProperty

```
public void setProperty(String slotName,  
                        boolean value)
```

setProperty

```
public void setProperty(String slotName,  
    java.util.Date value)
```

setProperty

```
public void setProperty(String slotName,  
    double value)
```

setProperty

```
public void setProperty(String slotName,  
    long value)
```

setProperty

```
public void setProperty(String slotName,  
    int value)
```

setProperty

```
public void setProperty(String slotName,  
    String value)
```

setProperty

```
public void setProperty(String slotName,  
    AMFData data)
```

setSlotValue

```
public void setSlotValue(IClient client,  
    String slotName,  
    AMFData amfData)
```

setSlotValue

```
public void setSlotValue(IClient client,  
    String slotName,  
    byte[] byteData)
```

(continued from last page)

setSlotValue

```
public void setSlotValue(IClient client,  
    String slotName,  
    byte[] byteData,  
    AMFDataContextDeserialize context)
```

setSlotValue

```
public void setSlotValue(IClient client,  
    String slotName,  
    byte[] byteData,  
    AMFData amfData)
```

setSlotValue

```
public void setSlotValue(IClient client,  
    String slotName,  
    byte[] byteData,  
    AMFData amfData,  
    AMFDataContextDeserialize context)
```

writeError

```
public static void writeError(IClient client,  
    String soName,  
    boolean isPersistent,  
    String errorMsg,  
    boolean isConnect)
```

writeSetValueError

```
public void writeSetValueError(IClient client,  
    String soName,  
    boolean isPersistent,  
    String slotName,  
    String errorMsg)
```

writeDeleteError

```
public void writeDeleteError(IClient client,  
    String soName,  
    boolean isPersistent,  
    String slotName,  
    String errorMsg)
```

getClientUpdates

```
public void getClientUpdates(IClient client)
```

(continued from last page)

disconnect

```
public void disconnect(IClient client)
```

getName

```
public String getName()
```

setName

```
public void setName(String name)
```

send

```
public void send(String handlerName,  
                 Object[] params)
```

send

```
public void send(String handlerName)
```

sendInternal

```
public void sendInternal(String handlerName,  
                        AMFData\[\] params)
```

sendInternal

```
public void sendInternal(IClient client,  
                        String handlerName,  
                        byte[] msgBytes)
```

sendInternal

```
public void sendInternal(IClient client,  
                        String handlerName,  
                        byte[] msgBytes,  
                        AMFDataContextDeserialize context)
```

(continued from last page)

addSlotListener

```
public void addSlotListener(ISharedObjectSlotNotify slotListener)
```

removeSlotListener

```
public void removeSlotListener(ISharedObjectSlotNotify slotListener)
```

notifySlotSetValue

```
public void notifySlotSetValue(ISharedObjectSlot slot)
```

notifySlotDelete

```
public void notifySlotDelete(ISharedObjectSlot slot)
```

getStorageDir

```
public String getStorageDir()
```

setStorageDir

```
public void setStorageDir(String storageDir)
```

flush

```
public void flush()
```

load

```
public void load()
```

close

```
public void close()
```

lock

```
public void lock()
```

(continued from last page)

unlock

```
public void unlock()
```

setParent

```
public void setParent(ISharedObjects parent)
```

getParent

```
public ISharedObjects getParent()
```

getAccess

```
public static boolean[] getAccess(IClient client,  
    String soName)
```

Package

com.wowza.wms.stream

com.wowza.wms.stream

Class FastPlaySettings

java.lang.Object

└─com.wowza.wms.stream.FastPlaySettings

public class **FastPlaySettings**
extends Object

FastPlaySettings: data container for fast play settings

Constructor Summary

public	FastPlaySettings() Create empty object
public	FastPlaySettings(double multiplier, int fps, int direction) Create object

Method Summary

int	getDirection() Get direction (1 forward, -1 reverse)
int	getFps() Get frames per second
double	getMultiplier() Get speed of fast play
long	getStartTC() Get the timecode (milliseconds) where this fast play started
long	getStartTCOffset() Get the timecode (milliseconds) where this fast play started (not sure why we have both values)
void	setDirection(int direction) Set direction (1 forward, -1 reverse)
void	setFps(int fps) Set frames per second
void	setMultiplier(double multiplier) Set speed of fast play
void	setStartTC(long startTC) Set the timecode (milliseconds) where this fast play started
void	setStartTCOffset(long startTCOffset) Set the timecode (milliseconds) where this fast play started (not sure why we have both values)

Methods inherited from class java.lang.Object


```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

Constructors

FastPlaySettings

```
public FastPlaySettings()
```

Create empty object

FastPlaySettings

```
public FastPlaySettings(double multiplier,  
                        int fps,  
                        int direction)
```

Create object

Parameters:

multiplier - speed

fps - frames per second

direction - direction (1 forward, -1 reverse)

Methods

getFps

```
public int getFps()
```

Get frames per second

Returns:

frames per second

setFps

```
public void setFps(int fps)
```

Set frames per second

Parameters:

fps - frames per second

getMultiplier

```
public double getMultiplier()
```

Get speed of fast play

Returns:

speed of fast play

(continued from last page)

setMultiplier

```
public void setMultiplier(double multiplier)
```

Set speed of fast play

Parameters:

multiplier - speed of fast play

getDirection

```
public int getDirection()
```

Get direction (1 forward, -1 reverse)

Returns:

direction (1 forward, -1 reverse)

setDirection

```
public void setDirection(int direction)
```

Set direction (1 forward, -1 reverse)

Parameters:

direction - direction (1 forward, -1 reverse)

getStartTC

```
public long getStartTC()
```

Get the timecode (milliseconds) where this fast play started

Returns:

timecode of start of fast play (milliseconds)

setStartTC

```
public void setStartTC(long startTC)
```

Set the timecode (milliseconds) where this fast play started

Parameters:

startTC - timecode of start of fast play (milliseconds)

getStartTCOffset

```
public long getStartTCOffset()
```

Get the timecode (milliseconds) where this fast play started (not sure why we have both values)

Returns:

timecode of start of fast play (milliseconds)

setStartTCOffset

```
public void setStartTCOffset(long startTCOffset)
```

(continued from last page)

Set the timecode (milliseconds) where this fast play started (not sure why we have both values)

Parameters:

`startTCOffset` - timecode of start of fast play (milliseconds)

com.wowza.wms.stream Interface IMediaIndexItem

public interface **IMediaIndexItem**
extends

IMediaIndexItem: generic interface to mediaIndex

Field Summary

public static final	MISSING media file is missing Value: **missing**
---------------------	---

Method Summary

IMediaIndexItem	clone() Make a clone copy of media index
String	getChecksum() Get checksum for media index
int	getExtent() Get the duration (milliseconds) of media index
long	getLastAccessed() Get last time media index was accessed (milliseconds)
boolean	isHitEnd() Is media index complete
void	setChecksum(IRandomAccessReader file) Set media index checksum.
void	setChecksum(String checksum) Set media index checksum
void	setLastAccessed(long time) Set last time media index accessed (milliseconds)
void	touch() Set last time accessed to current time

Fields

MISSING

public static final java.lang.String **MISSING**

media file is missing
Constant value: ****missing****

(continued from last page)

Methods

getExtent

```
public int getExtent()
```

Get the duration (milliseconds) of media index

Returns:

duration (milliseconds)

getChecksum

```
public String getChecksum()
```

Get checksum for media index

Returns:

checksum

setChecksum

```
public void setChecksum(String checksum)
```

Set media index checksum

Parameters:

checksum

setChecksum

```
public void setChecksum(IRandomAccessReader file)
```

Set media index checksum. Extract data from File spec

Parameters:

file

clone

```
public IMediaIndexItem clone()
```

Make a clone copy of media index

Returns:

shallow copy of media index

getLastAccessed

```
public long getLastAccessed()
```

Get last time media index was accessed (milliseconds)

Returns:

time media index was accessed (milliseconds)

(continued from last page)

setLastAccessed

```
public void setLastAccessed(long time)
```

Set last time media index accessed (milliseconds)

Parameters:

time - media index was accessed (milliseconds)

isHitEnd

```
public boolean isHitEnd()
```

Is media index complete

Returns:

true if media index is complete

touch

```
public void touch()
```

Set last time accessed to current time

com.wowza.wms.stream

Interface IMediaListProvider

public interface **IMediaListProvider**
extends

Method Summary	
MediaList	resolveMediaList (IMediaListReader mediaListReader, IMediaStream stream, String streamName) Called to resolve a amlst:streamname to a media list

Methods

resolveMediaList

```
public MediaList resolveMediaList(IMediaListReader mediaListReader,
    IMediaStream stream,
    String streamName)
```

Called to resolve a amlst:streamname to a media list

Parameters:

- mediaListReader - mediaListReader
- stream - media stream
- streamName - stream name

Returns:

media list

com.wowza.wms.stream Interface IMediaListReader

public interface **IMediaListReader**
extends

Method Summary

void	<code>close()</code> close file
<code>IHTTPStreamerSession</code>	<code>getHTTPStreamerSession()</code> Get the HTTPStreamerSession associated with this media list reader
String	<code>getMediaExtension()</code> Get media extension
<code>MediaList</code>	<code>getMediaList()</code> Get the MediaList object
String	<code>getPath()</code> Get abstract path to the media item
void	<code>init(IApplicationInstance</code> appInstance, <code>IMediaStream</code> stream, String mediaReadType, String basePath, String mediaName, <code>IHTTPStreamerSession</code> httpStreamerSession) Initialize mediaReader
boolean	<code>isOpen()</code> is file open
void	<code>open(String</code> basePath, String name) Open the file for reading
void	<code>setHTTPStreamerSession(IHTTPStreamerSession</code> httpStreamerSession) Set the HTTPStreamerSession associated with this media list reader
void	<code>setMediaReaderItem(MediaReaderItem</code> mediaReaderItem) Set the mediaReader item definition
void	<code>setProperties(WMSProperties</code> properties) Set the properties for this media reader

Methods

init

```
public void init(IApplicationInstance appInstance,  
                 IMediaStream stream,  
                 String mediaReadType,  
                 String basePath,  
                 String mediaName,  
                 IHTTPStreamerSession httpStreamerSession)
```


(continued from last page)

Initialize mediaReader

Parameters:

stream - parent stream
mediaReadType - media reader type id. Example: flv
basePath - Base path for application instance
mediaName - name of the media item

setMediaReaderItem

```
public void setMediaReaderItem(MediaReaderItem mediaReaderItem)
```

Set the mediaReader item definition

Parameters:

mediaReaderItem

open

```
public void open(String basePath,  
String name)
```

Open the file for reading

Parameters:

basePath - base path to file
name - file name without extension

close

```
public void close()
```

close file

isOpen

```
public boolean isOpen()
```

is file open

Returns:

is file open

setProperties

```
public void setProperties(WMSPProperties properties)
```

Set the properties for this media reader

Parameters:

properties - properties

getPath

```
public String getPath()
```

Get abstract path to the media item

Returns:

(continued from last page)

abstract path to the media item

getMediaExtension

```
public String getMediaExtension()
```

Get media extension

Returns:

media extension

getMediaList

```
public MediaList getMediaList()
```

Get the MediaList object

Returns:

MediaList object

getHTTPStreamerSession

```
public IHTTPStreamerSession getHTTPStreamerSession()
```

Get the HTTPStreamerSession associated with this media list reader

Returns:

HTTPStreamerSession

setHTTPStreamerSession

```
public void setHTTPStreamerSession(IHTTPStreamerSession httpStreamerSession)
```

Set the HTTPStreamerSession associated with this media list reader

Parameters:

httpStreamerSession - HTTPStreamerSession

com.wowza.wms.stream Interface IMediaReader

public interface **IMediaReader**
extends

IMediaReader: generic media reader interface. All media types implement this interface to interact with PlaylistPlayer.

Field Summary

public static final	<u>CONTENTTYPE_MEDIA</u> Value: 1
public static final	<u>CONTENTTYPE_MEDIALIST</u> Value: 2
public static final	<u>DEFAULT_RANDOMACCESSREADER</u> Value: com.wowza.io.DirectRandomAccessReader
public static final	<u>DEFAULT_RANDOMACCESSREADEROPTIMIZER</u> Value: com.wowza.io.RandomAccessReadOptimizer
public static final	<u>PLAYEVENT_AFTERBUFFERFILL</u> Value: 5
public static final	<u>PLAYEVENT_AFTERMETADATA</u> Value: 3
public static final	<u>PLAYEVENT_BEFOREBUFFERFILL</u> Value: 4
public static final	<u>PLAYEVENT_BEFOREMETADATA</u> Value: 2
public static final	<u>PLAYEVENT_STARTPLAYBACK</u> Value: 1
public static final	<u>SEEK_EXACT</u> Seek direction: closest frame (audio, video) (key, no-key) Value: 4
public static final	<u>SEEK_KEYCLOSE</u> Seek direction: closest key frame Value: 3
public static final	<u>SEEK_KEYDOWN</u> Seek direction: down to closets key frame Value: 2

public static final	SEEK_KEYUP Seek direction: up to closets key frame Value: 1
public static final	SEEKTARGET_AUDIO Value: 3
public static final	SEEKTARGET_ENHANCED Value: 4
public static final	SEEKTARGET_VIDEOKEYFRAME Value: 1

Method Summary

void	close() close file
long	getDuration() Get duration or time (milliseconds) of the media file
long	getLength() Get the stream length in bytes
String	getMediaExtension() Get media extension
java.util.List	getMetadata() Get a collection of metadata packets in ByteBuffers for this file.
String	getPath() Get abstract path to the media item
IMediaReaderStreamPosition	getStreamPosition() Get a reference to the current stream position
void	init() (IApplicationInstance appInstance, IMediaStream stream, String mediaReadType, String basePath, String mediaName) Initialize mediaReader
boolean	isOpen() is file open
void	open() (String basePath, String name) Open the file for reading
void	rewind() rewind file to start
PlaylistSeekResult	seek() (long timecode, int seektype) seek to timecode in file
int	sendZeroLengthPacket() (int type, int timecode, boolean isAbsolute, IMediaStream stream, java.io.OutputStream out, AMFObj wmsObj, PlaylistCursor flvCursor, PlaylistWriteControl control, PlaylistReaderWriteResults results, long[] sizes, boolean isForceTCZero)

void	<code>setMediaReaderItem</code> (<code>MediaReaderItem</code> mediaReaderItem) Set the mediaReader item definition
void	<code>setProperties</code> (<code>WMSProperties</code> properties) Set the properties for this media reader
void	<code>setStreamPosition</code> (<code>IMediaReaderStreamPosition</code> pos) Set the file position within the media file
void	<code>startPlayback</code> () Called each time the player being playback (before the buffer it filled)
int	<code>writeGeneratedKeyFrame</code> (<code>IMediaStream</code> stream, java.io.OutputStream out, <code>AMFObj</code> wmsObjAudio, <code>AMFObj</code> wmsObjVideo, <code>AMFObj</code> wmsObjData, <code>PlaylistCursor</code> flvCursor, <code>PlaylistWriteControl</code> control, <code>PlaylistReaderWriteResults</code> results, long[] sizes, <code>FastPlaySettings</code> fastPlaySettings, boolean isForceTCZero) From current location in file generate a key frame (enhanced seek) and write it to out
int	<code>writePackets</code> (<code>IMediaStream</code> stream, java.io.OutputStream out, <code>AMFObj</code> wmsObjAudio, <code>AMFObj</code> wmsObjVideo, <code>AMFObj</code> wmsObjData, <code>PlaylistCursor</code> flvCursor, <code>PlaylistWriteControl</code> control, <code>PlaylistReaderWriteResults</code> results, long[] sizes, <code>FastPlaySettings</code> fastPlaySettings, boolean isForceTCZero) From current location in file write packets to output
int	<code>writePackets</code> (java.util.List packetList, <code>PlaylistCursor</code> flvCursor, <code>PlaylistWriteControl</code> control, <code>PlaylistReaderWriteResults</code> results, long[] sizes, <code>FastPlaySettings</code> fastPlaySettings) Write packets to the packetList as AMFPackets

Fields

CONTENTTYPE_MEDIA

```
public static final int CONTENTTYPE_MEDIA
```

Constant value: **1**

CONTENTTYPE_MEDIALIST

```
public static final int CONTENTTYPE_MEDIALIST
```

Constant value: **2**

DEFAULT_RANDOMACCESSREADER

```
public static final java.lang.String DEFAULT_RANDOMACCESSREADER
```

Constant value: **com.wowza.io.DirectRandomAccessReader**

DEFAULT_RANDOMACCESSREADEROPTIMIZER

```
public static final java.lang.String DEFAULT_RANDOMACCESSREADEROPTIMIZER
```

Constant value: **com.wowza.io.RandomAccessReadOptimizer**

SEEKTARGET_VIDEOKEYFRAME

```
public static final int SEEKTARGET_VIDEOKEYFRAME
```

Constant value: **1**

SEEKTARGET_AUDIO

```
public static final int SEEKTARGET_AUDIO
```

Constant value: **3**

SEEKTARGET_ENHANCED

```
public static final int SEEKTARGET_ENHANCED
```

Constant value: **4**

PLAYEVENT_STARTPLAYBACK

```
public static final int PLAYEVENT_STARTPLAYBACK
```

Constant value: **1**

PLAYEVENT_BEFOREMETADATA

```
public static final int PLAYEVENT_BEFOREMETADATA
```

Constant value: **2**

PLAYEVENT_AFTERMETADATA

```
public static final int PLAYEVENT_AFTERMETADATA
```

Constant value: **3**

PLAYEVENT_BEFOREBUFFERFILL

```
public static final int PLAYEVENT_BEFOREBUFFERFILL
```

Constant value: **4**

PLAYEVENT_AFTERBUFFERFILL

```
public static final int PLAYEVENT_AFTERBUFFERFILL
```

Constant value: **5**

(continued from last page)

SEEK_KEYUP

```
public static final int SEEK_KEYUP
```

Seek direction: up to closets key frame
Constant value: **1**

SEEK_KEYDOWN

```
public static final int SEEK_KEYDOWN
```

Seek direction: down to closets key frame
Constant value: **2**

SEEK_KEYCLOSE

```
public static final int SEEK_KEYCLOSE
```

Seek direction: closest key frame
Constant value: **3**

SEEK_EXACT

```
public static final int SEEK_EXACT
```

Seek direction: closest frame (audio, video) (key, no-key)
Constant value: **4**

Methods

init

```
public void init(IApplicationInstance appInstance,  
                IMediaStream stream,  
                String mediaReadType,  
                String basePath,  
                String mediaName)
```

Initialize mediaReader

Parameters:

stream - parent stream
mediaReadType - media reader type id. Example: flv
basePath - Base path for application instance
mediaName - name of the media item

setMediaReaderItem

```
public void setMediaReaderItem(MediaReaderItem mediaReaderItem)
```

Set the mediaReader item definition

Parameters:

mediaReaderItem

open

```
public void open(String basePath,  
                String name)
```

(continued from last page)

Open the file for reading

Parameters:

basePath - base path to file
name - file name without extension

getMetadata

```
public java.util.List getMetadata()
```

Get a collection of metadata packets in ByteBuffers for this file. You can use 'new AMFDataList(data)' to convert to AMF objects.

Returns:

collection of metadata packets in ByteBuffers

rewind

```
public void rewind()
```

rewind file to start

close

```
public void close()
```

close file

isOpen

```
public boolean isOpen()
```

is file open

Returns:

is file open

seek

```
public PlaylistSeekResult seek(long timecode,  
int seektype)
```

seek to timecode in file

Parameters:

timecode - timecode (milliseconds) to seek to
seektype - seek type IMediaReader.SEEK_*

Returns:

detailed results object of seek or null if failure

(continued from last page)

writePackets

```
public int writePackets(IMediaStream stream,
    java.io.OutputStream out,
    AMFObj wmsObjAudio,
    AMFObj wmsObjVideo,
    AMFObj wmsObjData,
    PlaylistCursor flvCursor,
    PlaylistWriteControl control,
    PlaylistReaderWriteResults results,
    long[] sizes,
    FastPlaySettings fastPlaySettings,
    boolean isForceTCZero)
```

From current location in file write packets to output

Parameters:

stream - parent mediaStream
 out - output stream
 wmsObjAudio - audio channel amfObj
 wmsObjVideo - video channel amfObj
 wmsObjData - data channel amfObj
 flvCursor - current cursor that keeps track of timecodes and flags indicating position in media file
 control - write control flags. Mainly used to control when it should stop writing
 results - write results returned from this routine
 sizes - array of values that keep track of number of bytes written and number of packets for each data type (audio, video, data). See [IMediaStreamPlay.PLAYSIZES_*](#)
 fastPlaySettings - current fastPlay settings for the stream
 isForceTCZero - are we forcing all timecodes to zero

Returns:

number of bytes written

writePackets

```
public int writePackets(java.util.List packetList,
    PlaylistCursor flvCursor,
    PlaylistWriteControl control,
    PlaylistReaderWriteResults results,
    long[] sizes,
    FastPlaySettings fastPlaySettings)
```

Write packets to the packetList as AMFPackets

Parameters:

packetList - list to which to write packets
 flvCursor - current cursor that keeps track of timecodes and flags indicating position in media file
 control - write control flags. Mainly used to control when it should stop writing
 results - write results returned from this routine
 sizes - array of values that keep track of number of bytes written and number of packets for each data type (audio, video, data). See [IMediaStreamPlay.PLAYSIZES_*](#)
 fastPlaySettings - current fastPlay settings for the stream

Returns:

number of bytes written

(continued from last page)

writeGeneratedKeyFrame

```
public int writeGeneratedKeyFrame(IMediaStream stream,
    java.io.OutputStream out,
    AMFObj wmsObjAudio,
    AMFObj wmsObjVideo,
    AMFObj wmsObjData,
    PlaylistCursor flvCursor,
    PlaylistWriteControl control,
    PlaylistReaderWriteResults results,
    long[] sizes,
    FastPlaySettings fastPlaySettings,
    boolean isForceTCZero)
```

From current location in file generate a key frame (enhanced seek) and write it to out

Parameters:

stream - parent mediaStream
 out - output stream
 wmsObjAudio - audio channel amfObj
 wmsObjVideo - video channel amfObj
 wmsObjData - data channel amfObj
 flvCursor - current cursor that keeps track of timecodes and flags indicating position in media file
 control - write control flags. Mainly used to control when it should stop writing
 results - write results returned from this routine
 sizes - array of values that keep track of number of bytes written and number of packets for each data type (audio, video, data). See [IMediaStreamPlay.PLAYSIZES_*](#)
 fastPlaySettings - current fastPlay settings for the stream
 isForceTCZero - are we forcing all timecodes to zero

Returns:

number of bytes written

sendZeroLengthPacket

```
public int sendZeroLengthPacket(int type,
    int timecode,
    boolean isAbsolute,
    IMediaStream stream,
    java.io.OutputStream out,
    AMFObj wmsObj,
    PlaylistCursor flvCursor,
    PlaylistWriteControl control,
    PlaylistReaderWriteResults results,
    long[] sizes,
    boolean isForceTCZero)
```

startPlayback

```
public void startPlayback()
```

Called each time the player being playback (before the buffer it filled)

getDuration

```
public long getDuration()
```

Get duration or time (milliseconds) of the media file

Returns:

(continued from last page)

duration or time (milliseconds) of the media file

getLength

```
public long getLength()
```

Get the stream length in bytes

Returns:

stream length in bytes

getPath

```
public String getPath()
```

Get abstract path to the media item

Returns:

abstract path to the media item

getMediaExtension

```
public String getMediaExtension()
```

Get media extension

Returns:

media extension

getStreamPosition

```
public IMediaReaderStreamPosition getStreamPosition()
```

Get a reference to the current stream position

Returns:

stream position

setStreamPosition

```
public void setStreamPosition(IMediaReaderStreamPosition pos)
```

Set the file position within the media file

Parameters:

pos - stream position

setProperties

```
public void setProperties(WMSProperties properties)
```

Set the properties for this media reader

Parameters:

properties - properties

com.wowza.wms.stream Interface IMediaReaderActionNotify

public interface **IMediaReaderActionNotify**
extends

IMediaReaderActionNotify: listener interface for IMediaReader actions: See
IApplicationInstance.addMediaReaderListener(IMediaReaderActionNotify mediaReaderListener)

Method Summary

void	onMediaReaderClose (IMediaReader mediaReader, IMediaStream stream) Called when media reader is closed
void	onMediaReaderCreate (IMediaReader mediaReader) Called when media reader is created
void	onMediaReaderExtractMetaData (IMediaReader mediaReader, IMediaStream stream) Called after media reader metadata is extraced from the file
void	onMediaReaderInit (IMediaReader mediaReader, IMediaStream stream) Called after media reader is initialized
void	onMediaReaderOpen (IMediaReader mediaReader, IMediaStream stream) Called after media reader is opened

Methods

onMediaReaderCreate

public void **onMediaReaderCreate**([IMediaReader](#) mediaReader)

Called when media reader is created

Parameters:

mediaReader - media reader

onMediaReaderInit

public void **onMediaReaderInit**([IMediaReader](#) mediaReader, [IMediaStream](#) stream)

Called after media reader is initialized

Parameters:

mediaReader - media reader

stream - stream

(continued from last page)

onMediaReaderOpen

```
public void onMediaReaderOpen(IMediaReader mediaReader,  
    IMediaStream stream)
```

Called after media reader is opened

Parameters:

mediaReader - media reader
stream - stream

onMediaReaderExtractMetaData

```
public void onMediaReaderExtractMetaData(IMediaReader mediaReader,  
    IMediaStream stream)
```

Called after media reader metadata is extracted from the file

Parameters:

mediaReader - media reader
stream - stream

onMediaReaderClose

```
public void onMediaReaderClose(IMediaReader mediaReader,  
    IMediaStream stream)
```

Called when media reader is closed

Parameters:

mediaReader - media reader
stream - stream

com.wowza.wms.stream
Interface IMediaReaderEncInfo

public interface IMediaReaderEncInfo
extends

Method Summary	
MediaReaderEncInfo	getEncInfo()

Methods

getEncInfo
public [MediaReaderEncInfo](#) **getEncInfo()**

com.wowza.wms.stream

Interface IMediaReaderStreamPosition

public interface IMediaReaderStreamPosition

extends

IMediaReaderStreamPosition: Internal use

Method Summary	
boolean	isValid()

Methods

isValid

public boolean **isValid()**

com.wowza.wms.stream Interface IMediaStream

public interface **IMediaStream**
extends

IMediaStream: public, generic interface to mediaStream object. Upon creation of a new mediaStream, the client connection's default streamType value will be used to dynamically create a mediaStream object based on the definitions available in the vHosts.mediaStreamMap. All mediaStream implementations implement this interface.

Field Summary

public static final	AUDIOSAMPLEACCESS Value: 2
public static final	READACCESS Value: 0
public static final	VIDEOSAMPLEACCESS Value: 3
public static final	WRITEACCESS Value: 1

Method Summary

void	addAudioCodecConfigPacket (long timecode, AMFPacket packet) Set audio codec configuration packet (needed for H.264/AAC playback)
void	addAudioData (byte[] data, int offset, int size) Add data to current audio packet
void	addAudioExtraData (java.util.Map extraData) Add extra data to the current audio packet
void	addCallbackListener (IMediaStreamCallback callbackListener) Add a callback listener to listen for all stream metadata
void	addClientListener (IMediaStreamActionNotify actionListener) Add client listener.
void	addClientListener (IMediaStreamActionNotify2 actionListener) Add client listener.
void	addClientListener (IMediaStreamActionNotify3 actionListener) Add client listener.
void	addDataData (byte[] data, int offset, int size) Add data to current data packet

void	<u>addDataExtraData</u> (java.util.Map extraData) Add extra data to the current data packet
void	<u>addLivePacketListener</u> (<u>IMediaStreamLivePacketNotify</u> listener) Add a listener to receiver notification of new live AMF packets
void	<u>addVideoCodecConfigPacket</u> (long timecode, <u>AMFPacket</u> packet) Set video codec configuration packet (needed for H.264/AAC playback)
void	<u>addVideoData</u> (byte[] data, int offset, int size) Add data to current video packet
void	<u>addVideoExtraData</u> (java.util.Map extraData) Add extra data to the current video packet
void	<u>addVideoH264SEIListener</u> (<u>IMediaStreamH264SEINotify</u> h264SEIListener) Add an H.264 SEI listener.
void	<u>clear</u> () Delete media file pointed to by this mediaStream (be careful)
void	<u>clearFastPlaySettings</u> () Clear fastPlay settings
void	<u>clearLoggingValues</u> ()
void	<u>close</u> () Close mediaStream
void	<u>flush</u> () Force publishing packets to be flushed from the input buffers to the output buffers
boolean[]	<u>getAccess</u> (<u>IClient</u> client, String name) Get the read/write access to this stream for this client
<u>AMFPacket</u>	<u>getAudioCodecConfigPacket</u> (long timecode) Get audio codec configuration packet (needed for H.264/AAC playback)
int	<u>getAudioMissing</u> () Get number of audio bytes missing from current audio packet
int	<u>getAudioSize</u> () Get the size of the current audio packet that is being streamed from the client to the server
long	<u>getAudioTC</u> () Get last absolute audio timecode (milliseconds) sent to mediaStream
int	<u>getBufferTime</u> () Get buffer time for mediaStream (milliseconds)
byte[]	<u>getBurstStartStop</u> (boolean isStart) Get the dynamic streaming burst start/stop AMF packet
String	<u>getCacheName</u> () not used
<u>IClient</u>	<u>getClient</u> () Get parent client connection

int	<u>getClientId()</u> Get parent client connection (id)
String	<u>getContextStr()</u> Returns the stream context string in the form [application]/[appInstance]/[streamName].
int	<u>getDataMissing()</u> Get number of data bytes missing from current audio packet
int	<u>getDataSize()</u> Get the size of the current data packet that is being streamed from the client to the server
long	<u>getDataTC()</u> Get last absolute data timecode (milliseconds) sent to mediaStream
int	<u>getDataType()</u> Get the data packet type: (IVHost.CONTENTTYPE_DATA0 or IVHost.CONTENTTYPE_DATA3)
String	<u>getDvrRecorder()</u> Get the DVR Recorder for this stream
<u>ILiveStreamDvrRecorder</u>	<u>getDvrRecorder(String name)</u> Get the DVR Recorder interface to a stream by name
String	<u>getDvrRecorderList()</u> Get the comma separated list of DVR Recorder names being used by this stream (see conf/Dvr.xml)
String	<u>getDvrRepeater()</u> Get the DVR repeater name for this stream
<u>ElapsedTimer</u>	<u>getElapsedTime()</u> Get the interface to the elapse timer
String	<u>getExt()</u> Get media file extension
<u>FastPlaySettings</u>	<u>getFastPlaySettings()</u> Get current fastPlay settings
int	<u>getHeaderSize()</u> Get the last packet header size (debugging)
<u>IHTTPStreamerSession</u>	<u>getHTTPStreamerSession()</u> Get the HTTPStreamer session associated with this stream
<u>AMFPacket</u>	<u>getLastKeyFrame()</u> Get most recent video key frame
<u>AMFPacket</u>	<u>getLastPacket()</u> Get most recent live packet
String	<u>getLiveStreamPacketizer()</u> Get the live stream packetizer that this stream is using
<u>ILiveStreamPacketizer</u>	<u>getLiveStreamPacketizer(String name)</u> Get the LiveStreamPacketizer interface to a stream by name

String	<u>getLiveStreamPacketizerList()</u> Get the comma separated list of LiveStreamPacketizers names being used by this stream (see conf/LiveStreamPacketizers.xml)
String	<u>getLiveStreamRepeater()</u> Get the live stream repeater name for the stream
<u>ILiveStreamTranscoder</u>	<u>getLiveStreamTranscoder()</u> (String name) Get a live stream transcoder for this stream by name
String	<u>getLiveStreamTranscoderList()</u> Get the comma separated list of LiveStreamTranscoders names being used by this stream (see conf/LiveStreamTranscoders.xml)
java.util.Map	<u>getLiveStreamTranscoders()</u> Get the list of transcoders for this stream.
long	<u>getMaxTimecode()</u> Get the timecode of the latest received packet
<u>IOPerformanceCounter</u>	<u>getMediaIOPerformance()</u> Get IO performance counter
<u>IMediaStreamMetaDataProvider</u>	<u>getMetaDataProvider()</u> Get the metaData provider
String	<u>getName()</u> Get stream name
com.wowza.wms.netconnection.INetConnection	<u>getNetConnection()</u> Get parent netConnection (future server to server communication)
<u>IMediaStreamPlay</u>	<u>getPlayer()</u> Get underlying player (IMediaStreamPlay) object
java.util.List	<u>getPlayPackets()</u> Get all available live packets
<u>WMSProperties</u>	<u>getProperties()</u> Get mediaStream properties
int	<u>getPublishAudioCodecId()</u> Get the codec id of the most recently published audio packet
int	<u>getPublishVideoCodecId()</u> Get the codec id of the most recently published video packet
String	<u>getQueryStr()</u> Get play/publish name query string.
int	<u>getReceiveVideoFPS()</u> Set frame per seconds for video (not currently implemented)
<u>AMFObj</u>	<u>getRespAMFAudioObj()</u> Get audio response channel object
<u>AMFObj</u>	<u>getRespAMFDataObj()</u> Get data response channel object

AMFObj	getRespAMFVideoObj() Get video response channel object
RTPStream	getRTPStream() Get the RTP based stream this stream is associated with
int	getSrc() Get the stream id
java.io.File	getStreamFileForRead() Get the File object to read from a stream (get stream name, ext and query from stream object)
java.io.File	getStreamFileForRead(String name, String ext, String query) Get the File object to read from a stream (specify name, ext and query)
java.io.File	getStreamFileForWrite() Get the File object to write to a stream (get stream name, ext and query from stream object)
java.io.File	getStreamFileForWrite(String name, String ext, String query) Get the File object to write to a stream (specify name, ext and query)
MediaStreamMap	getStreams() Get parent mediaStreamMap (owned by applicationInstance)
String	getStreamType() Get mediaStream streamType
String	getUniqueStreamIdStr() Get a string that uniquely identifies this stream
AMFPacket	getVideoCodecConfigPacket(long timecode) Get video codec configuration packet (needed for H.264/AAC playback)
int	getVideoMissing() Get number of video bytes missing from current audio packet
int	getVideoSize() Get the size of the current video packet that is being streamed from the client to the server
long	getVideoTC() Get last absolute video timecode (milliseconds) sent to mediaStream
void	handleCallback(com.wowza.wms.request.RequestFunction function) Routes request function to callback handler onStatus, onPlayStatus or [method/handler]
boolean	idle() Allow the mediaStream to perform idle work (not currently in use)
long	incrementMediaInBytes(long increment) Increment the number of mediaStream bytes received
long	incrementMediaLossBytes(long bytes, long count) Increment the number of mediaStream loss bytes sent and number of packets sent
long	incrementMediaOutBytes(long bytes, long count) Increment the number of mediaStream bytes sent and number of packets sent
void	init(MediaStreamMap parent, int src, WMSProperties properties) Initialize the mediaStream object after creation.

void	<code>initLiveStreamRepeating</code> (String liveStreamPacketizer, String liveStreamRepeater) Initialize this stream for live stream repeating
boolean	<code>isAppend</code> () Is append to media file (only valid if isRecord)
boolean	<code>isClustered</code> () not used
boolean	<code>isMediaCasterPlay</code> () Is MediaCaster play enabled (if true, will trigger MediaCaster startup)
boolean	<code>isMergeOnMetadata</code> () If true, merge incoming onMetadata events with the current onMetadata event data.
boolean	<code>isOpen</code> () Is mediaStream open
boolean	<code>isPlay</code> () Is the stream a play stream (vs a publish stream)
boolean	<code>isPlaying</code> () Is mediaStream playing (or paused - false)
boolean	<code>isPublisherStream</code> () Returns true if stream was created by stream publisher API.
boolean	<code>isPublishStreamReady</code> (boolean checkAudio, boolean checkVideo) Returns true if the publishing stream contains enough video/audio data to start playback
boolean	<code>isReceiveAudio</code> () Is client currently receiving audio.
boolean	<code>isReceiveVideo</code> () Is client currently receiving video.
boolean	<code>isRecord</code> () Is this stream being recorded to a file
boolean	<code>isSendPlayStopLogEvent</code> () Get need to send a log event for stop
boolean	<code>isSendPublishStopLogEvent</code> () Get need to send a log event for publishing
boolean	<code>isSendRecordStopLogEvent</code> () Get need to send a log event for recording
boolean	<code>isTranscodeResult</code> () Is this stream the result of a transcode operation.
boolean	<code>isVideoH264SEIListenerEmpty</code> () Is H.264 SEI listener list empty.
double	<code>length</code> () Get length/duration (seconds) of media file pointed to by mediaStream

void	<u>notifyActionOnCodecInfoAudio</u> (com.wowza.wms.media.model.MediaCodecInfoAudio codecInfoAudio) Notify client listeners of audio codec information change
void	<u>notifyActionOnCodecInfoVideo</u> (com.wowza.wms.media.model.MediaCodecInfoVideo codecInfoVideo) Notify client listeners of video codec information change
void	<u>notifyActionOnMetaData</u> (<u>AMFPacket</u> metaDataPacket) Notify client listeners of onMetaData change
void	<u>notifyActionPause</u> (boolean isPause, long location) Notify client listeners of pause action
void	<u>notifyActionPauseRaw</u> (boolean isPause, long location) Notify client listeners of pauseRaw action
void	<u>notifyActionPlay</u> (String streamName, double playStart, double playLen, int playReset) Notify client listeners of play action
void	<u>notifyActionPublish</u> (String streamName, boolean isRecord, boolean isAppend) Notify client listeners of publish action
void	<u>notifyActionSeek</u> (double location) Notify client listeners of seek action
void	<u>notifyActionStop</u> () Notify client listeners of stop action
void	<u>notifyActionUnPublish</u> (String streamName, boolean isRecord, boolean isAppend) Notify client listeners of unpublish action
void	<u>notifyLivePacket</u> (<u>AMFPacket</u> packet) Notify new live AMF packet
void	<u>notifyVideoH264Packet</u> (<u>AMFPacket</u> packet, com.wowza.wms.media.h264.H264SEIMessages seiMessages) Notify H.264 SEI listener.
void	<u>packetComplete</u> () Invoked by requestAdapter when at the end of a set of packets
void	<u>publish</u> () Publish mediaStream
void	<u>putDvrRecorder</u> (String name, <u>ILiveStreamDvrRecorder</u> dvr) Add a live stream dvr to this stream
void	<u>putLiveStreamTranscoder</u> (String name, <u>ILiveStreamTranscoder</u> liveStreamTranscoder) Add a live stream transcoder to this stream
void	<u>registerCallback</u> (String handlerName, <u>IMediaStreamCallback</u> callback) Register a callback handler
void	<u>registerOnPlayStatus</u> (<u>IMediaStreamCallback</u> callback) Register onPlayStatus handler

void	<u>registerOnStatus</u> (<u>IMediaStreamCallback</u> callback) Register onStatus handler
void	<u>removeCallbackListener</u> (<u>IMediaStreamCallback</u> callbackListener) Remove a callback listener to listen for all stream metadata
void	<u>removeClientListener</u> (<u>IMediaStreamActionNotify</u> actionListener) Remove client listener.
void	<u>removeClientListener</u> (<u>IMediaStreamActionNotify2</u> actionListener) Remove client listener.
void	<u>removeClientListener</u> (<u>IMediaStreamActionNotify3</u> actionListener) Remove client listener.
<u>ILiveStreamDvrRecorder</u>	<u>removeDvrRecorder</u> (String name) Remove a live stream dvr by name
void	<u>removeLivePacketListener</u> (<u>IMediaStreamLivePacketNotify</u> listener) Remove a listener to receiver notification of new live AMF packets
<u>ILiveStreamTranscoder</u>	<u>removeLiveStreamTranscoder</u> (String name) Remove a live stream transcoder by name
void	<u>removeVideoH264SEIListener</u> (<u>IMediaStreamH264SEINotify</u> h264SEIListener) Remove an H.264 SEI listener.
void	<u>send</u> (String handlerName) Call client side NetStream method/handler with no parameters
void	<u>send</u> (String handlerName, Object[] params) Call client side NetStream method/handler
void	<u>sendAMF3</u> (String handlerName) Call client side NetStream method/handler with no parameters.
void	<u>sendAMF3</u> (String handlerName, Object[] params) Call client side NetStream method/handler.
int	<u>sendControlBytes</u> (int controlType, java.io.OutputStream out) Send playback control bytes.
void	<u>sendDirect</u> (String handlerName) Call client side NetStream method/handler and send event to underlying stream (will record event)
void	<u>sendDirect</u> (String handlerName, Object[] params) Call client side NetStream method/handler and send event to underlying stream (will record event)
void	<u>sendDirectAMF3</u> (String handlerName) Call client side NetStream method/handler and send event to underlying stream (will record event).
void	<u>sendDirectAMF3</u> (String handlerName, Object[] params) Call client side NetStream method/handler and send event to underlying stream (will record event).
int	<u>sendLivePlaySeek</u> (java.io.OutputStream out, String name, long timecode) Send onStatus(NetStream.Seek.Notify) event

int	<code>sendLivePlayStart</code> (java.io.OutputStream out, String name, long timecode, long timecodeOffset) Send onStatus(NetStream.Play.Start) event
int	<code>sendLivePlaySwitch</code> (java.io.OutputStream out, String name, long timecode) Send onStatus(NetStream.Play.Transition) event
int	<code>sendPauseNotify</code> (long timecode, String name) Send onStatus(NetStream.Pause.Notify) event
int	<code>sendPauseNotify</code> (java.io.OutputStream out, long timecode, String name) Send onStatus(NetStream.Pause.Notify) event.
int	<code>sendPlayReset</code> (java.io.OutputStream out, String name) Send onStatus(NetStream.Play.Reset) event.
int	<code>sendPlayReset</code> (String name) Send onStatus(NetStream.Play.Reset) event
int	<code>sendPlaySeek</code> (long location, long seekLocation, String name) Send onStatus(NetStream.Seek.Notify) event.
int	<code>sendPlaySeek</code> (java.io.OutputStream out, long location, long seekLocation, String name) Send onStatus(NetStream.Seek.Notify) event.
int	<code>sendPlaySeek</code> (java.io.OutputStream out, long location, long seekLocation, String name, java.util.List seekTypes) Send onStatus(NetStream.Seek.Notify) event.
int	<code>sendPlayStart</code> (java.io.OutputStream out, String name, boolean isSwitch, boolean isLive, long timecode, java.util.List seekTypes) Send onStatus(NetStream.Play.Start) event (it is not a typo, this also send play start but with different control codes for media switch).
int	<code>sendPlayStart</code> (java.io.OutputStream out, String name, boolean isSwitch, long timecode) Send onStatus(NetStream.Play.Start) event (it is not a typo, this also send play start but with different control codes for media switch).
int	<code>sendPlayStart</code> (java.io.OutputStream out, String name, boolean isSwitch, long timecode, java.util.List seekTypes) Send onStatus(NetStream.Play.Start) event (it is not a typo, this also send play start but with different control codes for media switch).
int	<code>sendPlayStart</code> (String name, long timecode) Send onStatus(NetStream.Play.Start) event
int	<code>sendPlayStatus</code> (long timecode, int statusType, double duration, double bytesSent) Send onPlayStatus(NetStream.Play.Switch, NetStream.Play.Complete, NetStream.Play.Stop) event
int	<code>sendPlayStatus</code> (java.io.OutputStream out, long timecode, int statusType, double duration, double bytesSent) Send onPlayStatus(NetStream.Play.Switch, NetStream.Play.Complete, NetStream.Play.Stop) event.
int	<code>sendPlayStop</code> (long location, String name) Send onStatus(NetStream.Play.Stop) event

int	<code>sendPlayStop</code> (java.io.OutputStream out, long location, String name) Send onStatus(NetStream.Play.Stop) event.
int	<code>sendPlaySwitch</code> (java.io.OutputStream out, String name, boolean isSwitch, long timecode) Send onStatus(NetStream.Play.Start) event (it is not a typo, this also send play start but with different control codes for media switch).
int	<code>sendPlaySwitch</code> (String name, long timecode) Send onStatus(NetStream.Play.Start) event (it is not a typo, this also send play start but with different control codes for media switch).
int	<code>sendStreamNotFound</code> (java.io.OutputStream out, String name) Send onStatus(NetStream.Play.StreamNotFound) event.
int	<code>sendStreamNotFound</code> (String name) Send onStatus(NetStream.Play.StreamNotFound) event
int	<code>sendUnpauseNotify</code> (long location, String name) Send onStatus(NetStream.Unpause.Notify) event
int	<code>sendUnpauseNotify</code> (java.io.OutputStream out, long location, String name) Send onStatus(NetStream.Unpause.Notify) event.
int	<code>sendUnpauseNotify</code> (java.io.OutputStream out, long location, String name, java.util.List seekTypes) Send onStatus(NetStream.Unpause.Notify) event
int	<code>sendVODPlaySwitch</code> (java.io.OutputStream out, String name, long timecode) Send onStatus(NetStream.Play.Transition) event
void	<code>setAppend</code> (boolean isAppend) Set is append to media file (only valid if isRecord)
void	<code>setAudioSize</code> (int audioSize) Set the size of the current audio packet that is being streamed from the client to the server
void	<code>setAudioTC</code> (long audioTC) Set last absolute audio timecode (milliseconds) sent to mediaStream
void	<code>setAudioTC</code> (long audioTC, boolean isAbsolute) Set last absolute audio timecode (milliseconds) sent to mediaStream
void	<code>setBufferTime</code> (int bufferTime) Set buffer time for mediaStream (milliseconds)
void	<code>setClient</code> (<code>IClient</code> client) Set parent client connection
void	<code>setClustered</code> (boolean isClustered) not used
void	<code>setDataSize</code> (int dataSize) Set the size of the current data packet that is being streamed from the client to the server
void	<code>setDataTC</code> (long dataTC) Set last absolute data timecode (milliseconds) sent to mediaStream
void	<code>setDataTC</code> (long dataTC, boolean isAbsolute) Set last absolute data timecode (milliseconds) sent to mediaStream

void	<u>setDataType</u> (int dataType) Set the data packet type: (IVHost.CONTENTTYPE_DATA0 or IVHost.CONTENTTYPE_DATA3)
void	<u>setDvrRecorder</u> (String recorderName) Set the DVR Recorder that this stream is using
void	<u>setDvrRecorderList</u> (String recorderList) Set the comma separated list of DVR Recorder names being used by this stream (see conf/Dvr.xml)
void	<u>setExt</u> (String ext) Set media file extension
void	<u>setFastPlaySettings</u> (<u>FastPlaySettings</u> fastPlaySettings) Set fastPlay settings
void	<u>setHeaderSize</u> (int headerSize) Set the last packet header size (debugging)
void	<u>setHTTPStreamerSession</u> (<u>IHTTPStreamerSession</u> httpStreamerSession) Set the HTTPStreamer session associated with this stream
void	<u>setIsPlaying</u> (boolean isPlaying) Set is mediaStream playing
void	<u>setLiveStreamPacketizer</u> (String liveStreamPacketizer) Set the live stream packetizer that this stream is using
void	<u>setLiveStreamPacketizerList</u> (String liveStreamPacketizerList) Set the comma separated list of LiveStreamPacketizers names being used by this stream (see conf/LiveStreamPacketizers.xml)
void	<u>setLiveStreamRepeater</u> (String liveStreamRepeater) Set the live stream repeater name for the stream
void	<u>setLiveStreamTranscoderList</u> (String liveStreamTranscoderList) Set the comma separated list of LiveStreamTranscoders names being used by this stream (see conf/LiveStreamTranscoders.xml)
void	<u>setMediaCasterPlay</u> (boolean isMediaCasterPlay) Is MediaCaster play enabled (if true, will trigger MediaCaster startup)
void	<u>setMergeOnMetadata</u> (boolean mergeOnMetadata) If true, merge incoming onMetadata events with the current onMetadata event data.
void	<u>setMetaDataProvider</u> (<u>IMediaStreamMetaDataProvider</u> metaDataProvider) Set the metaData provider
void	<u>setName</u> (String name) Set stream name
void	<u>setName</u> (String name, String ext) Set stream name and extension.
void	<u>setName</u> (String name, String oldName, String ext, String queryStr, double playStart, double playLen, int playTransition) Set stream name, extension, query string, play start, play len and play reset from play command.

void	<u>setNetConnection</u> (com.wowza.wms.netconnection.INetConnection netConnection) Set parent netConnection (future server to server communication)
void	<u>setOpen</u> (boolean isOpen) Set mediaStream open
void	<u>setPlay</u> (boolean isPlay) Set is the stream a play stream (vs a publish stream)
void	<u>setPlayer</u> (IMediaStreamPlay player) Set underlying player (IMediaStreamPlay) object
void	<u>setPublishAudioCodecId</u> (int publishAudioCodecId) Set the codec id of the most recently published audio packet
void	<u>setPublisherStream</u> (boolean isPublisherStream) Set stream was created by stream publisher.
void	<u>setPublishVideoCodecId</u> (int publishVideoCodecId) Set the codec id of the most recently published video packet
void	<u>setQueryStr</u> (String queryStr) Set play/publish name query string.
void	<u>setReceiveAudio</u> (boolean receiveAudio) Set receive audio
void	<u>setReceiveVideo</u> (boolean receiveVideo) Set receive video
void	<u>setReceiveVideoFPS</u> (int receiveVideoFPS) Set frame per second for video (not currently implemented)
void	<u>setRecord</u> (boolean isRecord) Set is the stream being recorded
void	<u>setRTPStream</u> (RTPStream rtpStream) Set the RTP based stream this stream is associated with
void	<u>setSendPlayStopLogEvent</u> (boolean sendPlayStopLogEvent) Set need to send a log event for stop
void	<u>setSendPublishStopLogEvent</u> (boolean sendPlayStopLogEvent) Set need to send a log event for publishing
void	<u>setSendRecordStopLogEvent</u> (boolean sendPlayStopLogEvent) Set need to send a log event for recording
void	<u>setSrc</u> (int src) Set stream id
void	<u>setStreamType</u> (String streamType) Set mediaStream streamType.
void	<u>setTranscodeResult</u> (boolean isTranscodeResult) Is this stream the result of a transcode operation.
void	<u>setVideoSize</u> (int videoSize) Set the size of the current video packet that is being streamed from the client to the server

void	<code>setVideoTC(long videoTC)</code> Set last absolute video timecode (milliseconds) sent to mediaStream
void	<code>setVideoTC(long videoTC, boolean isAbsolute)</code> Set last absolute video timecode (milliseconds) sent to mediaStream
void	<code>shutdown()</code> shutdown or close this mediaStream
long	<code>size()</code> Get size (bytes) of media file pointed to by mediaStream
void	<code>startAudioPacket()</code> Called when an audio packet is first being populated with data
void	<code>startDataPacket()</code> Called when a data packet is first being populated with data
void	<code>startPublishing()</code> Start publishing live stream
void	<code>startVideoPacket()</code> Called when a video packet is first being populated with data
void	<code>stopName(String name, String oldName, String ext, String queryStr, double playStart, double playLen, int playTransition)</code> Stop stream name
void	<code>stopPublishing()</code> Stop publishing live stream
void	<code>switchName(String name, String oldName, String ext, String queryStr, double playStart, double playLen, int playTransition)</code> Switch to stream name
void	<code>trim()</code> Trim mediaStream.
void	<code>unregisterCallback(String handlerName)</code> Unregister a callback handler
void	<code>unregisterOnPlayStatus(IMediaStreamCallback callback)</code> Unregister onPlayStatus handler
void	<code>unregisterOnStatus(IMediaStreamCallback callback)</code> Unregister onStatus handler
void	<code>updateLoggingDuration()</code> Update logging.MDC with mediaStream logging information
void	<code>updateLoggingValues()</code> Update logging.MDC with mediaStream logging information

Fields

(continued from last page)

READACCESS

```
public static final int READACCESS
```

Constant value: **0**

WRITEACCESS

```
public static final int WRITEACCESS
```

Constant value: **1**

AUDIOSAMPLEACCESS

```
public static final int AUDIOSAMPLEACCESS
```

Constant value: **2**

VIDEOSAMPLEACCESS

```
public static final int VIDEOSAMPLEACCESS
```

Constant value: **3**

Methods

init

```
public void init(MediaStreamMap parent,  
                int src,  
                WMSProperties properties)
```

Initialize the mediaStream object after creation. Creation and initialization are separated due to the order of the creation events.

Parameters:

parent - parent mediaStreamMap of the parent applicationInstance
src - stream id for this stream
properties - initial properties as defined in Streams.xml

getBufferTime

```
public int getBufferTime()
```

Get buffer time for mediaStream (milliseconds)

Returns:

buffer time (milliseconds)

setBufferTime

```
public void setBufferTime(int bufferTime)
```

Set buffer time for mediaStream (milliseconds)

(continued from last page)

Parameters:

bufferTime - buffer time (milliseconds)

isPlaying

```
public boolean isPlaying()
```

Is mediaStream playing (or paused - false)

Returns:

is playing

setIsPlaying

```
public void setIsPlaying(boolean isPlaying)
```

Set is mediaStream playing

Parameters:

isPlaying

getAudioSize

```
public int getAudioSize()
```

Get the size of the current audio packet that is being streamed from the client to the server

Returns:

size

getVideoSize

```
public int getVideoSize()
```

Get the size of the current video packet that is being streamed from the client to the server

Returns:

size

getDataSize

```
public int getDataSize()
```

Get the size of the current data packet that is being streamed from the client to the server

Returns:

size

setAudioSize

```
public void setAudioSize(int audioSize)
```

Set the size of the current audio packet that is being streamed from the client to the server

Parameters:

audioSize

(continued from last page)

setVideoSize

```
public void setVideoSize(int videoSize)
```

Set the size of the current video packet that is being streamed from the client to the server

Parameters:

videoSize

setDataSize

```
public void setDataSize(int dataSize)
```

Set the size of the current data packet that is being streamed from the client to the server

Parameters:

dataSize

addAudioExtraData

```
public void addAudioExtraData(java.util.Map extraData)
```

Add extra data to the current audio packet

Parameters:

extraData - extra data

addVideoExtraData

```
public void addVideoExtraData(java.util.Map extraData)
```

Add extra data to the current video packet

Parameters:

extraData - extra data

addDataExtraData

```
public void addDataExtraData(java.util.Map extraData)
```

Add extra data to the current data packet

Parameters:

extraData - extra data

getSrc

```
public int getSrc()
```

Get the stream id

Returns:

stream id

setSrc

```
public void setSrc(int src)
```

(continued from last page)

Set stream id

Parameters:

src - stream id

getName

```
public String getName()
```

Get stream name

Returns:

stream name

setName

```
public void setName(String name)
```

Set stream name

Parameters:

name - stream name

setName

```
public void setName(String name,  
String ext)
```

Set stream name and extension. Example: if play command sent flv:test name=test, ext=flv.

Parameters:

name - stream name

ext - stream extension

setName

```
public void setName(String name,  
String oldName,  
String ext,  
String queryStr,  
double playStart,  
double playLen,  
int playTransition)
```

Set stream name, extension, query string, play start, play len and play reset from play command.

Parameters:

name - stream name

oldName - old stream name

ext - stream extension

queryStr - query string

playStart - play start

playLen - play len

playTransition - play transition (see MediaBase.PLAYTRANSITION_*)

(continued from last page)

switchName

```
public void switchName(String name,  
    String oldName,  
    String ext,  
    String queryStr,  
    double playStart,  
    double playLen,  
    int playTransition)
```

Switch to stream name

Parameters:

name - stream name
oldName - old stream name
ext - stream extension
queryStr - query string
playStart - play start
playLen - play len
playTransition - play transition (see MediaBase.PLAYTRANSITION_*)

stopName

```
public void stopName(String name,  
    String oldName,  
    String ext,  
    String queryStr,  
    double playStart,  
    double playLen,  
    int playTransition)
```

Stop stream name

Parameters:

name - stream name
oldName - old stream name
ext - stream extension
queryStr - query string
playStart - play start
playLen - play len
playTransition - play transition (see MediaBase.PLAYTRANSITION_*)

shutdown

```
public void shutdown()
```

shutdown or close this mediaStream

getAudioTC

```
public long getAudioTC()
```

Get last absolute audio timecode (milliseconds) sent to mediaStream

Returns:

last absolute audio timecode (milliseconds)

(continued from last page)

setAudioTC

```
public void setAudioTC(long audioTC,  
    boolean isAbsolute)
```

Set last absolute audio timecode (milliseconds) sent to mediaStream

Parameters:

audioTC - timecode (milliseconds)

isAbsolute - is the timecode value relative to last timecode or absolute

setAudioTC

```
public void setAudioTC(long audioTC)
```

Set last absolute audio timecode (milliseconds) sent to mediaStream

Parameters:

audioTC - timecode (milliseconds)

getVideoTC

```
public long getVideoTC( )
```

Get last absolute video timecode (milliseconds) sent to mediaStream

Returns:

last absolute video timecode (milliseconds)

setVideoTC

```
public void setVideoTC(long videoTC,  
    boolean isAbsolute)
```

Set last absolute video timecode (milliseconds) sent to mediaStream

Parameters:

videoTC - timecode (milliseconds)

isAbsolute - is the timecode value relative to last timecode or absolute

setVideoTC

```
public void setVideoTC(long videoTC)
```

Set last absolute video timecode (milliseconds) sent to mediaStream

Parameters:

videoTC - timecode (milliseconds)

getDataTC

```
public long getDataTC( )
```

Get last absolute data timecode (milliseconds) sent to mediaStream

Returns:

last absolute data timecode (milliseconds)

(continued from last page)

setDataTC

```
public void setDataTC(long dataTC,  
    boolean isAbsolute)
```

Set last absolute data timecode (milliseconds) sent to mediaStream

Parameters:

dataTC - timecode (milliseconds)

isAbsolute - is the timecode value relative to last timecode or absolute

getDataType

```
public int getDataType()
```

Get the data packet type: (IVHost.CONTENTTYPE_DATA0 or IVHost.CONTENTTYPE_DATA3)

Returns:

data packet type

setDataType

```
public void setDataType(int dataType)
```

Set the data packet type: (IVHost.CONTENTTYPE_DATA0 or IVHost.CONTENTTYPE_DATA3)

Parameters:

dataType - data packet type

setDataTC

```
public void setDataTC(long dataTC)
```

Set last absolute data timecode (milliseconds) sent to mediaStream

Parameters:

dataTC - timecode (milliseconds)

isRecord

```
public boolean isRecord()
```

Is this stream being recorded to a file

Returns:

is stream being recorded

setRecord

```
public void setRecord(boolean isRecord)
```

Set is the stream being recorded

Parameters:

isRecord

(continued from last page)

isPlay

```
public boolean isPlay()
```

Is the stream a play stream (vs a publish stream)

Returns:

is play stream (has nothing to do with if its playing just that its play vs publish)

setPlay

```
public void setPlay(boolean isPlay)
```

Set is the stream a play stream (vs a publish stream)

Parameters:

isPlay - is play stream (has nothing to do with if its playing just that its play vs publish)

idle

```
public boolean idle()
```

Allow the mediaStream to perform idle work (not currently in use)

Returns:

true if it did some work

getClientId

```
public int getClientId()
```

Get parent client connection (id)

Returns:

parent client connection (id)

getClient

```
public IClient getClient()
```

Get parent client connection

Returns:

parent client connection

getNetConnection

```
public com.wowza.wms.netconnection.INetConnection getNetConnection()
```

Get parent netConnection (future server to server communication)

Returns:

parent netConnection connection

setNetConnection

```
public void setNetConnection(com.wowza.wms.netconnection.INetConnection netConnection)
```

(continued from last page)

Set parent netConnection (future server to server communication)

Parameters:

netConnection - netConnection connection

setClient

```
public void setClient(IClient client)
```

Set parent client connection

Parameters:

client - parent client connection

getStreams

```
public MediaStreamMap getStreams()
```

Get parent mediaStreamMap (owned by applicationInstance)

Returns:

parent mediaStreamMap

packetComplete

```
public void packetComplete()
```

Invoked by requestAdapter when at the end of a set of packets

sendStreamNotFound

```
public int sendStreamNotFound(String name)
```

Send onStatus(NetStream.Play.StreamNotFound) event

Parameters:

name - stream name

Returns:

bytes sent to client

sendStreamNotFound

```
public int sendStreamNotFound(java.io.OutputStream out,  
    String name)
```

Send onStatus(NetStream.Play.StreamNotFound) event. Send directly to OutputStream.

Parameters:

out - OutputStream

name - stream name

Returns:

bytes sent to client

(continued from last page)

sendLivePlayStart

```
public int sendLivePlayStart( java.io.OutputStream out,  
    String name,  
    long timecode,  
    long timecodeOffset)
```

Send onStatus(NetStream.Play.Start) event

Parameters:

out - OutputStream
name - stream name
timecode - timecode
timecodeOffset - timecode offset

Returns:

bytes sent to client

sendVODPlaySwitch

```
public int sendVODPlaySwitch( java.io.OutputStream out,  
    String name,  
    long timecode)
```

Send onStatus(NetStream.Play.Transition) event

Parameters:

out - OutputStream
name - stream name
timecode - timecode

Returns:

bytes sent to client

sendLivePlaySwitch

```
public int sendLivePlaySwitch( java.io.OutputStream out,  
    String name,  
    long timecode)
```

Send onStatus(NetStream.Play.Transition) event

Parameters:

out - OutputStream
name - stream name
timecode - timecode

Returns:

bytes sent to client

sendLivePlaySeek

```
public int sendLivePlaySeek( java.io.OutputStream out,  
    String name,  
    long timecode)
```

Send onStatus(NetStream.Seek.Notify) event

Parameters:

out - OutputStream

(continued from last page)

name - stream name
timecode - timecode

Returns:

bytes sent to client

sendPlayReset

```
public int sendPlayReset(String name)
```

Send onStatus(NetStream.Play.Reset) event

Parameters:

name - stream name

Returns:

bytes sent to client

sendPlayReset

```
public int sendPlayReset(java.io.OutputStream out,  
    String name)
```

Send onStatus(NetStream.Play.Reset) event. Send directly to OutputStream.

Parameters:

out - OutputStream
name - stream name

Returns:

bytes sent to client

sendPlayStop

```
public int sendPlayStop(long location,  
    String name)
```

Send onStatus(NetStream.Play.Stop) event

Parameters:

location - timecode where play stopped
name - stream name

Returns:

bytes sent to client

sendPlayStop

```
public int sendPlayStop(java.io.OutputStream out,  
    long location,  
    String name)
```

Send onStatus(NetStream.Play.Stop) event. Send directly to OutputStream

Parameters:

out - OutputStream
location - timecode where play stopped
name - stream name

(continued from last page)

Returns:

bytes sent to client

sendPlaySeek

```
public int sendPlaySeek(java.io.OutputStream out,  
    long location,  
    long seekLocation,  
    String name,  
    java.util.List seekTypes)
```

Send onStatus(NetStream.Seek.Notify) event.

Parameters:

out - OutputStream
location - request location of seek
seekLocation - result location of seek
name - stream name
seekTypes - list of commands to respond to (seek, unpause, play)

Returns:

bytes sent to client

sendPlaySeek

```
public int sendPlaySeek(long location,  
    long seekLocation,  
    String name)
```

Send onStatus(NetStream.Seek.Notify) event.

Parameters:

location - request location of seek
seekLocation - result location of seek
name - stream name

Returns:

bytes sent to client

sendPlaySeek

```
public int sendPlaySeek(java.io.OutputStream out,  
    long location,  
    long seekLocation,  
    String name)
```

Send onStatus(NetStream.Seek.Notify) event. Send directly to OutputStream

Parameters:

out - OutputStream
location - request location of seek
seekLocation - result location of seek
name - stream name

Returns:

bytes sent to client

(continued from last page)

sendPlayStart

```
public int sendPlayStart(String name,  
    long timecode)
```

Send onStatus(NetStream.Play.Start) event

Parameters:

name - stream name

Returns:

bytes sent to client

sendPlaySwitch

```
public int sendPlaySwitch(String name,  
    long timecode)
```

Send onStatus(NetStream.Play.Start) event (it is not a typo, this also send play start but with different control codes for media switch).

Parameters:

name - stream name

timecode - timecode of event

Returns:

bytes sent to client

sendPlaySwitch

```
public int sendPlaySwitch(java.io.OutputStream out,  
    String name,  
    boolean isSwitch,  
    long timecode)
```

Send onStatus(NetStream.Play.Start) event (it is not a typo, this also send play start but with different control codes for media switch). Send directly to OutputStream

Parameters:

out - OutputStream

name - stream name

isSwitch - is this a switch or a start

timecode - timecode of event

Returns:

bytes sent to client

sendPlayStart

```
public int sendPlayStart(java.io.OutputStream out,  
    String name,  
    boolean isSwitch,  
    boolean isLive,  
    long timecode,  
    java.util.List seekTypes)
```

Send onStatus(NetStream.Play.Start) event (it is not a typo, this also send play start but with different control codes for media switch). Send directly to OutputStream

Parameters:

out - OutputStream

(continued from last page)

name - stream name
isSwitch - is this a switch or a start
isLive - is the stream live
timecode - timecode of event
seekTypes - commands to respond to (seek, unpause, play)

sendPlayStart

```
public int sendPlayStart( java.io.OutputStream out,  
    String name,  
    boolean isSwitch,  
    long timecode,  
    java.util.List seekTypes)
```

Send onStatus(NetStream.Play.Start) event (it is not a typo, this also send play start but with different control codes for media switch). Send directly to OutputStream

Parameters:

out - OutputStream
name - stream name
isSwitch - is this a switch or a start
timecode - timecode of event
seekTypes - commands to respond to (seek, unpause, play)

Returns:

bytes sent to client

sendPlayStart

```
public int sendPlayStart( java.io.OutputStream out,  
    String name,  
    boolean isSwitch,  
    long timecode)
```

Send onStatus(NetStream.Play.Start) event (it is not a typo, this also send play start but with different control codes for media switch). Send directly to OutputStream

Parameters:

out - OutputStream
name - stream name
isSwitch - is this a switch or a start
timecode - timecode of event

Returns:

bytes sent to client

sendPlayStatus

```
public int sendPlayStatus(long timecode,  
    int statusType,  
    double duration,  
    double bytesSent)
```

Send onPlayStatus(NetStream.Play.Switch, NetStream.Play.Complete, NetStream.Play.Stop) event

Parameters:

timecode - timecode of event
statusType - status type IMediaStreamPlay.PLAYSTATUSTYPE_*
duration - (not used)
bytesSent - (not used)

(continued from last page)

Returns:

bytes sent to client

sendPlayStatus

```
public int sendPlayStatus(java.io.OutputStream out,  
    long timecode,  
    int statusType,  
    double duration,  
    double bytesSent)
```

Send onPlayStatus(NetStream.Play.Switch, NetStream.Play.Complete, NetStream.Play.Stop) event. Send directly to OutputStream.

Parameters:

out - OutputStream
timecode - timecode of event
statusType - status type IMediaStreamPlay.PLAYSTATUSTYPE_*
duration - (not used)
bytesSent - (not used)

Returns:

bytes sent to client

sendPauseNotify

```
public int sendPauseNotify(long timecode,  
    String name)
```

Send onStatus(NetStream.Pause.Notify) event

Parameters:

timecode - timecode of event
name - stream name

Returns:

bytes sent to client

sendPauseNotify

```
public int sendPauseNotify(java.io.OutputStream out,  
    long timecode,  
    String name)
```

Send onStatus(NetStream.Pause.Notify) event. Send directly to OutputStream.

Parameters:

out - OutputStream
timecode - timecode of event
name - stream name

Returns:

bytes sent to client

sendUnpauseNotify

```
public int sendUnpauseNotify(java.io.OutputStream out,  
    long location,  
    String name,  
    java.util.List seekTypes)
```

(continued from last page)

Send onStatus(NetStream.Unpause.Notify) event

Parameters:

out - OutputStream
location - timecode of event
name - stream name
seekTypes - list of operations to respond to (seek, play, unpause)

Returns:

bytes sent to client

sendUnpauseNotify

```
public int sendUnpauseNotify(long location,  
    String name)
```

Send onStatus(NetStream.Unpause.Notify) event

Parameters:

location - timecode of event
name - stream name

Returns:

bytes sent to client

sendUnpauseNotify

```
public int sendUnpauseNotify(java.io.OutputStream out,  
    long location,  
    String name)
```

Send onStatus(NetStream.Unpause.Notify) event. Send directly to OutputStream.

Parameters:

out - OutputStream
location - timecode of event
name - stream name

Returns:

bytes sent to client

isAppend

```
public boolean isAppend( )
```

Is append to media file (only valid if isRecord)

Returns:

is appending or rewriting media file

setAppend

```
public void setAppend(boolean isAppend)
```

Set is append to media file (only valid if isRecord)

Parameters:

isAppend - is appending or rewriting media file

setPlayer

```
public void setPlayer(IMediaStreamPlay player)
```

Set underlying player (IMediaStreamPlay) object

Parameters:

player - underlying player (IMediaStreamPlay) object

getPlayer

```
public IMediaStreamPlay getPlayer()
```

Get underlying player (IMediaStreamPlay) object

Returns:

underlying player (IMediaStreamPlay) object

getAudioMissing

```
public int getAudioMissing()
```

Get number of audio bytes missing from current audio packet

Returns:

number of bytes missing

getVideoMissing

```
public int getVideoMissing()
```

Get number of video bytes missing from current audio packet

Returns:

number of bytes missing

getDataMissing

```
public int getDataMissing()
```

Get number of data bytes missing from current audio packet

Returns:

number of bytes missing

addVideoData

```
public void addVideoData(byte[] data,  
    int offset,  
    int size)
```

Add data to current video packet

Parameters:

data - byte array

offset - offset in byte array

size - size of data to add

addAudioData

```
public void addAudioData(byte[] data,  
    int offset,  
    int size)
```

Add data to current audio packet

Parameters:

data - byte array
offset - offset in byte array
size - size of data to add

addDataData

```
public void addDataData(byte[] data,  
    int offset,  
    int size)
```

Add data to current data packet

Parameters:

data - byte array
offset - offset in byte array
size - size of data to add

getProperties

```
public WMSProperties getProperties()
```

Get mediaStream properties

Returns:

properties

getPlayPackets

```
public java.util.List getPlayPackets()
```

Get all available live packets

Returns:

play packets

getMaxTimecode

```
public long getMaxTimecode()
```

Get the timecode of the latest received packet

Returns:

timecode of the latest received packet

getLastKeyFrame

```
public AMFPacket getLastKeyFrame()
```

Get most recent video key frame

(continued from last page)

Returns:

play packets

getLastPacket

```
public AMFPacket getLastPacket()
```

Get most recent live packet

Returns:

most recent live packet or null or no live packets

getStreamType

```
public String getStreamType()
```

Get mediaStream streamType

Returns:

streamType

setStreamType

```
public void setStreamType(String streamType)
```

Set mediaStream streamType. This method will not change the type of the current stream.

Parameters:

streamType

sendDirectAMF3

```
public void sendDirectAMF3(String handlerName,  
    Object[] params)
```

Call client side NetStream method/handler and send event to underlying stream (will record event). Force AMF3 encoding.

Parameters:

handlerName - handler name

params - variable list of parameters. All parameters will be wrapped in AMFDataObj.

sendDirect

```
public void sendDirect(String handlerName,  
    Object[] params)
```

Call client side NetStream method/handler and send event to underlying stream (will record event)

Parameters:

handlerName - handler name

params - variable list of parameters. All parameters will be wrapped in AMFDataObj.

sendDirectAMF3

```
public void sendDirectAMF3(String handlerName)
```

Call client side NetStream method/handler and send event to underlying stream (will record event). Force AMF3 encoding.

(continued from last page)

Parameters:

handlerName - handler name

sendDirect

```
public void sendDirect(String handlerName)
```

Call client side NetStream method/handler and send event to underlying stream (will record event)

Parameters:

handlerName - handler name

sendAMF3

```
public void sendAMF3(String handlerName,  
    Object[] params)
```

Call client side NetStream method/handler. Force AMF3 encoding.

Parameters:

handlerName - handler name

params - variable list of parameters. All parameters will be wrapped in AMFDataObj.

send

```
public void send(String handlerName,  
    Object[] params)
```

Call client side NetStream method/handler

Parameters:

handlerName - handler name

params - variable list of parameters. All parameters will be wrapped in AMFDataObj.

sendAMF3

```
public void sendAMF3(String handlerName)
```

Call client side NetStream method/handler with no parameters. Force AMF3 encoding.

Parameters:

handlerName - handler name

send

```
public void send(String handlerName)
```

Call client side NetStream method/handler with no parameters

Parameters:

handlerName - handler name

close

```
public void close()
```

Close mediaStream

(continued from last page)

isOpen

```
public boolean isOpen()
```

Is mediaStream open

Returns:

is mediaStream open

setOpen

```
public void setOpen(boolean isOpen)
```

Set mediaStream open

Parameters:

isOpen - mediaStream open

getFastPlaySettings

```
public FastPlaySettings getFastPlaySettings()
```

Get current fastPlay settings

Returns:

fastPlay settings

setFastPlaySettings

```
public void setFastPlaySettings(FastPlaySettings fastPlaySettings)
```

Set fastPlay settings

Parameters:

fastPlaySettings - fastPlay settings

clearFastPlaySettings

```
public void clearFastPlaySettings()
```

Clear fastPlay settings

isReceiveAudio

```
public boolean isReceiveAudio()
```

Is client currently receiving audio. Controlled by client side call receiveAudio.

Returns:

receive audio

setReceiveAudio

```
public void setReceiveAudio(boolean receiveAudio)
```

Set receive audio

Parameters:

(continued from last page)

`receiveAudio` - receive audio

`isReceiveVideo`

```
public boolean isReceiveVideo()
```

Is client currently receiving video. Controlled by client side call `receiveVideo`

Returns:

receive video

`setReceiveVideo`

```
public void setReceiveVideo(boolean receiveVideo)
```

Set receive video

Parameters:

receiveVideo - receive video

`getReceiveVideoFPS`

```
public int getReceiveVideoFPS()
```

Set frame per seconds for video (not currently implemented)

Returns:

video frames per second

`setReceiveVideoFPS`

```
public void setReceiveVideoFPS(int receiveVideoFPS)
```

Set frame per second for video (not currently implemented)

Parameters:

receiveVideoFPS - video frames per second

`getMediaIOPerformance`

```
public IOPerformanceCounter getMediaIOPerformance()
```

Get IO performance counter

Returns:

IO performance counter

`incrementMediaOutBytes`

```
public long incrementMediaOutBytes(long bytes,  
    long count)
```

Increment the number of mediaStream bytes sent and number of packets sent

Parameters:

bytes - number of bytes sent
count - number of packets sent

Returns:

(continued from last page)

total number of bytes sent (after increment)

incrementMediaLossBytes

```
public long incrementMediaLossBytes(long bytes,  
                                     long count)
```

Increment the number of mediaStream loss bytes sent and number of packets sent

Parameters:

bytes - number of bytes sent
count - number of packets sent

Returns:

total number of bytes sent (after increment)

incrementMediaInBytes

```
public long incrementMediaInBytes(long increment)
```

Increment the number of mediaStream bytes received

Parameters:

increment - number of byte received

Returns:

total number of bytes received (after increment)

publish

```
public void publish()
```

Publish mediaStream

trim

```
public void trim()
```

Trim mediaStream. This method will remove live packets that are older than the live buffer size.

handleCallback

```
public void handleCallback(com.wowza.wms.request.RequestFunction function)
```

Routes request function to callback handler onStatus, onPlayStatus or [method/handler]

Parameters:

function - request function

unregisterCallback

```
public void unregisterCallback(String handlerName)
```

Unregister a callback handler

Parameters:

handlerName - handler name

registerCallback

```
public void registerCallback(String handlerName,  
    IMediaStreamCallback callback)
```

Register a callback handler

Parameters:

handlerName - handler name
callback - callback object

unregisterOnStatus

```
public void unregisterOnStatus(IMediaStreamCallback callback)
```

Unregister onStatus handler

Parameters:

callback

registerOnStatus

```
public void registerOnStatus(IMediaStreamCallback callback)
```

Register onStatus handler

Parameters:

callback - callback object

unregisterOnPlayStatus

```
public void unregisterOnPlayStatus(IMediaStreamCallback callback)
```

Unregister onPlayStatus handler

Parameters:

callback - callback object

registerOnPlayStatus

```
public void registerOnPlayStatus(IMediaStreamCallback callback)
```

Register onPlayStatus handler

Parameters:

callback - callback object

addClientListener

```
public void addClientListener(IMediaStreamActionNotify actionListener)
```

Add client listener. Listens for (onPlay, onPublish, onPause, onSeek, onStop)

Parameters:

actionListener - listener

(continued from last page)

addClientListener

```
public void addClientListener(IMediaStreamActionNotify2 actionListener)
```

Add client listener. Listens for (onPlay, onPublish, onPause, onSeek, onStop)

Parameters:

actionListener - listener

addClientListener

```
public void addClientListener(IMediaStreamActionNotify3 actionListener)
```

Add client listener. Listens for (onPlay, onPublish, onPause, onSeek, onStop)

Parameters:

actionListener - listener

removeClientListener

```
public void removeClientListener(IMediaStreamActionNotify actionListener)
```

Remove client listener. Listens for (onPlay, onPublish, onPause, onSeek, onStop)

Parameters:

actionListener - listener

removeClientListener

```
public void removeClientListener(IMediaStreamActionNotify2 actionListener)
```

Remove client listener. Listens for (onPlay, onPublish, onPause, onSeek, onStop)

Parameters:

actionListener - listener

removeClientListener

```
public void removeClientListener(IMediaStreamActionNotify3 actionListener)
```

Remove client listener. Listens for (onPlay, onPublish, onPause, onSeek, onStop)

Parameters:

actionListener - listener

notifyActionPlay

```
public void notifyActionPlay(String streamName,  
    double playStart,  
    double playLen,  
    int playReset)
```

Notify client listeners of play action

Parameters:

streamName - stream name

playStart - play start

playLen - play length

playReset - play reset

notifyActionPauseRaw

```
public void notifyActionPauseRaw(boolean isPause,  
    long location)
```

Notify client listeners of pauseRaw action

Parameters:

isPause - is pause or unpause
location - timecode (milliseconds) of action

notifyActionPause

```
public void notifyActionPause(boolean isPause,  
    long location)
```

Notify client listeners of pause action

Parameters:

isPause - is pause or unpause
location - timecode (milliseconds) of action

notifyActionSeek

```
public void notifyActionSeek(double location)
```

Notify client listeners of seek action

Parameters:

location - timecode (milliseconds) of seek request

notifyActionPublish

```
public void notifyActionPublish(String streamName,  
    boolean isRecord,  
    boolean isAppend)
```

Notify client listeners of publish action

Parameters:

streamName - stream name
isRecord - is record or live
isAppend - is append if isRecord is true

notifyActionUnPublish

```
public void notifyActionUnPublish(String streamName,  
    boolean isRecord,  
    boolean isAppend)
```

Notify client listeners of unpublish action

Parameters:

streamName - stream name
isRecord - is record or live
isAppend - is append if isRecord is true

(continued from last page)

notifyActionOnMetaData

```
public void notifyActionOnMetaData(AMFPacket metaDataPacket)
```

Notify client listeners of onMetaData change

Parameters:

metaDataPacket - metaDataPacket

notifyActionOnCodecInfoVideo

```
public void notifyActionOnCodecInfoVideo(com.wowza.wms.media.model.MediaCodecInfoVideo  
codecInfoVideo)
```

Notify client listeners of video codec information change

Parameters:

codecInfoVideo - video codec information

notifyActionOnCodecInfoAudio

```
public void notifyActionOnCodecInfoAudio(com.wowza.wms.media.model.MediaCodecInfoAudio  
codecInfoAudio)
```

Notify client listeners of audio codec information change

Parameters:

codecInfoAudio - audio codec information

notifyActionStop

```
public void notifyActionStop()
```

Notify client listeners of stop action

isClustered

```
public boolean isClustered()
```

not used

Returns:

isClustered

setClustered

```
public void setClustered(boolean isClustered)
```

not used

Parameters:

isClustered

getCacheName

```
public String getCacheName()
```

not used

(continued from last page)

Returns:

cache name

startPublishing

```
public void startPublishing()
```

Start publishing live stream

stopPublishing

```
public void stopPublishing()
```

Stop publishing live stream

getStreamFileForWrite

```
public java.io.File getStreamFileForWrite()
```

Get the File object to write to a stream (get stream name, ext and query from stream object)

Returns:

resultant File object

getStreamFileForWrite

```
public java.io.File getStreamFileForWrite(String name,  
                                           String ext,  
                                           String query)
```

Get the File object to write to a stream (specify name, ext and query)

Parameters:

name - stream name

ext - stream prefix (Ex. mp4:)

query - query part of stream name (Ex. mystream?param1=value1)

Returns:

resultant File object

getStreamFileForRead

```
public java.io.File getStreamFileForRead()
```

Get the File object to read from a stream (get stream name, ext and query from stream object)

Returns:

resultant File object

getStreamFileForRead

```
public java.io.File getStreamFileForRead(String name,  
                                           String ext,  
                                           String query)
```

Get the File object to read from a stream (specify name, ext and query)

Parameters:

name - stream name

(continued from last page)

ext - stream prefix (Ex. mp4:)

query - query part of stream name (Ex. mystream?param1=value1)

Returns:

resultant File object

sendControlBytes

```
public int sendControlBytes(int controlType,  
    java.io.OutputStream out)
```

Send playback control bytes. Valid values are (0, 1, 4)

Parameters:

controlType - control types (0, 1, 4)

out - OutputStream

Returns:

bytes sent to client

getBurstStartStop

```
public byte[] getBurstStartStop(boolean isStart)
```

Get the dynamic streaming burst start/stop AMF packet

Parameters:

isStart - is start

Returns:

byte array with AMF packet

getRespAMFAudioObj

```
public AMFObj getRespAMFAudioObj()
```

Get audio response channel object

Returns:

audio response channel object

getRespAMFVideoObj

```
public AMFObj getRespAMFVideoObj()
```

Get video response channel object

Returns:

video response channel object

getRespAMFDataObj

```
public AMFObj getRespAMFDataObj()
```

Get data response channel object

Returns:

data response channel object

getQueryStr

```
public String getQueryStr()
```

Get play/publish name query string. Example: if play name is flv:test?param1=data1¶m2=data2, query string is "param1=data1¶m2=data2".

Returns:

query string

setQueryStr

```
public void setQueryStr(String queryStr)
```

Set play/publish name query string. Example: if play name is flv:test?param1=data1¶m2=data2, query string is "param1=data1¶m2=data2".

Parameters:

queryStr

updateLoggingDuration

```
public void updateLoggingDuration()
```

Update logging.MDC with mediaStream logging information

updateLoggingValues

```
public void updateLoggingValues()
```

Update logging.MDC with mediaStream logging information

clearLoggingValues

```
public void clearLoggingValues()
```

length

```
public double length()
```

Get length/duration (seconds) of media file pointed to by mediaStream

Returns:

length (seconds)

size

```
public long size()
```

Get size (bytes) of media file pointed to by mediaStream

Returns:

size (bytes)

(continued from last page)

getExt

```
public String getExt()
```

Get media file extension

Returns:

media file extension

setExt

```
public void setExt(String ext)
```

Set media file extension

Parameters:

ext

clear

```
public void clear()
```

Delete media file pointed to by this mediaStream (be careful)

isSendPlayStopLogEvent

```
public boolean isSendPlayStopLogEvent()
```

Get need to send a log event for stop

Returns:

need to send a log event for stop

setSendPlayStopLogEvent

```
public void setSendPlayStopLogEvent(boolean sendPlayStopLogEvent)
```

Set need to send a log event for stop

Parameters:

sendPlayStopLogEvent - need to send a log event for stop

isSendRecordStopLogEvent

```
public boolean isSendRecordStopLogEvent()
```

Get need to send a log event for recording

Returns:

need to send a log event for stop

setSendRecordStopLogEvent

```
public void setSendRecordStopLogEvent(boolean sendPlayStopLogEvent)
```

Set need to send a log event for recording

Parameters:

(continued from last page)

sendPlayStopLogEvent - need to send a log event for stop

isSendPublishStopLogEvent

```
public boolean isSendPublishStopLogEvent()
```

Get need to send a log event for publishing

Returns:

need to send a log event for stop

setSendPublishStopLogEvent

```
public void setSendPublishStopLogEvent(boolean sendPlayStopLogEvent)
```

Set need to send a log event for publishing

Parameters:

sendPlayStopLogEvent - need to send a log event for stop

getAccess

```
public boolean[] getAccess(IClient client,  
    String name)
```

Get the read/write access to this stream for this client

Parameters:

client - client

name - stream name

Returns:

array of booleans read[0], write[1]

getMetaDataProvider

```
public IMediaStreamMetaDataProvider getMetaDataProvider()
```

Get the metaData provider

Returns:

metaData provider

setMetaDataProvider

```
public void setMetaDataProvider(IMediaStreamMetaDataProvider metaDataProvider)
```

Set the metaData provider

Parameters:

metaDataProvider - metaData provider

getHeaderSize

```
public int getHeaderSize()
```

Get the last packet header size (debugging)

Returns:

(continued from last page)

last packet header size

setHeaderSize

```
public void setHeaderSize(int headerSize)
```

Set the last packet header size (debugging)

Parameters:

headerSize - last packet header size

getAudioCodecConfigPacket

```
public AMFPacket getAudioCodecConfigPacket(long timecode)
```

Get audio codec configuration packet (needed for H.264/AAC playback)

Parameters:

timecode - timecode of the packet to which you want to get the codec config information

Returns:

audio codec configuration packet (needed for H.264/AAC playback)

addAudioCodecConfigPacket

```
public void addAudioCodecConfigPacket(long timecode,  
    AMFPacket packet)
```

Set audio codec configuration packet (needed for H.264/AAC playback)

Parameters:

timecode - timecode in milliseconds of first packet that uses this timecode

packet - audio codec configuration packet (needed for H.264/AAC playback)

getVideoCodecConfigPacket

```
public AMFPacket getVideoCodecConfigPacket(long timecode)
```

Get video codec configuration packet (needed for H.264/AAC playback)

Parameters:

timecode - timecode of the packet to which you want to get the codec config information

Returns:

video codec configuration packet (needed for H.264/AAC playback)

addVideoCodecConfigPacket

```
public void addVideoCodecConfigPacket(long timecode,  
    AMFPacket packet)
```

Set video codec configuration packet (needed for H.264/AAC playback)

Parameters:

timecode - timecode in milliseconds of first packet that uses this timecode

packet - video codec configuration packet (needed for H.264/AAC playback)

(continued from last page)

getRTPStream

```
public RTPStream getRTPStream( )
```

Get the RTP based stream this stream is associated with

Returns:

RTP based stream this stream is associated with

setRTPStream

```
public void setRTPStream(RTPStream rtpStream)
```

Set the RTP based stream this stream is associated with

Parameters:

rtpStream - RTP based stream this stream is associated with

flush

```
public void flush( )
```

Force publishing packets to be flushed from the input buffers to the output buffers

startAudioPacket

```
public void startAudioPacket( )
```

Called when an audio packet is first being populated with data

startVideoPacket

```
public void startVideoPacket( )
```

Called when a video packet is first being populated with data

startDataPacket

```
public void startDataPacket( )
```

Called when a data packet is first being populated with data

getLiveStreamPacketizerList

```
public String getLiveStreamPacketizerList( )
```

Get the comma separated list of LiveStreamPacketizers names being used by this stream (see conf/LiveStreamPacketizers.xml)

Returns:

comma separated list of LiveStreamPacketizers names

setLiveStreamPacketizerList

```
public void setLiveStreamPacketizerList(String liveStreamPacketizerList)
```

Set the comma separated list of LiveStreamPacketizers names being used by this stream (see conf/LiveStreamPacketizers.xml)

Parameters:

(continued from last page)

liveStreamPacketizerList - comma separated list of LiveStreamPacketizers names

getLiveStreamTranscoderList

```
public String getLiveStreamTranscoderList()
```

Get the comma separated list of LiveStreamTranscoders names being used by this stream (see conf/LiveStreamTranscoders.xml)

Returns:

comma separated list of LiveStreamTranscoders names

setLiveStreamTranscoderList

```
public void setLiveStreamTranscoderList(String liveStreamTranscoderList)
```

Set the comma separated list of LiveStreamTranscoders names being used by this stream (see conf/LiveStreamTranscoders.xml)

Parameters:

liveStreamTranscoderList - comma separated list of LiveStreamTranscoders names

getLiveStreamPacketizer

```
public ILiveStreamPacketizer getLiveStreamPacketizer(String name)
```

Get the LiveStreamPacketizer interface to a stream by name

Parameters:

name - LiveStreamPacketizer name

Returns:

LiveStreamPacketizer interface

getDvrRecorderList

```
public String getDvrRecorderList()
```

Get the comma separated list of DVR Recorder names being used by this stream (see conf/Dvr.xml)

Returns:

comma separated list of DVR Recorder names

setDvrRecorderList

```
public void setDvrRecorderList(String recorderList)
```

Set the comma separated list of DVR Recorder names being used by this stream (see conf/Dvr.xml)

Parameters:

recorderList - comma separated list of DVR Recorder names

getDvrRecorder

```
public ILiveStreamDvrRecorder getDvrRecorder(String name)
```

Get the DVR Recorder interface to a stream by name

Parameters:

(continued from last page)

name - DVR Recorder name

Returns:

DVR Recorder interface

getUniqueStreamIdStr

```
public String getUniqueStreamIdStr()
```

Get a string that uniquely identifies this stream

Returns:

unique stream identifier

getHTTPStreamerSession

```
public IHTTPStreamerSession getHTTPStreamerSession()
```

Get the HTTPStreamer session associated with this stream

Returns:

HTTPStreamer session associated with this stream

setHTTPStreamerSession

```
public void setHTTPStreamerSession(IHTTPStreamerSession httpStreamerSession)
```

Set the HTTPStreamer session associated with this stream

Parameters:

httpStreamerSession - HTTPStreamer session associated with this stream

getElapsedTime

```
public ElapsedTimer getElapsedTime()
```

Get the interface to the elapse timer

Returns:

interface to the elapse timer

getLiveStreamPacketizer

```
public String getLiveStreamPacketizer()
```

Get the live stream packetizer that this stream is using

Returns:

live stream packetizer

setLiveStreamPacketizer

```
public void setLiveStreamPacketizer(String liveStreamPacketizer)
```

Set the live stream packetizer that this stream is using

Parameters:

liveStreamPacketizer - live stream packetizer

getLiveStreamRepeater

```
public String getLiveStreamRepeater()
```

Get the live stream repeater name for the stream

Returns:

live stream repeater name

setLiveStreamRepeater

```
public void setLiveStreamRepeater(String liveStreamRepeater)
```

Set the live stream repeater name for the stream

Parameters:

liveStreamRepeater - live stream repeater name

initLiveStreamRepeating

```
public void initLiveStreamRepeating(String liveStreamPacketizer,  
    String liveStreamRepeater)
```

Initialize this stream for live stream repeating

Parameters:

liveStreamPacketizer - live stream packetizer

liveStreamRepeater - live stream repeater name

getPublishVideoCodecId

```
public int getPublishVideoCodecId()
```

Get the codec id of the most recently published video packet

Returns:

codec id of the most recently published video packet

setPublishVideoCodecId

```
public void setPublishVideoCodecId(int publishVideoCodecId)
```

Set the codec id of the most recently published video packet

Parameters:

publishVideoCodecId - codec id of the most recently published video packet

getPublishAudioCodecId

```
public int getPublishAudioCodecId()
```

Get the codec id of the most recently published audio packet

Returns:

codec id of the most recently published audio packet

(continued from last page)

setPublishAudioCodecId

```
public void setPublishAudioCodecId(int publishAudioCodecId)
```

Set the codec id of the most recently published audio packet

Parameters:

publishAudioCodecId - codec id of the most recently published audio packet

isPublishStreamReady

```
public boolean isPublishStreamReady(boolean checkAudio,  
    boolean checkVideo)
```

Returns true if the publishing stream contains enough video/audio data to start playback

Parameters:

checkAudio - check audio stream

checkVideo - check video stream

Returns:

true if the publishing stream contains enough video/audio data to start playback

getContextStr

```
public String getContextStr()
```

Returns the stream context string in the form [application]/[appInstance]/[streamName].

Returns:

stream context string

isMediaCasterPlay

```
public boolean isMediaCasterPlay()
```

Is MediaCaster play enabled (if true, will trigger MediaCaster startup)

Returns:

true if MediaCaster play enabled

setMediaCasterPlay

```
public void setMediaCasterPlay(boolean isMediaCasterPlay)
```

Is MediaCaster play enabled (if true, will trigger MediaCaster startup)

Parameters:

isMediaCasterPlay - true if MediaCaster play enabled

isMergeOnMetadata

```
public boolean isMergeOnMetadata()
```

If true, merge incoming onMetadata events with the current onMetadata event data. If false, replace.

Returns:

true, merge incoming onMetadata events with the current onMetadata event data. If false, replace.

setMergeOnMetadata

```
public void setMergeOnMetadata(boolean mergeOnMetadata)
```

If true, merge incoming onMetadata events with the current onMetadata event data. If false, replace.

Parameters:

mergeOnMetadata - true, merge incoming onMetadata events with the current onMetadata event data. If false, replace.

getDvrRecorder

```
public String getDvrRecorder()
```

Get the DVR Recorder for this stream

Returns:

DVR Recorder

getDvrRepeater

```
public String getDvrRepeater()
```

Get the DVR repeater name for this stream

Returns:

repeater name

setDvrRecorder

```
public void setDvrRecorder(String recorderName)
```

Set the DVR Recorder that this stream is using

Parameters:

recorderName - DVR Recorder

removeDvrRecorder

```
public ILiveStreamDvrRecorder removeDvrRecorder(String name)
```

Remove a live stream dvr by name

Parameters:

name - dvr name

Returns:

live stream dvr

putDvrRecorder

```
public void putDvrRecorder(String name,  
    ILiveStreamDvrRecorder dvr)
```

Add a live stream dvr to this stream

Parameters:

name - dvr name

dvr - live stream dvr

getLiveStreamTranscoders

```
public java.util.Map getLiveStreamTranscoders()
```

Get the list of transcoders for this stream.

Returns:

list of transcoders for this stream

getLiveStreamTranscoder

```
public ILiveStreamTranscoder getLiveStreamTranscoder(String name)
```

Get a live stream transcoder for this stream by name

Parameters:

name - transcoder name

Returns:

live stream transcoder

removeLiveStreamTranscoder

```
public ILiveStreamTranscoder removeLiveStreamTranscoder(String name)
```

Remove a live stream transcoder by name

Parameters:

name - transcoder name

Returns:

live stream transcoder

putLiveStreamTranscoder

```
public void putLiveStreamTranscoder(String name,  
    ILiveStreamTranscoder liveStreamTranscoder)
```

Add a live stream transcoder to this stream

Parameters:

name - transcoder name

liveStreamTranscoder - live stream transcoder

isTranscodeResult

```
public boolean isTranscodeResult()
```

Is this stream the result of a transcode operation.

Returns:

true if stream the result of a transcode operation

setTranscodeResult

```
public void setTranscodeResult(boolean isTranscodeResult)
```

Is this stream the result of a transcode operation.

(continued from last page)

Parameters:

isTranscodeResult - true if stream the result of a transcode operation

addVideoH264SEIListener

```
public void addVideoH264SEIListener(IMediaStreamH264SEINotify h264SEIListener)
```

Add an H.264 SEI listener. This listener will be notified of all incoming H.264 video packets and has the ability to read and/or modify SEI section of the video frame.

Parameters:

h264SEIListener - H.264 SEI listener

removeVideoH264SEIListener

```
public void removeVideoH264SEIListener(IMediaStreamH264SEINotify h264SEIListener)
```

Remove an H.264 SEI listener. This listener will be notified of all incoming H.264 video packets and has the ability to read and/or modify SEI section of the video frame.

Parameters:

h264SEIListener - H.264 SEI listener

isVideoH264SEIListenerEmpty

```
public boolean isVideoH264SEIListenerEmpty()
```

Is H.264 SEI listener list empty. This listener will be notified of all incoming H.264 video packets and has the ability to read and/or modify SEI section of the video frame.

Returns:

true if H.264 SEI listener list empty

notifyVideoH264Packet

```
public void notifyVideoH264Packet(AMFPacket packet,  
com.wowza.wms.media.h264.H264SEIMessages seiMessages)
```

Notify H.264 SEI listener.

Parameters:

packet - AMF Packet

seiMessages - seiMessages

addCallbackListener

```
public void addCallbackListener(IMediaStreamCallback callbackListener)
```

Add a callback listener to listen for all stream metadata

Parameters:

callbackListener

removeCallbackListener

```
public void removeCallbackListener(IMediaStreamCallback callbackListener)
```

Remove a callback listener to listen for all stream metadata

(continued from last page)

Parameters:

callbackListener

addLivePacketListener

```
public void addLivePacketListener(IMediaStreamLivePacketNotify listener)
```

Add a listener to receiver notification of new live AMF packets

Parameters:

listener - listener

removeLivePacketListener

```
public void removeLivePacketListener(IMediaStreamLivePacketNotify listener)
```

Remove a listener to receiver notification of new live AMF packets

Parameters:

listener - listener

notifyLivePacket

```
public void notifyLivePacket(AMFPacket packet)
```

Notify new live AMF packet

Parameters:

packet - new live AMF packet

isPublisherStream

```
public boolean isPublisherStream()
```

Returns true if stream was created by stream publisher API.

Returns:

true if stream was created by stream publisher API

setPublisherStream

```
public void setPublisherStream(boolean isPublisherStream)
```

Set stream was created by stream publisher.

Parameters:

isPublisherStream - true if stream was created by stream publisher API

com.wowza.wms.stream Interface IMediaStreamActionNotify

All Subinterfaces:

[IMediaStreamActionNotify2](#), [IMediaStreamActionNotify3](#)

public interface **IMediaStreamActionNotify**
extends

IMediaStreamActionNotify: listener interface used by IMediaStream addClientListener

Method Summary

void	onPause (IMediaStream stream, boolean isPause, double location) Triggered on mediaStream pause
void	onPlay (IMediaStream stream, String streamName, double playStart, double playLen, int playReset) Triggered on mediaStream play
void	onPublish (IMediaStream stream, String streamName, boolean isRecord, boolean isAppend) Triggered on mediaStream publish
void	onSeek (IMediaStream stream, double location) Triggered on mediaStream seek
void	onStop (IMediaStream stream) Triggered on mediaStream stop
void	onUnPublish (IMediaStream stream, String streamName, boolean isRecord, boolean isAppend) Triggered on mediaStream unpublish

Methods

onPlay

```
public void onPlay(IMediaStream stream,  
    String streamName,  
    double playStart,  
    double playLen,  
    int playReset)
```

Triggered on mediaStream play

Parameters:

stream - mediaStream
streamName - streamName
playStart - playStart offset
playLen - playLen
playReset - reset playlist

onPublish

```
public void onPublish(IMediaStream stream,  
    String streamName,  
    boolean isRecord,  
    boolean isAppend)
```

Triggered on mediaStream publish

Parameters:

stream - mediaStream
streamName - streamName
isRecord - recording stream
isAppend - appending to file

onUnPublish

```
public void onUnPublish(IMediaStream stream,  
    String streamName,  
    boolean isRecord,  
    boolean isAppend)
```

Triggered on mediaStream unpublish

Parameters:

stream - mediaStream
streamName - streamName
isRecord - recording stream
isAppend - appending to file

onPause

```
public void onPause(IMediaStream stream,  
    boolean isPause,  
    double location)
```

Triggered on mediaStream pause

Parameters:

stream - mediaStream
isPause - pause or unpause
location - location (milliseconds)

onSeek

```
public void onSeek(IMediaStream stream,  
    double location)
```

Triggered on mediaStream seek

Parameters:

stream - mediaStream
location - location (milliseconds)

onStop

```
public void onStop(IMediaStream stream)
```

Triggered on mediaStream stop

(continued from last page)

Parameters:

stream - mediaStream

com.wowza.wms.stream Interface IMediaStreamActionNotify2

All Superinterfaces:

[IMediaStreamActionNotify](#)

All Subinterfaces:

[IMediaStreamActionNotify3](#)

public interface **IMediaStreamActionNotify2**
extends [IMediaStreamActionNotify](#)

Method Summary

void	onMetaData (IMediaStream stream, AMFPacket metaDataPacket) Triggered when a published streams metadata is set or changes
void	onPauseRaw (IMediaStream stream, boolean isPause, double location) Triggered on mediaStream pauseRaw.

Methods inherited from interface [com.wowza.wms.stream.IMediaStreamActionNotify](#)

[onPause](#), [onPlay](#), [onPublish](#), [onSeek](#), [onStop](#), [onUnPublish](#)

Methods

onMetaData

```
public void onMetaData(IMediaStream stream,  
    AMFPacket metaDataPacket)
```

Triggered when a published streams metadata is set or changes

onPauseRaw

```
public void onPauseRaw(IMediaStream stream,  
    boolean isPause,  
    double location)
```

Triggered on mediaStream pauseRaw. The pauseRaw method is called when a pause occurs in the player.

Parameters:

stream - mediaStream
isPause - pause or unpause
location - location (milliseconds)

com.wowza.wms.stream Interface IMediaStreamActionNotify3

All Superinterfaces:

[IMediaStreamActionNotify2](#), [IMediaStreamActionNotify](#)

public interface **IMediaStreamActionNotify3**

extends [IMediaStreamActionNotify2](#)

Method Summary

void	onCodecInfoAudio (IMediaStream stream, com.wowza.wms.media.model.MediaCodecInfoAudio codecInfoAudio) Triggered when publishing stream receives codec information.
void	onCodecInfoVideo (IMediaStream stream, com.wowza.wms.media.model.MediaCodecInfoVideo codecInfoVideo) Triggered when publishing stream receives codec information.

Methods inherited from interface [com.wowza.wms.stream.IMediaStreamActionNotify2](#)

[onMetaData](#), [onPauseRaw](#)

Methods inherited from interface [com.wowza.wms.stream.IMediaStreamActionNotify](#)

[onPause](#), [onPlay](#), [onPublish](#), [onSeek](#), [onStop](#), [onUnPublish](#)

Methods

onCodecInfoVideo

```
public void onCodecInfoVideo(IMediaStream stream,  
    com.wowza.wms.media.model.MediaCodecInfoVideo codecInfoVideo)
```

Triggered when publishing stream receives codec information.

Parameters:

stream - mediaStream

codecInfoVideo - video codec information

onCodecInfoAudio

```
public void onCodecInfoAudio(IMediaStream stream,  
    com.wowza.wms.media.model.MediaCodecInfoAudio codecInfoAudio)
```

Triggered when publishing stream receives codec information.

Parameters:

stream - mediaStream

codecInfoAudio - video codec information

com.wowza.wms.stream

Interface IMediaStreamCallback

public interface **IMediaStreamCallback**
extends

IMediaStreamCallback: callback interface used by IMediaStream registerCallback, registerOnStatus, registerOnPlayStatus

Method Summary	
void	<div><div>onCallback(IMediaStream stream, com.wowza.wms.request.RequestFunction function, AMFDataList params)</div><div>Triggered on callback event</div></div>

Methods

onCallback

```
public void onCallback(IMediaStream stream,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

Triggered on callback event

Parameters:

- stream - mediaStream
- function - function that triggered callback
- params - function parameters

com.wowza.wms.stream Interface IMediaStreamFileMapper

public interface **IMediaStreamFileMapper**
extends

Interface for mapping a IMediaStream to the underlying file system. Implement this interface and set your class using `IApplicationInstance.setStreamFileMapper(IMediaStreamFileMapper streamFileMapper)`. Your class will then be called each time a stream needs to be mapped to the underlying file system.

Method Summary

java.io.File	<code>streamToFileForRead(IMediaStream stream)</code> Get the File object to read from a stream (get stream name, ext and query from stream object)
java.io.File	<code>streamToFileForRead(IMediaStream stream, String name, String ext, String query)</code> Get the File object to read from a stream (specify name, ext and query)
java.io.File	<code>streamToFileForWrite(IMediaStream stream)</code> Get the File object to write to a stream (get stream name, ext and query from stream object)
java.io.File	<code>streamToFileForWrite(IMediaStream stream, String name, String ext, String query)</code> Get the File object to write to a stream (specify name, ext and query)

Methods

streamToFileForRead

```
public java.io.File streamToFileForRead(IMediaStream stream,  
    String name,  
    String ext,  
    String query)
```

Get the File object to read from a stream (specify name, ext and query)

Parameters:

stream - stream
name - stream name
ext - stream prefix (Ex. mp4:)
query - query part of stream name (Ex. mystream?param1=value1)

Returns:

resultant File object

streamToFileForRead

```
public java.io.File streamToFileForRead(IMediaStream stream)
```

Get the File object to read from a stream (get stream name, ext and query from stream object)

Parameters:

(continued from last page)

stream - stream

Returns:

resultant File object

streamToFileForWrite

```
public java.io.File streamToFileForWrite(IMediaStream stream,  
    String name,  
    String ext,  
    String query)
```

Get the File object to write to a stream (specify name, ext and query)

Parameters:

stream - stream

name - stream name

ext - stream prefix (Ex. mp4:)

query - query part of stream name (Ex. mystream?param1=value1)

Returns:

resultant File object

streamToFileForWrite

```
public java.io.File streamToFileForWrite(IMediaStream stream)
```

Get the File object to write to a stream (get stream name, ext and query from stream object)

Parameters:

stream - stream

Returns:

resultant File object

com.wowza.wms.stream

Interface IMediaStreamH264SEINotify

public interface IMediaStreamH264SEINotify

extends

Method Summary	
void	onVideoH264Packet (IMediaStream stream, AMFPacket packet, com.wowza.wms.media.h264.H264SEIMessages seiMessages)

Methods

onVideoH264Packet

public void **onVideoH264Packet**([IMediaStream](#) stream, [AMFPacket](#) packet, [com.wowza.wms.media.h264.H264SEIMessages](#) seiMessages)

com.wowza.wms.stream**Interface IMediaStreamLivePacketNotify**

public interface **IMediaStreamLivePacketNotify**
extends

Method Summary

void	onLivePacket (IMediaStream stream, AMFPacket packet)
------	---

Methods**onLivePacket**

public void **onLivePacket**([IMediaStream](#) stream,
[AMFPacket](#) packet)

com.wowza.wms.stream Interface IMediaStreamMediaCaster

public interface **IMediaStreamMediaCaster**
extends

IMediaStreamMediaCaster: Internal use

Method Summary

IMediaCaster	getMediaCaster()
void	setMediaCasterItem(MediaCasterItem mediaCasterItem)

Methods

setMediaCasterItem

public void **setMediaCasterItem**([MediaCasterItem](#) mediaCasterItem)

getMediaCaster

public [IMediaCaster](#) **getMediaCaster**()

com.wowza.wms.stream Interface IMediaStreamMetaDataProvider

public interface **IMediaStreamMetaDataProvider**
extends

IMediaStreamMetaDataProvider: Live streaming metadata provider.

Method Summary

void	onStreamStart (java.util.List metaDataList, long timecode) Called to get the onMetadata data at a particular point in time in the stream
------	---

Methods

onStreamStart

```
public void onStreamStart(java.util.List metaDataList,  
    long timecode)
```

Called to get the onMetadata data at a particular point in time in the stream

Parameters:

metaDataList - list of onMetadata data (should always be one item in list)
timecode - timecode in milliseconds

com.wowza.wms.stream Interface IMediaStreamNameAliasProvider

All Subinterfaces:

[IMediaStreamNameAliasProvider2](#)

public interface **IMediaStreamNameAliasProvider**
extends

IMediaStreamNameAliasProvider: alias provider interface. See `IApplication.setStreamNameAliasProvider(IMediaStreamNameAliasProvider streamNameAliasProvider)`. See updated interface `IMediaStreamNameAliasProvider2` which provides more complete context information.

Method Summary

String	resolvePlayAlias (IApplicationInstance appInstance, String name) Called to resolve a play alias
String	resolveStreamAlias (IApplicationInstance appInstance, String name) Called to resolve a stream alias

Methods

resolvePlayAlias

```
public String resolvePlayAlias(IApplicationInstance appInstance,  
    String name)
```

Called to resolve a play alias

Parameters:

appInstance - applicationInstance
name - stream name

Returns:

resultant name, null if want to block playback

resolveStreamAlias

```
public String resolveStreamAlias(IApplicationInstance appInstance,  
    String name)
```

Called to resolve a stream alias

Parameters:

appInstance - applicationInstance
name - stream name

Returns:

resultant name, null if want to block playback

com.wowza.wms.stream Interface IMediaStreamNameAliasProvider2

All Superinterfaces:

[IMediaStreamNameAliasProvider](#)

public interface **IMediaStreamNameAliasProvider2**

extends [IMediaStreamNameAliasProvider](#)

IMediaStreamNameAliasProvider: alias provider interface. See

[IApplication.setStreamNameAliasProvider\(IMediaStreamNameAliasProvider streamNameAliasProvider\)](#).

Method Summary

String	resolvePlayAlias (IApplicationInstance appInstance, String name, IClient client) Resolve play alias for RTMP streaming
String	resolvePlayAlias (IApplicationInstance appInstance, String name, IHTTPStreamerSession httpSession) Resolve play alias for HTTP streaming.
String	resolvePlayAlias (IApplicationInstance appInstance, String name, ILiveStreamPacketizer liveStreamPacketizer) Resolve play alias for live stream packetizer
String	resolvePlayAlias (IApplicationInstance appInstance, String name, RTPSession rtpSession) Resolve play alias for RTSP/RTP streaming
String	resolveStreamAlias (IApplicationInstance appInstance, String name, IMediaCaster mediaCaster) Resolve stream alias for MediaCaster

Methods inherited from interface [com.wowza.wms.stream.IMediaStreamNameAliasProvider](#)

[resolvePlayAlias](#), [resolveStreamAlias](#)

Methods

resolvePlayAlias

```
public String resolvePlayAlias(IApplicationInstance appInstance,  
    String name,  
    IClient client)
```

Resolve play alias for RTMP streaming

Parameters:

appInstance - application instance
name - stream name
client - client interface

Returns:

(continued from last page)

resultant name, null if want to block playback

resolvePlayAlias

```
public String resolvePlayAlias(IApplicationInstance appInstance,  
    String name,  
    IHTTPStreamerSession httpSession)
```

Resolve play alias for HTTP streaming. This callback may be called when there is not a valid HTTP session. In this case a phantom HTTP session will be created and will be populated with information from the underlying HTTP request (such as IP address and query string) but the session Id will be null.

Parameters:

appInstance - application instance

name - name

httpSession - HTTP session

Returns:

resultant name, null if want to block playback

resolvePlayAlias

```
public String resolvePlayAlias(IApplicationInstance appInstance,  
    String name,  
    RTPSession rtpSession)
```

Resolve play alias for RTSP/RTP streaming

Parameters:

appInstance - application instance

name - name

rtpSession - RTP session

Returns:

resultant name, null if want to block playback

resolvePlayAlias

```
public String resolvePlayAlias(IApplicationInstance appInstance,  
    String name,  
    ILiveStreamPacketizer liveStreamPacketizer)
```

Resolve play alias for live stream packetizer

Parameters:

appInstance - application instance

name - name

liveStreamPacketizer - live stream packetizer

Returns:

resultant name, null if want to block playback

resolveStreamAlias

```
public String resolveStreamAlias(IApplicationInstance appInstance,  
    String name,  
    IMediaCaster mediaCaster)
```

Resolve stream alias for MediaCaster

(continued from last page)

Parameters:

appInstance - application instance
name - name
mediaCaster - media caster

Returns:

resultant name, null if want to block playback

com.wowza.wms.stream Interface IMediaStreamNotify

public interface **IMediaStreamNotify**
extends

IMediaStreamNotify: listener interface used by IApplicationInstance addMediaStreamListener

Method Summary

void	onMediaStreamCreate (IMediaStream stream) Triggered when mediaStream created
void	onMediaStreamDestroy (IMediaStream stream) Triggered when mediaStream destroyed

Methods

onMediaStreamCreate

public void **onMediaStreamCreate**([IMediaStream](#) stream)

Triggered when mediaStream created

Parameters:

stream - mediaStream

onMediaStreamDestroy

public void **onMediaStreamDestroy**([IMediaStream](#) stream)

Triggered when mediaStream destroyed

Parameters:

stream - mediaStream

com.wowza.wms.stream Interface IMediaStreamPlay

public interface **IMediaStreamPlay**
extends

IMediaStreamPlay: public interface of mediaPlayer object.

Field Summary

public static final	PAUSE_PAUSE Pause type: pause Value: 1
public static final	PAUSE_PLAY Pause type: play Value: 0
public static final	PAUSE_TOGGLE Pause type: toggle Value: -1
public static final	PLAYSIZES_AUDIO_BYTES IMediaReader sizes array: audio byte count Value: 0
public static final	PLAYSIZES_AUDIO_COUNT IMediaReader sizes array: audio packet count Value: 1
public static final	PLAYSIZES_DATA_BYTES IMediaReader sizes array: data byte count Value: 4
public static final	PLAYSIZES_DATA_COUNT IMediaReader sizes array: data packet count Value: 5
public static final	PLAYSIZES_LOSS_BYTES IMediaReader sizes array: data byte count Value: 6
public static final	PLAYSIZES_LOSS_COUNT IMediaReader sizes array: data packet count Value: 7
public static final	PLAYSIZES_SIZE IMediaReader sizes array: size of sizes array long[PLAYSIZES_SIZE] Value: 8
public static final	PLAYSIZES_VIDEO_BYTES IMediaReader sizes array: video byte count Value: 2

public static final	PLAYSIZES_VIDEO_COUNT IMediaReader sizes array: video packet count Value: 3
public static final	PLAYSTATUSTYPE_COMPLETE onPlayStatus type: complete Value: 2
public static final	PLAYSTATUSTYPE_STOP onPlayStatus type: stop Value: 3
public static final	PLAYSTATUSTYPE_SWITCH onPlayStatus type: switch Value: 1

Method Summary

void	close() Close mediaStreamPlay
IMediaStream	getParent() Get the parent media stream object
void	init (IMediaStream parent, MediaStreamMap streams) Initialize mediaStreamPlayer
void	initLiveStreamRepeating (String liveStreamPacketizer, String liveStreamRepeater) Initialize this stream for live stream repeating
void	interruptPlay() Interrupt play to perform operation.
double	length() Get stream length/seconds (seconds)
void	pause (int pauseType, long timecode) pause mediaPlayStream
void	pauseRaw (int pauseType, long timecode) pauseRaw mediaPlayStream
boolean	play() Return true if there are packets to play
int	play (java.io.OutputStream out, AMFObj wmsObjAudio, AMFObj wmsObjVideo, AMFObj wmsObjData, long[] sizes) Write new packets or play packets
void	reset (boolean isReset) Reset mediaStreamPlayer
void	resetNoLookup() Reset but do not lookup current position in live stream
void	seek (int location) seek mediaPlayStream

void	<code>setBufferTime</code> (int bufferTime) Set buffer time
void	<code>setName</code> (String name, String oldName, String ext, String queryStr, double playStart, double playLen, int playTransition) Set mediaPlayStream name, extension, query string, play start, play len, play reset
void	<code>shutdown</code> () Shutdown mediaStreamPlayer
long	<code>size</code> () Get stream media file size
void	<code>startPlay</code> () Start playing stream
void	<code>stopName</code> (String name, String oldName, String ext, String queryStr, double playStart, double playLen, int playTransition) Stop stream name
void	<code>switchName</code> (String name, String oldName, String ext, String queryStr, double playStart, double playLen, int playTransition) Switch to stream name
void	<code>switchPlay</code> ()
void	<code>updateLoggingValues</code> () Update internal logging values

Fields

PAUSE_PAUSE

public static final int **PAUSE_PAUSE**

Pause type: pause
Constant value: **1**

PAUSE_TOGGLE

public static final int **PAUSE_TOGGLE**

Pause type: toggle
Constant value: **-1**

PAUSE_PLAY

public static final int **PAUSE_PLAY**

Pause type: play
Constant value: **0**

PLAYSTATUSTYPE_SWITCH

public static final int **PLAYSTATUSTYPE_SWITCH**

onPlayStatus type: switch

(continued from last page)

Constant value: **1**

PLAYSTATUSTYPE_COMPLETE

```
public static final int PLAYSTATUSTYPE_COMPLETE
```

onPlayStatus type: complete
Constant value: **2**

PLAYSTATUSTYPE_STOP

```
public static final int PLAYSTATUSTYPE_STOP
```

onPlayStatus type: stop
Constant value: **3**

PLAYSIZES_AUDIO_BYTES

```
public static final int PLAYSIZES_AUDIO_BYTES
```

IMediaReader sizes array: audio byte count
Constant value: **0**

PLAYSIZES_AUDIO_COUNT

```
public static final int PLAYSIZES_AUDIO_COUNT
```

IMediaReader sizes array: audio packet count
Constant value: **1**

PLAYSIZES_VIDEO_BYTES

```
public static final int PLAYSIZES_VIDEO_BYTES
```

IMediaReader sizes array: video byte count
Constant value: **2**

PLAYSIZES_VIDEO_COUNT

```
public static final int PLAYSIZES_VIDEO_COUNT
```

IMediaReader sizes array: video packet count
Constant value: **3**

PLAYSIZES_DATA_BYTES

```
public static final int PLAYSIZES_DATA_BYTES
```

IMediaReader sizes array: data byte count
Constant value: **4**

PLAYSIZES_DATA_COUNT

```
public static final int PLAYSIZES_DATA_COUNT
```

IMediaReader sizes array: data packet count
Constant value: **5**

(continued from last page)

PLAYSIZES_LOSS_BYTES

```
public static final int PLAYSIZES_LOSS_BYTES
```

IMediaReader sizes array: data byte count
Constant value: **6**

PLAYSIZES_LOSS_COUNT

```
public static final int PLAYSIZES_LOSS_COUNT
```

IMediaReader sizes array: data packet count
Constant value: **7**

PLAYSIZES_SIZE

```
public static final int PLAYSIZES_SIZE
```

IMediaReader sizes array: size of sizes array long[PLAYSIZES_SIZE]
Constant value: **8**

Methods

init

```
public void init(IMediaStream parent,  
                MediaStreamMap streams)
```

Initialize mediaStreamPlayer

Parameters:

parent - parent mediaStream object
streams - parent mediaStreamMap

shutdown

```
public void shutdown()
```

Shutdown mediaStreamPlayer

reset

```
public void reset(boolean isReset)
```

Reset mediaStreamPlayer

Parameters:

isReset - is this a playlist reset or addition, true if reset

resetNoLookup

```
public void resetNoLookup()
```

Reset but do not lookup current position in live stream

(continued from last page)

play

```
public int play(java.io.OutputStream out,
    AMFObj wmsObjAudio,
    AMFObj wmsObjVideo,
    AMFObj wmsObjData,
    long[] sizes)
```

Write new packets or play packets

Parameters:

out - OutputStream
wmsObjAudio - audio response channel
wmsObjVideo - video response channel
wmsObjData - data response channel
sizes - sizes array. See IMediaStreamPlay.PLAYSIZES_*

Returns:

total byte output

play

```
public boolean play()
```

Return true if there are packets to play

Returns:

true if packets to play

seek

```
public void seek(int location)
```

seek mediaPlayStream

Parameters:

location - timecode (milliseconds)

pause

```
public void pause(int pauseType,
    long timecode)
```

pause mediaPlayStream

Parameters:

pauseType - pause type. See IMediaStreamPlay.PAUSE_*
timecode - timecode (milliseconds)

pauseRaw

```
public void pauseRaw(int pauseType,
    long timecode)
```

pauseRaw mediaPlayStream

Parameters:

pauseType - pause type. See IMediaStreamPlay.PAUSE_*
timecode - timecode (milliseconds)

setBufferTime

```
public void setBufferTime(int bufferTime)
```

Set buffer time

Parameters:

bufferTime

setName

```
public void setName(String name,  
    String oldName,  
    String ext,  
    String queryStr,  
    double playStart,  
    double playLen,  
    int playTransition)
```

Set mediaPlayStream name, extension, query string, play start, play len, play reset

Parameters:

name - stream name
oldName - old stream name
ext - extension
queryStr - query string
playStart - play start
playLen - play len
playTransition - play reset

switchName

```
public void switchName(String name,  
    String oldName,  
    String ext,  
    String queryStr,  
    double playStart,  
    double playLen,  
    int playTransition)
```

Switch to stream name

Parameters:

name - stream name
oldName - old stream name
ext - stream extension
queryStr - query string
playStart - play start
playLen - play len
playTransition - play transition (see MediaBase.PLAYTRANSITION_*)

stopName

```
public void stopName(String name,  
    String oldName,  
    String ext,  
    String queryStr,  
    double playStart,  
    double playLen,  
    int playTransition)
```

(continued from last page)

Stop stream name

Parameters:

name - stream name
oldName - old stream name
ext - stream extension
queryStr - query string
playStart - play start
playLen - play len
playTransition - play transition (see MediaBase.PLAYTRANSITION_*)

close

```
public void close()
```

Close mediaStreamPlay

interruptPlay

```
public void interruptPlay()
```

Interrupt play to perform operation. Usually for seek or pause.

startPlay

```
public void startPlay()
```

Start playing stream

switchPlay

```
public void switchPlay()
```

updateLoggingValues

```
public void updateLoggingValues()
```

Update internal logging values

length

```
public double length()
```

Get stream length/seconds (seconds)

Returns:

length/duration (seconds)

size

```
public long size()
```

Get stream media file size

Returns:

media file size

getParent

```
public IMediaStream getParent( )
```

Get the parent media stream object

Returns:

parent media stream object

initLiveStreamRepeating

```
public void initLiveStreamRepeating(String liveStreamPacketizer,  
    String liveStreamRepeater)
```

Initialize this stream for live stream repeating

Parameters:

liveStreamPacketizer - live stream packetizer

liveStreamRepeater - live stream repeater name

com.wowza.wms.stream**Interface IMediaStreamTimecodeControl**

public interface **IMediaStreamTimecodeControl**
extends

IMediaStreamTimecodeControl: Internal use.

Method Summary

void	resetTimecodes()
------	----------------------------------

Methods

resetTimecodes

public void **resetTimecodes**()

com.wowza.wms.stream

Interface IMediaWriter

public interface **IMediaWriter**
extends

IMediaWriter: generic media writer interface. The flv recording system using this interface to persist .flv data captured from the Flash client. These classes are referenced in [\[install-dir\]/conf/MediaWriters.xml](#).

Example IMediaWriter implementation: MediaWriterFLVBasic

This is a basic IMediaWriter implementation that can handle record and append.

```
import java.io.*;
import java.nio.ByteBuffer;
import java.util.*;

import com.wowza.util.*;
import com.wowza.wms.stream.*;
import com.wowza.wms.amf.AMFData;
import com.wowza.wms.logging.*;

public class MediaWriterFLV implements IMediaWriter
{
    private IMediaStream parent = null;
    private MediaWriterItem mediaWriterItem = null;
    private long[] currentTCs = new long[3];
    private long duration = 0;
    private Map extraMetadata = new HashMap();
    private boolean versionFile = false;

    public void setMediaWriterItem(MediaWriterItem mediaWriterItem)
    {
        this.mediaWriterItem = mediaWriterItem;
    }

    public void setParent(IMediaStream parent)
    {
        this.parent = parent;
    }

    public void writePackets(List audioPackets, List videoPackets,
                             List dataPackets, List audioTCs, List videoTCs, List dataTCs, List
dataTypes,
                             boolean isFirst, boolean isLast)
    {
        File newFile = this.parent.getStreamFile();

        boolean localAppend = this.parent.isAppend();

        if (isFirst)
        {
            long startTC = 0;
            if (newFile.exists())
            {
                if (localAppend)
                    startTC = FLVUtils.getLastTC(newFile);
                else
                {
                    if (versionFile)
                        FileUtils.versionFile(newFile);
                    else
                    {
                        try
```

```

        {
            newFile.delete();
        }
        catch (Exception e)
        {
        }
    }
}
else
    localAppend = false;

this.currentTCs[FLVUtils.FLV_TCINDEXAUDIO] = startTC;
this.currentTCs[FLVUtils.FLV_TCINDEXVIDEO] = startTC;
this.currentTCs[FLVUtils.FLV_TCINDEXDATA] = startTC;
}
else
    localAppend = true;

try
{
    if (newFile.getParentFile() == null)

WMSLoggerFactory.getLogger(MediaWriterFLV.class).warn("MediaWriterFLV: File path does not
exist: "+newFile.getPath());
        else if (!newFile.getParentFile().exists())

WMSLoggerFactory.getLogger(MediaWriterFLV.class).warn("MediaWriterFLV: Folder does not exist:
"+newFile.getParentFile().getPath());
        else if (newFile.exists() && !newFile.canWrite())

WMSLoggerFactory.getLogger(MediaWriterFLV.class).warn("MediaWriterFLV: Cannot write to file
(permission error): "+newFile.getPath());

        FileOutputStream ds = new FileOutputStream(newFile, localAppend);

        if (isFirst)
        {
            if (!localAppend)
            {
                FLVUtils.writeHeader(ds, 0.0, extraMetadata);

                boolean writeZeroPacket = true;
                while(true)
                {
                    if (audioPackets.size() == 0)
                        break;

                    ByteBuffer data =
(ByteBuffer)audioPackets.get(0);

                    long tcA = ((Long)audioTCs.get(0)).longValue();

```

```

        if (tcA == 0 && data.limit() == 0)
            writeZeroPacket = false;

        break;
    }

    if (writeZeroPacket)
    {
        FLVUtils.writeChunk(ds, null, 0,

this.currentTCs[FLVUtils.FLV_TCINDEXAUDIO],

                                                                    (byte) 0x08); //
write zero length audio block
    }
}

    FLVUtils.writePackets(ds, audioPackets, videoPackets, dataPackets,
        audioTCs, videoTCs, dataTCs, dataTypes, currentTCs);

    ds.flush();
    ds.close();
}
catch (Exception e)
{
    WMSLoggerFactory.getLogger(MediaWriterFLV.class).error(
        "MediaWriterFLV: Error writing to file:
"+newFile.getPath()+": "+e.toString());
    e.printStackTrace();
}

    if (isLast)
    {
        duration = Math.max(Math.max(currentTCs[FLVUtils.FLV_TCINDEXAUDIO],
            currentTCs[FLVUtils.FLV_TCINDEXVIDEO]),
            currentTCs[FLVUtils.FLV_TCINDEXDATA]);
        double durationSecs = ((double)duration) / 1000.0;

        FLVUtils.writeDuration(newFile, durationSecs);
    }
}

public Map getExtraMetadata()
{
    return extraMetadata;
}

public void setExtraMetadata(Map extraMetadata)
{
    this.extraMetadata = extraMetadata;
}

```

```
    }

    public boolean isVersionFile()
    {
        return versionFile;
    }

    public void setVersionFile(boolean versionFile)
    {
        this.versionFile = versionFile;
    }

    public void putMetaData(String name, AMFData value)
    {
        this.extraMetadata.put(name, value);
    }
}
```

To use this class, edit [install-dir]/conf/MediaWriter and replace the definition for the **flv** MediaWriter:

```
<MediaWriter>
  <Name>flv</Name>
  <Description>FLV Media Writer</Description>
  <FileExtension>flv</FileExtension>
  <ClassBase>com.wowza.wms.plugin.mediawriter.flv.MediaWriterFLVBasic</ClassBase>
</MediaWriter>
```

Example IMediaWriter implementation: MediaWriterFLVMetadata

This example illustrates how to write custom metadata into the recorded flv file on the fly.

```
public class MediaWriterFLVMetadata implements IMediaWriter
{
    private IMediaStream parent = null;
    private MediaWriterItem mediaWriterItem = null;
    private long[] currentTCs = new long[3];
    private long duration = 0;
    private File tmpFile = null;
    private Map extraMetadata = new HashMap();
    private boolean versionFile = false;

    public void setMediaWriterItem(MediaWriterItem mediaWriterItem)
    {
        this.mediaWriterItem = mediaWriterItem;
    }

    public void setParent(IMediaStream parent)
    {
        this.parent = parent;
    }

    public void writePackets(List audioPackets, List videoPackets,
        List dataPackets, List audioTCs, List videoTCs, List dataTCs,
        boolean isFirst, boolean isLast)
    {
        File newFile = this.parent.getStreamFile();
        try
        {
            if (tmpFile == null)
                tmpFile = File.createTempFile("wowza", "flv");
        }
        catch (Exception e)
        {
            WMSLoggerFactory.getLogger(MediaWriterFLVBasic.class).error(
                "MediaWriterFLVMetadata: Error createTempFile: "+
                tmpFile+" :"+e.toString());
        }

        boolean localAppend = this.parent.isAppend();
        if (isFirst)
        {
            AMFDataArray keyFrames = null;

            long startTC = 0;
            if (newFile.exists())
            {
                if (localAppend)
                {
                    startTC = FLVUtils.getLastTC(newFile);
                    keyFrames = getKeyFrames(newFile);
                    copyPacketsToTmpFile(newFile, tmpFile);
                }
            }
        }
    }
}
```

```

        if (versionFile)
            FileUtils.versionFile(newFile);
        else
        {
            try
            {
                newFile.delete();
            }
            catch (Exception e)
            {
            }
        }
    }
    else
        localAppend = false;

    if (keyFrames == null)
        keyFrames = new AMFDataArray();
    extraMetadata.put("keyFrames", keyFrames);

    this.currentTCs[FLVUtils.FLV_TCINDEXAUDIO] = startTC;
    this.currentTCs[FLVUtils.FLV_TCINDEXVIDEO] = startTC;
    this.currentTCs[FLVUtils.FLV_TCINDEXDATA] = startTC;
}
else
    localAppend = true;

    AMFDataArray keyFrames = (AMFDataArray)extraMetadata.get("keyFrames");
    long timecode = this.currentTCs[FLVUtils.FLV_TCINDEXVIDEO];
    int size = videoPackets.size();
    for(int i=0;i<size;i++)
    {
        ByteBuffer data = (ByteBuffer)videoPackets.get(i);
        int firstByte = data.get(0);
        timecode += ((Long)videoTCs.get(i)).longValue();
        if (FLVUtils.getFrameType(firstByte) == FLVUtils.FLV_KFRAME)
        {
            double durationSecs = ((double)timecode) / 1000.0;
            AMFDataObj dataObj = new AMFDataObj();
            dataObj.put("name", new AMFDataItem("keyframe
"+durationSecs));

            dataObj.put("time", new AMFDataItem(durationSecs));
            keyFrames.add(dataObj);
        }
    }

    try
    {
        FileOutputStream ds = new FileOutputStream(tmpFile, localAppend);
        FLVUtils.writePackets(ds, audioPackets, videoPackets, dataPackets,

```



```

        audioTCs, videoTCs, dataTCs, currentTCs);

        ds.flush();
        ds.close();
    }
    catch (Exception e)
    {
        WMSLoggerFactory.getLogger(MediaWriterFLVBasic.class).error(
            "MediaWriterFLVMetadata: Error writing to tmp file:

"+

            newFile.getPath()+" :"+e.toString());
    }

    if (isLast)
    {
        duration = Math.max(Math.max(currentTCs[FLVUtils.FLV_TCINDEXAUDIO],
            currentTCs[FLVUtils.FLV_TCINDEXVIDEO]),
            currentTCs[FLVUtils.FLV_TCINDEXDATA]);
        double durationSecs = ((double)duration) / 1000.0;

        try
        {
            AMFPacket packet = null;
            FileOutputStream ds = new FileOutputStream(newFile);

            FileInputStream di = new FileInputStream(tmpFile);
            FLVUtils.writeHeader(ds, durationSecs, extraMetadata);
            while((packet = FLVUtils.readChunk(di)) != null)
            {
                FLVUtils.writeChunk(ds, packet.getDataBuffer(),
packet.getSize(),

                                packet.getTimecode(),
(byte)packet.getType());
            }
            di.close();

            ds.flush();
            ds.close();

            tmpFile.delete();
        }
        catch (Exception e)
        {
            WMSLoggerFactory.getLogger(MediaWriterFLVBasic.class).error(
                "MediaWriterFLVMetadata: Error tmp writing to

file: "+

                newFile.getPath()+" :"+e.toString());
        }
    }
}

private void copyPacketsToTmpFile(File newFile, File tmpFile)

```

```

{
    AMFDataArray keyFrames = null;
    try
    {
        AMFPacket packet = null;
        FileOutputStream ds = new FileOutputStream(tmpFile);

        FileInputStream di = new FileInputStream(newFile);
        FLVUtils.readHeader(di);
        FLVUtils.readChunk(di); // skip metaData packet
        while((packet = FLVUtils.readChunk(di)) != null)
        {
            FLVUtils.writeChunk(ds, packet.getDataBuffer(),
packet.getSize(),
                                packet.getTimecode(),
(byte)packet.getType());
        }
        di.close();

        ds.flush();
        ds.close();
    }
    catch (Exception e)
    {
        WMSLoggerFactory.getLogger(MediaWriterFLVBasic.class).error(
            "MediaWriterFLVMetadata: Error copyPacketsToTmpFile:
"+
            newFile.getPath()+" :"+e.toString());
    }
}

private AMFDataArray getKeyFrames(File newFile)
{
    AMFDataArray keyFrames = null;
    try
    {
        BufferedInputStream inStream = new BufferedInputStream(new
FileInputStream(newFile));
        FLVUtils.readHeader(inStream);
        AMFPacket packet = FLVUtils.readChunk(inStream);
        if (packet.getType() == IVHost.CONTENTTYPE_DATA0 || packet.getType()
== IVHost.CONTENTTYPE_DATA3)
        {
            byte[] mbytes = packet.getData();
            int moffset = 0;
            if (packet.getType() == IVHost.CONTENTTYPE_DATA3 &&
mbytes.length > 0)
            {
                if (mbytes[0] == 0)
                    moffset = 1;
            }
        }
    }
}

```

```

        AMFDataList dataList = new AMFDataList(mbytes, moffset,
mbytes.length-moffset);
        if (dataList.size() > 1)
        {
            if (dataList.get(1).getType() ==
AMFData.DATA_TYPE_MIXED_ARRAY)
            {
                AMFDataMixedArray metaValues =
                (AMFDataMixedArray)dataList.get(1);
                if (metaValues.containsKey("keyFrames"))
                    keyFrames =
                (AMFDataArray)metaValues.get("keyFrames");
            }
        }
        inStream.close();
    }
    catch (Exception e)
    {
        WMSLoggerFactory.getLogger(MediaWriterFLVBasic.class).error(
            "MediaWriterFLVMetadata: Error getKeyFrames: " +
            newFile.getPath()+" :"+e.toString());
    }

    return keyFrames;
}

public boolean isVersionFile()
{
    return versionFile;
}

public void setVersionFile(boolean versionFile)
{
    this.versionFile = versionFile;
}

public void putMetaData(String name, AMFData value)
{
    this.extraMetadata.put(name, value);
}
}

```

To use this class, edit [install-dir]/conf/MediaWriter and replace the definition for the **flv** MediaWriter:

```

<MediaWriter>
  <Name>flv</Name>
  <Description>FLV Media Writer</Description>
  <FileExtension>flv</FileExtension>
  <ClassBase>com.wowza.wms.plugin.mediawriter.flv.MediaWriterFLVMetadata</ClassBase>
</MediaWriter>

```

Method Summary

long	getDuration() Get the recorded duration of the file in seconds
boolean	isVersionFile() Return true if the old file is to be versioned
boolean	isWaitForVideoKeyFrame() get wait for key frame
void	putMetaData (String name, AMFData value) Add metadata to the metadata packet.
void	setMediaWriterItem (MediaWriterItem mediaWriterItem) Set the media write definition
void	setParent (IMediaStream parent) Set the parent stream for this media write object
void	setVersionFile (boolean versionFile) Set to true if the old file is to be versioned
void	setWaitForVideoKeyFrame (boolean waitForVideoKeyFrame) Set to true if you want the recorder to skip opening frames until it hits a key frame
void	writePackets (java.util.List audioPackets, java.util.List videoPackets, java.util.List dataPackets, java.util.List audioTCs, java.util.List videoTCs, java.util.List dataTCs, java.util.List dataTypes, boolean isFirst, boolean isLast) Invoked each time a set of packets are ready to be presisted.

Methods

writePackets

```

public void writePackets(java.util.List audioPackets,
    java.util.List videoPackets,
    java.util.List dataPackets,
    java.util.List audioTCs,
    java.util.List videoTCs,
    java.util.List dataTCs,
    java.util.List dataTypes,
    boolean isFirst,
    boolean isLast)

```

Invoked each time a set of packets are ready to be presisted.

(continued from last page)

Parameters:

audioPackets - List of audio packets
videoPackets - List of video packets
dataPackets - List of data packets
audioTCs - List of audio timecodes
videoTCs - List of video timecodes
dataTCs - List of data timecodes
dataTypes - list of integer packets types (IVHost.CONTENTTYPE_DATA0, IVHost.CONTENTTYPE_DATA3) - if null assumed to be IVHost.CONTENTTYPE_DATA0
isFirst - true if first packet to be written
isLast - false if last packet to be written

setMediaWriterItem

```
public void setMediaWriterItem(MediaWriterItem mediaWriterItem)
```

Set the media write definition

Parameters:

mediaWriterItem - media write definition

setParent

```
public void setParent(IMediaStream parent)
```

Set the parent stream for this media write object

Parameters:

parent

isVersionFile

```
public boolean isVersionFile()
```

Return true if the old file is to be versioned

Returns:

true if the old file is to be versioned

setVersionFile

```
public void setVersionFile(boolean versionFile)
```

Set to true if the old file is to be versioned

Parameters:

versionFile

isWaitForVideoKeyFrame

```
public boolean isWaitForVideoKeyFrame()
```

get wait for key frame

Returns:

wait for key frame

(continued from last page)

setWaitForVideoKeyFrame

```
public void setWaitForVideoKeyFrame(boolean waitForVideoKeyFrame)
```

Set to true if you want the recorder to skip opening frames until it hits a key frame

Parameters:

waitForVideoKeyFrame - wait for key frame

putMetaData

```
public void putMetaData(String name,  
    AMFData value)
```

Add metadata to the metadata packet. Only metadata added before the first call to writePackets will be included in the file

Parameters:

name - field name

value - metadata value

getDuration

```
public long getDuration()
```

Get the recorded duration of the file in seconds

Returns:

recorded duration of the file in seconds

com.wowza.wms.stream Interface IMediaWriterActionNotify

public interface **IMediaWriterActionNotify**
extends

IMediaWriterActionNotify: listener interface for file writing. See
IApplicationInstance.addMediaWriterListener(IMediaWriterActionNotify listener)

Method Summary

void	onFLVAddMetadata (IMediaStream stream, java.util.Map extraMetadata) Called just before metadata is written to the file (FLV only)
void	onWriteComplete (IMediaStream stream, java.io.File file) Called when writing is complete

Methods

onWriteComplete

```
public void onWriteComplete(IMediaStream stream,  
                             java.io.File file)
```

Called when writing is complete

Parameters:

stream - stream
file - file handle

onFLVAddMetadata

```
public void onFLVAddMetadata(IMediaStream stream,  
                             java.util.Map extraMetadata)
```

Called just before metadata is written to the file (FLV only)

Parameters:

stream - stream
extraMetadata - additional metadata, add to this collection to add items to onMetadata event written to FLV file.

com.wowza.wms.stream Class MediaReaderEncInfo

java.lang.Object

└─com.wowza.wms.stream.MediaReaderEncInfo

public class **MediaReaderEncInfo**
extends Object

Field Summary

public static final	METHOD_NONE Value: 0
public static final	METHOD_SAMPLE_PLAYREADY Value: 1

Constructor Summary

public	MediaReaderEncInfo()
--------	--------------------------------------

Method Summary

void	addPlayReadyDecryptorKey (com.wowza.wms.drm.playready.PlayReadyKeyInfo decryptorKey)
int	getEncMethod ()
com.wowza.wms.drm.playready.PlayReadyKeyInfo	getPlayReadyDecryptorKey ()
com.wowza.wms.drm.playready.PlayReadyKeyInfo	getPlayReadyDecryptorKey (String keyId)
void	setEncMethod (int encMethod)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

METHOD_NONE

public static final int **METHOD_NONE**

(continued from last page)

Constant value: **0**

METHOD_SAMPLE_PLAYREADY

```
public static final int METHOD_SAMPLE_PLAYREADY
```

Constant value: **1**

Constructors

MediaReaderEncInfo

```
public MediaReaderEncInfo()
```

Methods

addPlayReadyDecryptorKey

```
public void addPlayReadyDecryptorKey(com.wowza.wms.drm.playready.PlayReadyKeyInfo  
decryptorKey)
```

getPlayReadyDecryptorKey

```
public com.wowza.wms.drm.playready.PlayReadyKeyInfo getPlayReadyDecryptorKey()
```

getPlayReadyDecryptorKey

```
public com.wowza.wms.drm.playready.PlayReadyKeyInfo getPlayReadyDecryptorKey(String  
keyId)
```

getEncMethod

```
public int getEncMethod()
```

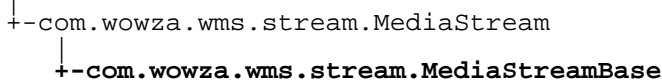
setEncMethod

```
public void setEncMethod(int encMethod)
```

com.wowza.wms.stream

Class MediaStreamBase

java.lang.Object



All Implemented Interfaces:

[IMediaStream](#)

```
public class MediaStreamBase
extends MediaStream
```

Field Summary

public static	p Deprecated.
public static	sinfo Deprecated.

Fields inherited from class com.wowza.wms.stream.MediaStream

actionListeners, AMLST_STREAM_EXT, audioBytes, audioSize, audioTC, BASE_STREAM_EXT, bufferTime, callbackListeners, callbacks, client, dataBytes, dataSize, dataTC, dataType, dvrLock, dvrMap, dvrPlayer, dvrRecorder, dvrRecorderList, dvrRepeater, elapsedTime, ext, fastPlaySettings, FLV_STREAM_EXT, h264SEIListeners, headerSize, httpStreamerSession, ID3_STREAM_EXT, isAppend, isClustered, isMediaCasterPlay, isOpen, isPlay, isPlaying, isPublisherStream, isRecord, isTranscodeResult, livePacketListeners, liveStreamPacketizer, liveStreamPacketizerList, liveStreamRepeater, liveStreamTranscoderList, lock, mediaIOPerformance, mergeOnMetadata, metaDataProvider, MP3_STREAM_EXT, MP4_STREAM_EXT, name, netConnection, NGRP_STREAM_EXT, parent, player, playLen, playStart, playTransition, properties, queryStr, receiveAudio, receiveVideo, receiveVideoFPS, rtpStream, SEND_CONTROL0, SEND_CONTROL1, SEND_CONTROL3, SEND_CONTROL4, sendDirectLock, sendDirectMessages, sendPlayStopLogEvent, sendPublishStopLogEvent, sendRecordStopLogEvent, SMIL_STREAM_EXT, src, STREAM_DEFAULTNAME, streamType, transcoderLock, transcoderMap, tss, videoBytes, videoSize, videoTC

Fields inherited from interface [com.wowza.wms.stream.IMediaStream](#)

[AUDIOSAMPLEACCESS](#), [READACCESS](#), [VIDEOSAMPLEACCESS](#), [WRITEACCESS](#)

Constructor Summary

public	MediaStreamBase()
--------	-----------------------------------

Method Summary

void	init (MediaStreamMap parent, int src, WMSProperties properties)
------	--

void	publish()
void	trim()

Methods inherited from class `com.wowza.wms.stream.MediaStream`

```
addAudioCodecConfigPacket, addAudioData, addAudioExtraData, addCallbackListener,
addClientListener, addClientListener, addClientListener, addDataData,
addDataExtraData, addLivePacketListener, addVideoCodecConfigPacket, addVideoData,
addVideoExtraData, addVideoH264SEILListener, clear, clearFastPlaySettings,
clearLoggingValues, close, flush, getAccess, getAudioCodecConfigPacket,
getAudioMissing, getAudioSize, getAudioTC, getBufferTime, getBurstStartStop,
getCacheName, getClient, getClientId, getContextStr, getDataMissing, getDataSize,
getDataTC, getDataType, getDvrRecorder, getDvrRecorder, getDvrRecorderList,
getDvrRepeater, getElapsedTime, getExt, getFastPlaySettings, getHeaderSize,
getHTTPStreamerSession, getLastKeyFrame, getLastPacket, getLiveStreamDvr,
getLiveStreamDvrs, getLiveStreamPacketizer, getLiveStreamPacketizer,
getLiveStreamPacketizerList, getLiveStreamRepeater, getLiveStreamTranscoder,
getLiveStreamTranscoderList, getLiveStreamTranscoders, getMaxTimecode,
getMediaIOPerformance, getMetaDataProvider, getName, getNetConnection, getPlayer,
getPlayPackets, getProperties, getPublishAudioCodecId, getPublishVideoCodecId,
getQueryStr, getReceiveVideoFPS, getRespAMFAudioObj, getRespAMFDataObj,
getRespAMFVideoObj, getRTPStream, getSrc, getStreamFileForRead, getStreamFileForRead,
getStreamFileForWrite, getStreamFileForWrite, getStreams, getStreamType, getTss,
getUniqueStreamIdStr, getVideoCodecConfigPacket, getVideoMissing, getVideoSize,
getVideoTC, handleCallback, idle, incrementMediaInBytes, incrementMediaLossBytes,
incrementMediaOutBytes, init, initDvrRepeating, initLiveStreamRepeating, isAppend,
isClustered, isMediaCasterPlay, isMergeOnMetadata, isOpen, isPlay, isPlaying,
isPublisherStream, isPublishStreamReady, isReceiveAudio, isReceiveVideo, isRecord,
isSendDirectMessages, isSendPlayStopLogEvent, isSendPublishStopLogEvent,
isSendRecordStopLogEvent, isTranscodeResult, isVideoH264SEILListenerEmpty, length,
notifyActionOnCodecInfoAudio, notifyActionOnCodecInfoVideo, notifyActionOnMetaData,
notifyActionPause, notifyActionPauseRaw, notifyActionPlay, notifyActionPublish,
notifyActionSeek, notifyActionStop, notifyActionUnPublish, notifyLivePacket,
notifyVideoH264Packet, packetComplete, processSendDirectMessages, publish,
putDvrRecorder, putLiveStreamTranscoder, registerCallback, registerOnPlayStatus,
registerOnStatus, removeCallbackListener, removeClientListener, removeClientListener,
removeClientListener, removeDvrRecorder, removeLivePacketListener,
removeLiveStreamTranscoder, removeVideoH264SEILListener, reset, send, send, sendAMF3,
sendAMF3, sendControlBytes, sendDataToCallback, sendDirect, sendDirect,
sendDirectAMF3, sendDirectAMF3, sendDirectInternal, sendDirectInternal, sendInternal,
sendInternal, sendLivePlaySeek, sendLivePlayStart, sendLivePlaySwitch,
sendPauseNotify, sendPauseNotify, sendPlayReset, sendPlayReset, sendPlaySeek,
sendPlaySeek, sendPlaySeek, sendPlayStart, sendPlayStart, sendPlayStart,
sendPlayStart, sendPlayStatus, sendPlayStatus, sendPlayStop, sendPlayStop,
sendPlaySwitch, sendPlaySwitch, sendStreamNotFound, sendStreamNotFound,
sendUnpauseNotify, sendUnpauseNotify, sendUnpauseNotify, sendVODPlaySwitch,
setAppend, setAudioSize, setAudioTC, setAudioTC, setBufferTime, setClient,
setClustered, setDataSize, setDataTC, setDataTC, setDataTC, setDataSize, setDvrRecorder,
setDvrRecorderList, setDvrRepeater, setExt, setFastPlaySettings, setHeaderSize,
setHTTPStreamerSession, setIsPlaying, setJustName, setLiveStreamPacketizer,
setLiveStreamPacketizerList, setLiveStreamRepeater, setLiveStreamTranscoderList,
setMediaCasterPlay, setMergeOnMetadata, setMetaDataProvider, setName, setName,
setName, setNetConnection, setOpen, setPlay, setPlayer, setPublishAudioCodecId,
setPublisherStream, setPublishVideoCodecId, setQueryStr, setReceiveAudio,
setReceiveVideo, setReceiveVideoFPS, setRecord, setRTPStream,
setSendPlayStopLogEvent, setSendPublishStopLogEvent, setSendRecordStopLogEvent,
setSrc, setStreamType, setTranscodeResult, setTss, setVideoSize, setVideoTC,
setVideoTC, shutdown, size, startAudioPacket, startDataPacket, startPublishing,
startVideoPacket, stopName, stopPublishing, switchName, trim, unregisterCallback,
unregisterOnPlayStatus, unregisterOnStatus, updateLoggingDuration,
updateLoggingValues
```

Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Methods inherited from interface [com.wowza.wms.stream.IMediaStream](#)

[addAudioCodecConfigPacket](#), [addAudioData](#), [addAudioExtraData](#), [addCalbackListener](#),
[addClientListener](#), [addClientListener](#), [addClientListener](#), [addDataData](#),
[addDataExtraData](#), [addLivePacketListener](#), [addVideoCodecConfigPacket](#), [addVideoData](#),
[addVideoExtraData](#), [addVideoH264SEIListener](#), [clear](#), [clearFastPlaySettings](#),
[clearLoggingValues](#), [close](#), [flush](#), [getAccess](#), [getAudioCodecConfigPacket](#),
[getAudioMissing](#), [getAudioSize](#), [getAudioTC](#), [getBufferTime](#), [getBurstStartStop](#),
[getCacheName](#), [getClient](#), [getClientId](#), [getContextStr](#), [getDataMissing](#), [getDataSize](#),
[getDataTC](#), [getDataType](#), [getDvrRecorder](#), [getDvrRecorder](#), [getDvrRecorderList](#),
[getDvrRepeater](#), [getElapsedTime](#), [getExt](#), [getFastPlaySettings](#), [getHeaderSize](#),
[getHTTPStreamerSession](#), [getLastKeyFrame](#), [getLastPacket](#), [getLiveStreamPacketizer](#),
[getLiveStreamPacketizer](#), [getLiveStreamPacketizerList](#), [getLiveStreamRepeater](#),
[getLiveStreamTranscoder](#), [getLiveStreamTranscoderList](#), [getLiveStreamTranscoders](#),
[getMaxTimecode](#), [getMediaIOPerformance](#), [getMetaDataProvider](#), [getName](#),
[getNetConnection](#), [getPlayer](#), [getPlayPackets](#), [getProperties](#), [getPublishAudioCodecId](#),
[getPublishVideoCodecId](#), [getQueryStr](#), [getReceiveVideoFPS](#), [getRespAMFAudioObj](#),
[getRespAMFDataObj](#), [getRespAMFVideoObj](#), [getRTPStream](#), [getSrc](#), [getStreamFileForRead](#),
[getStreamFileForRead](#), [getStreamFileForWrite](#), [getStreamFileForWrite](#), [getStreams](#),
[getStreamType](#), [getUniqueStreamIdStr](#), [getVideoCodecConfigPacket](#), [getVideoMissing](#),
[getVideoSize](#), [getVideoTC](#), [handleCallback](#), [idle](#), [incrementMediaInBytes](#),
[incrementMediaLossBytes](#), [incrementMediaOutBytes](#), [init](#), [initLiveStreamRepeating](#),
[isAppend](#), [isClustered](#), [isMediaCasterPlay](#), [isMergeOnMetadata](#), [isOpen](#), [isPlay](#),
[isPlaying](#), [isPublisherStream](#), [isPublishStreamReady](#), [isReceiveAudio](#), [isReceiveVideo](#),
[isRecord](#), [isSendPlayStopLogEvent](#), [isSendPublishStopLogEvent](#),
[isSendRecordStopLogEvent](#), [isTranscodeResult](#), [isVideoH264SEIListenerEmpty](#), [length](#),
[notifyActionOnCodecInfoAudio](#), [notifyActionOnCodecInfoVideo](#), [notifyActionOnMetaData](#),
[notifyActionPause](#), [notifyActionPauseRaw](#), [notifyActionPlay](#), [notifyActionPublish](#),
[notifyActionSeek](#), [notifyActionStop](#), [notifyActionUnPublish](#), [notifyLivePacket](#),
[notifyVideoH264Packet](#), [packetComplete](#), [publish](#), [putDvrRecorder](#),
[putLiveStreamTranscoder](#), [registerCallback](#), [registerOnPlayStatus](#), [registerOnStatus](#),
[removeCalbackListener](#), [removeClientListener](#), [removeClientListener](#),
[removeClientListener](#), [removeDvrRecorder](#), [removeLivePacketListener](#),
[removeLiveStreamTranscoder](#), [removeVideoH264SEIListener](#), [send](#), [send](#), [sendAMF3](#),
[sendAMF3](#), [sendControlBytes](#), [sendDirect](#), [sendDirect](#), [sendDirectAMF3](#), [sendDirectAMF3](#),
[sendLivePlaySeek](#), [sendLivePlayStart](#), [sendLivePlaySwitch](#), [sendPauseNotify](#),
[sendPauseNotify](#), [sendPlayReset](#), [sendPlayReset](#), [sendPlaySeek](#), [sendPlaySeek](#),
[sendPlaySeek](#), [sendPlayStart](#), [sendPlayStart](#), [sendPlayStart](#), [sendPlayStart](#),
[sendPlayStatus](#), [sendPlayStatus](#), [sendPlayStop](#), [sendPlayStop](#), [sendPlaySwitch](#),
[sendPlaySwitch](#), [sendStreamNotFound](#), [sendStreamNotFound](#), [sendUnpauseNotify](#),
[sendUnpauseNotify](#), [sendUnpauseNotify](#), [sendVODPlaySwitch](#), [setAppend](#), [setAudioSize](#),
[setAudioTC](#), [setAudioTC](#), [setBufferTime](#), [setClient](#), [setClustered](#), [setDataSize](#),
[setDataTC](#), [setDataTC](#), [setDataType](#), [setDvrRecorder](#), [setDvrRecorderList](#), [setExt](#),
[setFastPlaySettings](#), [setHeaderSize](#), [setHTTPStreamerSession](#), [setIsPlaying](#),
[setLiveStreamPacketizer](#), [setLiveStreamPacketizerList](#), [setLiveStreamRepeater](#),
[setLiveStreamTranscoderList](#), [setMediaCasterPlay](#), [setMergeOnMetadata](#),
[setMetaDataProvider](#), [setName](#), [setName](#), [setName](#), [setNetConnection](#), [setOpen](#), [setPlay](#),
[setPlayer](#), [setPublishAudioCodecId](#), [setPublisherStream](#), [setPublishVideoCodecId](#),
[setQueryStr](#), [setReceiveAudio](#), [setReceiveVideo](#), [setReceiveVideoFPS](#), [setRecord](#),
[setRTPStream](#), [setSendPlayStopLogEvent](#), [setSendPublishStopLogEvent](#),
[setSendRecordStopLogEvent](#), [setSrc](#), [setStreamType](#), [setTranscodeResult](#), [setVideoSize](#),
[setVideoTC](#), [setVideoTC](#), [shutdown](#), [size](#), [startAudioPacket](#), [startDataPacket](#),
[startPublishing](#), [startVideoPacket](#), [stopName](#), [stopPublishing](#), [switchName](#), [trim](#),
[unregisterCallback](#), [unregisterOnPlayStatus](#), [unregisterOnStatus](#),
[updateLoggingDuration](#), [updateLoggingValues](#)

Fields

(continued from last page)

sinfo

```
public static java.util.Map sinfo
```

Deprecated.

p

```
public static java.lang.String p
```

Deprecated.

Constructors

MediaStreamBase

```
public MediaStreamBase()
```

Methods

init

```
public void init(MediaStreamMap parent,  
                int src,  
                WMSPProperties properties)
```

publish

```
public void publish()
```

trim

```
public void trim()
```

com.wowza.wms.stream

Class MediaStreamMap

java.lang.Object

└─com.wowza.wms.stream.MediaStreamMap

public class **MediaStreamMap**
extends Object

MediaStreamMap: collection of IMediaStream object. This collection is usually attached to an IApplicationInstance object.

Field Summary

protected	appInstance
public	debugDVRShutdown
protected	dvrRecorders
protected	liveStreamPacketizers
public static final	MAXSTREAMINDEX Value: 65536
protected	mediaStreamListeners
protected	nameGroupDelayRemoveMap
protected	nameGroupId
protected	nameGroupRemoveDelayTime
protected	nameGroups
protected	nextStreamId
protected	packetizerLicenses
protected	streamLicenses
protected	streamLock
protected	streamNames
protected	streamNamesLock

protected	streamNameToGroup
protected	streams

Constructor Summary

public	MediaStreamMap (IApplicationInstance appInstance) Create empty MediaStreamMap collection
--------	--

Method Summary

LicenseHolder	addLicense (ILiveStreamPacketizer liveStreamPacketizer, int licenseType)
LicenseHolder	addLicense (IMediaStream stream, int licenseType)
void	addMediaStreamListener (IMediaStreamNotify mediaStreamListener) Add a media stream listener.
MediaStreamMapGroup	addNameGroup (MediaStreamMapGroup newGroup)
void	broadcastPlayMessage (IMediaStream stream, long timecode, java.nio.ByteBuffer msg, int objectEncoding) Send a broadcast message to all play stream that are listening to this live published stream.
int	broadcastGetObjectEncoding (IMediaStream stream) Get the minimum object encoding level for the clients playing this stream.
void	clearStreamName (String name) Unregister a published live media stream name.
void	clearStreamName (String name, IMediaStream stream) Unregister a published live media stream name.
void	doWatchdog ()
IApplicationInstance	getAppInstance () Get the parent applicationInstance.
String	getAppInstanceName () Get the name of the parent applicationInstance.
String	getAppName () Get the name of the parent application.
int	getCount () Get the total number of streams stored in the mediaStreamMap
ILiveStreamDvrRecorder	getDvrRecorder (String streamName, String recorderName, boolean doCreate) Get a DVR recorder by name and recorder name
java.util.List	getDvrRecorders () Returns a list of ILiveStreamDvrRecorder objects

ILiveStreamPacketizer	getLiveStreamPacketizer (String streamName, String packetizerName, boolean doCreate) Get a live stream packetizer by name and packetizer id
Object	getLiveStreamPacketizerLock () Get the lock to the live stream packetizer system
MediaStreamMapGroup	getNameGroupByGroupName (String groupName)
int	getNameGroupRemoveDelayTime ()
java.util.Set	getNameGroups ()
java.util.Set	getNameGroups (String streamName)
java.util.Set	getNameGroupStreamNames (String streamName)
long	getNextNameGroupId (MediaStreamMapGroup newGroup)
int	getNextStreamIndex () Reserve a clientless stream id for a new media stream.
int	getNextStreamIndex (IClient client) Reserve a stream for a client connection.
int	getNextStreamIndex (com.wowza.wms.netconnection.INetConnection netConnection) Reserve a stream for a netConnection connection.
java.util.List	getPublishStreamNames () Returns a List of published stream names
IMediaStream	getStream (IClient client, int index) Get a media stream reference by stream id.
IMediaStream	getStream (IClient client, int index, boolean doCreate) Get a media stream reference by stream id.
IMediaStream	getStream (com.wowza.wms.netconnection.INetConnection netConnection, int index) Get a media stream object that is owned by a server to server netConnection object (not yet implemented).
IMediaStream	getStream (com.wowza.wms.netconnection.INetConnection netConnection, int index, boolean doCreate) Get a media stream reference by stream id.
IMediaStream	getStream (String name) Get a media stream by stream name.
IMediaStream	getStreamClientless (int index, String streamTypeStr) Get a media stream reference by stream id.
edu.emory.mathcs.backport.java.util.concurrent.locks.WMSReadWriteLock	getStreamListLock () Get the underlying read/write lock associated with the list of streams

edu.emory.mathcs.backport.java.util.concurrent.locks.WMSReadWriteLock	getStreamNameLock() Get the underlying read/write lock associated with the stream names list
java.util.List	getStreams() Returns a list of IMediaStream objects
IVHost	getVHost() Get the parent vHost.
void	notifyMediaStreamCreate(IMediaStream mediaStream) Notify all media stream listeners that a new media stream object has been created.
void	notifyMediaStreamDestroy(IMediaStream mediaStream) Notify all media stream listeners that a media stream object is being destroyed.
void	notifyPlayPublish(IMediaStream stream) Notify all play streams that are listening to this stream that the stream is going into a state of publish (NetStream.Play.PublishNotify).
void	notifyPlayUnpublish(IMediaStream stream) Notify all play streams that are listening to this stream that the stream is going into a state of unpublished (NetStream.Play.UnpublishNotify).
void	notifyPlayUnpublish(String streamName) Notify all play streams that are listening to this stream name that the stream is going into a state of unpublished (NetStream.Play.UnpublishNotify).
void	removeDvrRecorder(String streamName)
ILiveStreamDvrRecorder	removeDvrRecorder(String streamName, String recorderName) Remove DVR Recorder
void	removeLiveStreamPacketizer(String streamName) Remove all live stream packetizers for this stream name
ILiveStreamPacketizer	removeLiveStreamPacketizer(String streamName, String packetizerName) Remove live stream packetizer
void	removeMediaStreamListener(IMediaStreamNotify mediaStreamListener) Remove a media stream listener.
MediaStreamMapGroup	removeNameGroup(int groupId)
MediaStreamMapGroup	removeNameGroup(MediaStreamMapGroup nameGroup)
void	removeStream(IClient client, int index) Remove a stream associated with a client connection
void	removeStream(com.wowza.wms.netconnection.INetConnection netConnection, int index) Remove a stream associated with a netConnection object
void	removeStream(int index) Remove a clientless media stream
void	setNameGroupRemoveDelayTime(int nameGroupRemoveDelayTime)

void	setStreamName (IMediaStream stream, String name) Insert live media stream into the mediaStreamMap by name.
long	streamToIndex (IMediaStream stream) Get the unique stream identifier for a given stream

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

MAXSTREAMINDEX

```
public static final int MAXSTREAMINDEX
```

Constant value: **65536**

streams

```
protected java.util.Map streams
```

streamNames

```
protected java.util.Map streamNames
```

streamLicenses

```
protected java.util.Map streamLicenses
```

packetizerLicenses

```
protected java.util.Map packetizerLicenses
```

streamNameToGroup

```
protected java.util.Map streamNameToGroup
```

nameGroups

```
protected java.util.List nameGroups
```

nameGroupId

protected long **nameGroupId**

nameGroupRemoveDelayTime

protected int **nameGroupRemoveDelayTime**

nameGroupDelayRemoveMap

protected java.util.Map **nameGroupDelayRemoveMap**

streamLock

protected edu.emory.mathcs.backport.java.util.concurrent.locks.WMSReadWriteLock
streamLock

streamNamesLock

protected edu.emory.mathcs.backport.java.util.concurrent.locks.WMSReadWriteLock
streamNamesLock

appInstance

protected com.wowza.wms.application.IApplicationInstance **appInstance**

mediaStreamListeners

protected java.util.List **mediaStreamListeners**

nextStreamId

protected java.util.concurrent.atomic.AtomicLong **nextStreamId**

liveStreamPacketizers

protected java.util.Map **liveStreamPacketizers**

(continued from last page)

dvrRecorders

protected java.util.Map **dvrRecorders**

debugDVRShutdown

public boolean **debugDVRShutdown**

Constructors

MediaStreamMap

public **MediaStreamMap**([IApplicationInstance](#) appInstance)

Create empty MediaStreamMap collection

Parameters:

appInstance - parent applicationInstance

Methods

getStreamListLock

public edu.emory.mathcs.backport.java.util.concurrent.locks.WMSReadWriteLock
getStreamListLock()

Get the underlying read/write lock associated with the list of streams

Returns:

underlying read/write lock associated with the list of streams

getStreamNameLock

public edu.emory.mathcs.backport.java.util.concurrent.locks.WMSReadWriteLock
getStreamNameLock()

Get the underlying read/write lock associated with the stream names list

Returns:

underlying read/write lock associated with the stream names list

getPublishStreamNames

public java.util.List **getPublishStreamNames**()

Returns a List of published stream names

Returns:

List of published stream names

getStreams

public java.util.List **getStreams**()

(continued from last page)

Returns a list of IMediaStream objects

Returns:

list of IMediaStream objects

getStream

```
public IMediaStream getStream(IClient client,
    int index)
```

Get a media stream reference by stream id.

Parameters:

client - client
index - stream id

Returns:

media stream object or null if does not exist.

getStreamClientless

```
public IMediaStream getStreamClientless(int index,
    String streamTypeStr)
```

Get a media stream reference by stream id. If it does not exist, create a new one with the given id. This is a clientless stream that is served by the backend of the server. If creating a new media stream object, its id must first be obtained by calling [getNextStreamIndex\(\)](#).

Parameters:

index - stream id
streamTypeStr - stream type

Returns:

media stream object

getStream

```
public IMediaStream getStream(IClient client,
    int index,
    boolean doCreate)
```

Get a media stream reference by stream id. If it does not exist, create a new one with the given id. If creating a new media stream object, its id must first be obtained by calling [getNextStreamIndex\(IClient\)](#).

Parameters:

client - client
index - stream id
doCreate - true to create if it does not exist

Returns:

media stream object

getStream

```
public IMediaStream getStream(com.wowza.wms.netconnection.INetConnection
    netConnection,
    int index)
```

Get a media stream object that is owned by a server to server netConnection object (not yet implemented).

(continued from last page)

Parameters:

netConnection - netConnection to remote server
index - stream id

Returns:

media stream object or null if does not exist

getStream

```
public IMediaStream getStream(com.wowza.wms.netconnection.INetConnection  
netConnection,  
    int index,  
    boolean doCreate)
```

Get a media stream reference by stream id. If it does not exist, create a new one with the given id. If creating a new media stream object, its id must first be obtained by calling [getNextStreamIndex\(INetConnection\)](#) .

Parameters:

netConnection - netConnection to remote server
index - stream id
doCreate - true to create if it does not exist

Returns:

media stream object

getStream

```
public IMediaStream getStream(String name)
```

Get a media stream by stream name. Only published live streams are stored in the mediaStreamMap by name. This method is used to lookup a published live stream by name.

Parameters:

name - stream name

Returns:

media stream object or null if does not exist

setStreamName

```
public void setStreamName(IMediaStream stream,  
    String name)
```

Insert live media stream into the mediaStreamMap by name.

Parameters:

stream - media stream object
name - media stream name

clearStreamName

```
public void clearStreamName(String name)
```

Unregister a published live media stream name.

Parameters:

name - stream name

(continued from last page)

streamToIndex

```
public long streamToIndex(IMediaStream stream)
```

Get the unique stream identifier for a given stream

Parameters:

stream - stream

Returns:

unique stream identifier

clearStreamName

```
public void clearStreamName(String name,  
    IMediaStream stream)
```

Unregister a published live media stream name.

Parameters:

name - stream name

stream - stream

getNextStreamIndex

```
public int getNextStreamIndex(com.wowza.wms.netconnection.INetConnection  
netConnection)
```

Reserve a stream for a netConnection connection. Use [getStream\(INetConnection, int, boolean\)](#) to create stream.

Parameters:

netConnection

Returns:

next stream index

getNextStreamIndex

```
public int getNextStreamIndex(IClient client)
```

Reserve a stream for a client connection. Use [getStream\(IClient, int, boolean\)](#) to create stream.

Parameters:

client - parent client

Returns:

stream index

getNextStreamIndex

```
public int getNextStreamIndex( )
```

Reserve a clientless stream id for a new media stream. Use this method to obtain a stream id for a new media stream object that is then created with a call to `getClientlessStream()`.

Returns:

new reserved stream id

(continued from last page)

getCount

```
public int getCount()
```

Get the total number of streams stored in the mediaStreamMap

Returns:

total number of streams stored in the mediaStreamMap

removeStream

```
public void removeStream(com.wowza.wms.netconnection.INetConnection netConnection,  
    int index)
```

Remove a stream associated with a netConnection object

Parameters:

netConnection - net connection

index - stream index

removeStream

```
public void removeStream(IClient client,  
    int index)
```

Remove a stream associated with a client connection

Parameters:

client - client

index - stream index

removeStream

```
public void removeStream(int index)
```

Remove a clientless media stream

Parameters:

index - stream index

getAppName

```
public String getAppName()
```

Get the name of the parent application.

Returns:

parent application name.

getAppInstanceName

```
public String getAppInstanceName()
```

Get the name of the parent applicationInstance.

Returns:

parent applicationInstance name

(continued from last page)

getVHost

```
public IVHost getVHost()
```

Get the parent vHost.

Returns:

parent vHost

getAppInstance

```
public IApplicationInstance getAppInstance()
```

Get the parent applicationInstance.

Returns:

parent applicationInstance

notifyPlayUnpublish

```
public void notifyPlayUnpublish(IMediaStream stream)
```

Notify all play streams that are listening to this stream that the stream is going into a state of unpublished (NetStream.Play.UnpublishNotify).

Parameters:

stream - live published stream that is being unpublished

notifyPlayUnpublish

```
public void notifyPlayUnpublish(String streamName)
```

Notify all play streams that are listening to this stream name that the stream is going into a state of unpublished (NetStream.Play.UnpublishNotify).

Parameters:

streamName - live published stream that is being unpublished

broadcastGetObjectEncoding

```
public int broadcastGetObjectEncoding(IMediaStream stream)
```

Get the minimum object encoding level for the clients playing this stream.

Parameters:

stream - publish stream

Returns:

object encoding level (AMF0 or AMF3)

broadcasePlayMessage

```
public void broadcasePlayMessage(IMediaStream stream,  
    long timecode,  
    java.nio.ByteBuffer msg,  
    int objectEncoding)
```

Send a broadcast message to all play stream that are listening to this live published stream.

(continued from last page)

Parameters:

stream - media stream
timecode - timecode (milliseconds)
msg - byte[] of data that will be deserialized as an AMFData object.
objectEncoding - object encoding (AMF0 or AMF3)

notifyPlayPublish

```
public void notifyPlayPublish(IMediaStream stream)
```

Notify all play streams that are listening to this stream that the stream is going into a state of publish (NetStream.Play.PublishNotify).

Parameters:

stream - live published stream that is being published

addMediaStreamListener

```
public void addMediaStreamListener(IMediaStreamNotify mediaStreamListener)
```

Add a media stream listener. A media stream listener will receive the following events: onMediaStreamCreate, onMediaStreamDestroy.

Parameters:

mediaStreamListener - media stream listener

removeMediaStreamListener

```
public void removeMediaStreamListener(IMediaStreamNotify mediaStreamListener)
```

Remove a media stream listener.

Parameters:

mediaStreamListener - media stream listener

notifyMediaStreamCreate

```
public void notifyMediaStreamCreate(IMediaStream mediaStream)
```

Notify all media stream listeners that a new media stream object has been created.

Parameters:

mediaStream - new media stream object

notifyMediaStreamDestroy

```
public void notifyMediaStreamDestroy(IMediaStream mediaStream)
```

Notify all media stream listeners that a media stream object is being destroyed.

Parameters:

mediaStream - media stream object being destroyed

removeLiveStreamPacketizer

```
public void removeLiveStreamPacketizer(String streamName)
```

Remove all live stream packetizers for this stream name

(continued from last page)

Parameters:

streamName - streamName

removeLiveStreamPacketizer

```
public ILiveStreamPacketizer removeLiveStreamPacketizer(String streamName,  
String packetizerName)
```

Remove live stream packetizer

Parameters:

streamName - stream name

packetizerName - packetizer id

Returns:

live stream packetizer

getLiveStreamPacketizerLock

```
public Object getLiveStreamPacketizerLock()
```

Get the lock to the live stream packetizer system

Returns:

lock to the live stream packetizer system

getLiveStreamPacketizer

```
public ILiveStreamPacketizer getLiveStreamPacketizer(String streamName,  
String packetizerName,  
boolean doCreate)
```

Get a live stream packetizer by name and packetizer id

Parameters:

streamName - stream name

packetizerName - packetizer id

doCreate - create if does not exist

Returns:

live stream packetizer

getNextNameGroupId

```
public long getNextNameGroupId(MediaStreamMapGroup newGroup)
```

addNameGroup

```
public MediaStreamMapGroup addNameGroup(MediaStreamMapGroup newGroup)
```

removeNameGroup

```
public MediaStreamMapGroup removeNameGroup(int groupId)
```

removeNameGroup

```
public MediaStreamMapGroup removeNameGroup(MediaStreamMapGroup nameGroup)
```

getNameGroups

```
public java.util.Set getNameGroups()
```

getNameGroupByGroupName

```
public MediaStreamMapGroup getNameGroupByGroupName(String groupName)
```

getNameGroups

```
public java.util.Set getNameGroups(String streamName)
```

getNameGroupStreamNames

```
public java.util.Set getNameGroupStreamNames(String streamName)
```

getDvrRecorder

```
public ILiveStreamDvrRecorder getDvrRecorder(String streamName,  
      String recorderName,  
      boolean doCreate)
```

Get a DVR recorder by name and recorder name

Parameters:

streamName - stream name
recorderName - packetizer id
doCreate - create if does not exist

Returns:

dvr recorder

removeDvrRecorder

```
public void removeDvrRecorder(String streamName)
```

removeDvrRecorder

```
public ILiveStreamDvrRecorder removeDvrRecorder(String streamName,  
      String recorderName)
```

Remove DVR Recorder

(continued from last page)

Parameters:

streamName - stream name
recorderName - recorder name

Returns:

DVR Recorder

getDvrRecorders

```
public java.util.List getDvrRecorders()
```

Returns a list of `ILiveStreamDvrRecorder` objects

Returns:

list of `ILiveStreamDvrRecorder` objects

addLicense

```
public LicenseHolder addLicense(IMediaStream stream,  
int licenseType)
```

addLicense

```
public LicenseHolder addLicense(ILiveStreamPacketizer liveStreamPacketizer,  
int licenseType)
```

doWatchdog

```
public void doWatchdog()
```

getNameGroupRemoveDelayTime

```
public int getNameGroupRemoveDelayTime()
```

setNameGroupRemoveDelayTime

```
public void setNameGroupRemoveDelayTime(int nameGroupRemoveDelayTime)
```

Package

com.wowza.wms.stream.livedvr

com.wowza.wms.stream.livedvr Interface IDvrStreamManagerActionNotify

All Subinterfaces:

[IDvrStreamManagerActionNotify2](#)

public interface **IDvrStreamManagerActionNotify**
extends

IDvrStreamManagerActionNotify notify interface for dvr stores. See
IApplicationInstance.addDvrStreamManagerListener(IDvrStreamManagerActionNotify listener)

Method Summary

void	onDvrStreamManagerCreate (IDvrStreamManager dvrMgr) Called when DVR stream manager created but before it is initialized.
void	onDvrStreamManagerDestroy (IDvrStreamManager dvrMgr) Called when DVR stream manager destroyed.
void	onDvrStreamManagerInit (IDvrStreamManager dvrMgr) Called when DVR stream manager initialized.

Methods

onDvrStreamManagerCreate

public void **onDvrStreamManagerCreate**([IDvrStreamManager](#) dvrMgr)

Called when DVR stream manager created but before it is initialized.

Parameters:

dvrMgr - DVR stream manager

onDvrStreamManagerInit

public void **onDvrStreamManagerInit**([IDvrStreamManager](#) dvrMgr)

Called when DVR stream manager initialized.

Parameters:

dvrMgr - DVR stream manager

onDvrStreamManagerDestroy

public void **onDvrStreamManagerDestroy**([IDvrStreamManager](#) dvrMgr)

Called when DVR stream manager destroyed.

Parameters:

dvrMgr - DVR stream manager

com.wowza.wms.stream.livedvr Interface IDvrStreamManagerActionNotify2

All Superinterfaces:

[IDvrStreamManagerActionNotify](#)

public interface **IDvrStreamManagerActionNotify2**

extends [IDvrStreamManagerActionNotify](#)

IDvrStreamManagerActionNotify notify interface for dvr stores. See
IApplicationInstance.addDvrStreamManagerListener(IDvrStreamManagerActionNotify listener)

Method Summary

void	onDvrStreamManagerCreate (IDvrStreamManager dvrMgr) Called when DVR stream manager created but before it is initialized.
void	onDvrStreamManagerDestroy (IDvrStreamManager dvrMgr) Called when DVR stream manager destroyed.
void	onDvrStreamManagerInit (IDvrStreamManager dvrMgr) Called when DVR stream manager initialized.
void	onDvrStreamManagerInitStreamName (IDvrStreamManager dvrManager, String streamName)

Methods inherited from interface [com.wowza.wms.stream.livedvr.IDvrStreamManagerActionNotify](#)

[onDvrStreamManagerCreate](#), [onDvrStreamManagerDestroy](#), [onDvrStreamManagerInit](#)

Methods

onDvrStreamManagerCreate

public void **onDvrStreamManagerCreate**([IDvrStreamManager](#) dvrMgr)

Called when DVR stream manager created but before it is initialized.

Parameters:

dvrMgr - DVR stream manager

onDvrStreamManagerInit

public void **onDvrStreamManagerInit**([IDvrStreamManager](#) dvrMgr)

Called when DVR stream manager initialized.

Parameters:

dvrMgr - DVR stream manager

(continued from last page)

onDvrStreamManagerDestroy

```
public void onDvrStreamManagerDestroy(IDvrStreamManager dvrMgr)
```

Called when DVR stream manager destroyed.

Parameters:

dvrMgr - DVR stream manager

onDvrStreamManagerInitStreamName

```
public void onDvrStreamManagerInitStreamName(IDvrStreamManager dvrManager,  
String streamName)
```

com.wowza.wms.stream.livedvr Interface **ILiveStreamDvrRecorder**

public interface **ILiveStreamDvrRecorder**
extends

ILiveStreamDvrRecorder: DVR Recorder interface.

Method Summary

boolean	<u>canRecordAudio()</u> True if audio is being recorded
boolean	<u>canRecordData()</u> True if data is being recorded
boolean	<u>canRecordVideo()</u> True if video is being recorded
<u>IMediaStream</u>	<u>getAndSetStartStream()</u> (<u>IMediaStream</u> stream)
<u>IApplicationInstance</u>	<u>getAppInstance()</u> Get associated applicationInstance.
<u>IDvrStreamManager</u>	<u>getDvrManager()</u> Get DVR recorder's associated DVR stream manager
int	<u>getDvrRecorderId()</u> Get the DVR recorder id
LicenseHolder	<u>getLicenseHolder()</u>
<u>WMSProperties</u>	<u>getProperties()</u> Get properties
String	<u>getRecordingName()</u>
<u>IMediaStream</u>	<u>getStream()</u> Get the current stream that is being recorded
void	<u>handlePacket()</u> (<u>IMediaStream</u> stream, <u>AMFPacket</u> packet) Called to handle an incoming packet
void	<u>init()</u> (String streamName, String recorderName, <u>IApplicationInstance</u> appInstance, DvrRecorderItem dvrRecorderItem) Initialize DVR recorder.
boolean	<u>isActive()</u> Is the DVR recorder active
boolean	<u>isRecording()</u> Is this stream currently recording.

boolean	<u>isRecordingPaused()</u> Is this stream currently paused from recording.
boolean	<u>pauseRecording()</u> Request that stream recording pause.
void	<u>resetStream()</u> (<u>IMediaStream</u> stream) Called when something happens that forces the stream to reset
boolean	<u>resumeRecording()</u> Request that stream recording resume.
void	<u>setDvrRecorderId()</u> (int liveStreamId) Set the DVR recorder id
void	<u>setRecordAudio()</u> (boolean recordAudio) Set to true to record audio
void	<u>setRecordData()</u> (boolean recordVideo) Set to true to record data
void	<u>setRecordingName()</u> (String name)
void	<u>setRecordVideo()</u> (boolean recordVideo) Set to true to record video
void	<u>setStartRecordingOnStartup()</u> (boolean shouldStartRecordingOnStartup) Set recording behavior of DVR Manager on startup.
boolean	<u>shouldStartRecordingOnStartup()</u> Should DVR start recording when packets start flowing.
void	<u>shutdown()</u> Called to shutdown the DVR recorder
<u>IDvrStreamManager</u>	<u>startRecording()</u> Request that stream recording start.
void	<u>startStream()</u> (<u>IMediaStream</u> stream) Called when the stream starts
boolean	<u>stopRecording()</u> Request that stream recording stop.
void	<u>touch()</u> (long timecode) Touch the stream to keep it active

Methods

init

```
public void init(String streamName,
    String recorderName,
    IApplicationInstance appInstance,
    DvrRecorderItem dvrRecorderItem)
```

Initialize DVR recorder.

(continued from last page)

Parameters:

streamName - stream name
recorderName - DVR recorder name
appInstance - application instance
dvrRecorderItem - DVR recorder

shutdown

```
public void shutdown()
```

Called to shutdown the DVR recorder

getStream

```
public IMediaStream getStream()
```

Get the current stream that is being recorded

Returns:

stream

getAppInstance

```
public IApplicationInstance getAppInstance()
```

Get associated applicationInstance.

Returns:

application Instance

handlePacket

```
public void handlePacket(IMediaStream stream,  
    AMFPacket packet)
```

Called to handle an incoming packet

Parameters:

stream - stream
packet - packet

startStream

```
public void startStream(IMediaStream stream)
```

Called when the stream starts

Parameters:

stream - stream

resetStream

```
public void resetStream(IMediaStream stream)
```

Called when something happens that forces the stream to reset

Parameters:

stream - stream

touch

```
public void touch(long timecode)
```

Touch the stream to keep it active

Parameters:

timecode - timecode of touch in milliseconds

isActive

```
public boolean isActive()
```

Is the DVR recorder active

Returns:

true if active

getProperties

```
public WMSProperties getProperties()
```

Get properties

Returns:

properties

getDvrRecorderId

```
public int getDvrRecorderId()
```

Get the DVR recorder id

Returns:

DVR recorder id

setDvrRecorderId

```
public void setDvrRecorderId(int liveStreamId)
```

Set the DVR recorder id

Parameters:

liveStreamId

getDvrManager

```
public IDvrStreamManager getDvrManager()
```

Get DVR recorder's associated DVR stream manager

Returns:

DVR manager

canRecordAudio

```
public boolean canRecordAudio()
```

(continued from last page)

True if audio is being recorded

Returns:

True if audio is being recorded

setRecordAudio

```
public void setRecordAudio(boolean recordAudio)
```

Set to true to record audio

Parameters:

recordAudio - true to record audio

canRecordVideo

```
public boolean canRecordVideo( )
```

True if video is being recorded

Returns:

True if video is being recorded

setRecordVideo

```
public void setRecordVideo(boolean recordVideo)
```

Set to true to record video

Parameters:

recordVideo - true to record video

canRecordData

```
public boolean canRecordData( )
```

True if data is being recorded

Returns:

True if data is being recorded

setRecordData

```
public void setRecordData(boolean recordVideo)
```

Set to true to record data

Parameters:

recordVideo - true to record data

startRecording

```
public IDvrStreamManager startRecording( )
```

Request that stream recording start.

Returns:

stream manager if successful. null otherwise.

isRecording

```
public boolean isRecording()
```

Is this stream currently recording. If this stream is not recordable, the method returns false.

Returns:

true if has a recording stream and it is recording.

isRecordingPaused

```
public boolean isRecordingPaused()
```

Is this stream currently paused from recording. If this stream is not recording and not paused, the method returns false.

Returns:

true if has a recording is paused.

stopRecording

```
public boolean stopRecording()
```

Request that stream recording stop. Note that this call places the DVR stream in the *not* recording state.

A successful stop will result in registered [IDvrRecordingListeners](#) to have their [IDvrRecordingListener.recordingStopped\(IDvrStreamStore\)](#) method called.

Success only occurs if the stream is already in the recording state [IDvrStreamStore.isRecording\(\)](#).

Returns:

store if successful. null otherwise.

pauseRecording

```
public boolean pauseRecording()
```

Request that stream recording pause. The stream does not have to be actively recording to be paused. For example, it could be paused before packets start flowing.

A successful pause will result in registered [IDvrRecordingListeners](#) to have their [IDvrRecordingListener.recordingPaused\(IDvrStreamStore\)](#) method called.

Returns:

store if successful. null otherwise.

resumeRecording

```
public boolean resumeRecording()
```

Request that stream recording resume. The stream does not have to be actively recording to be resumed. For example, it could have been paused before the stream started, and this call would move it out of the paused state.

A successful resume will result in registered [IDvrRecordingListeners](#) to have their [IDvrRecordingListener.recordingResumed\(IDvrStreamStore\)](#) method called.

Returns:

store if successful. null otherwise.

(continued from last page)

setRecordingName

```
public void setRecordingName(String name)
```

getRecordingName

```
public String getRecordingName()
```

shouldStartRecordingOnStartup

```
public boolean shouldStartRecordingOnStartup()
```

Should DVR start recording when packets start flowing.

Returns:

true if should start recording initially, false otherwise

setStartRecordingOnStartup

```
public void setStartRecordingOnStartup(boolean shouldStartRecordingOnStartup)
```

Set recording behavior of DVR Manager on startup.

Parameters:

shouldStartRecordingOnStartup - should recording start when DVR manager starts.

getLicenseHolder

```
public LicenseHolder getLicenseHolder()
```

getAndSetStartStream

```
public IMediaStream getAndSetStartStream(IMediaStream stream)
```

com.wowza.wms.stream.livedvr Interface **ILiveStreamDvrRecorderActionNotify**

public interface **ILiveStreamDvrRecorderActionNotify**
extends

ILiveStreamDvrRecorderActionNotify: notify interface for dvr recorders.

See Also:

[IApplicationInstance.addDvrRecorderListener\(ILiveStreamDvrRecorderActionNotify\)](#),

Method Summary

void	onLiveStreamDvrRecorderCreate (ILiveStreamDvrRecorder recorder, String streamName) Called when recorder created
void	onLiveStreamDvrRecorderDestroy (ILiveStreamDvrRecorder recorder) Called when recorder destroyed
void	onLiveStreamDvrRecorderInit (ILiveStreamDvrRecorder recorder, String streamName) Called after recorder is initialized

Methods

onLiveStreamDvrRecorderCreate

```
public void onLiveStreamDvrRecorderCreate(ILiveStreamDvrRecorder recorder,  
String streamName)
```

Called when recorder created

Parameters:

recorder - recorder
streamName - stream name

onLiveStreamDvrRecorderInit

```
public void onLiveStreamDvrRecorderInit(ILiveStreamDvrRecorder recorder,  
String streamName)
```

Called after recorder is initialized

Parameters:

recorder - recorder
streamName - stream name

onLiveStreamDvrRecorderDestroy

```
public void onLiveStreamDvrRecorderDestroy(ILiveStreamDvrRecorder recorder)
```

Called when recorder destroyed

(continued from last page)

Parameters:

recorder - recorder

com.wowza.wms.stream.livedvr Interface ILiveStreamDvrRecorderControl

public interface **ILiveStreamDvrRecorderControl**
extends

ILiveStreamDvrRecorderControl: interface to control which recorders are run for which streams.
See Also:
[IApplicationInstance.setLiveStreamDvrRecorderControl\(ILiveStreamDvrRecorderControl\)](#)

Method Summary	
boolean	shouldDvrRecord (String recorderName, IMediaStream stream) Returns true if given string should be recorded.;

Methods

shouldDvrRecord

public boolean **shouldDvrRecord**(String recorderName, [IMediaStream](#) stream)

Returns true if given string should be recorded.;

Parameters:

recorderName - recorder name
stream - stream

Returns:

true to record

Package

com.wowza.wms.stream.livepacketizer

com.wowza.wms.stream.livepacketizer

Interface **ILiveStreamPacketizer**

All Subinterfaces:

[IDvrStreamManager](#)

public interface **ILiveStreamPacketizer**

extends

ILiveStreamPacketizer: live stream packetizer interface.

Method Summary

IMediaStream	getAndSetStartStream (IMediaStream stream) Get the current stream that is being packetized
IApplicationInstance	getApplicationInstance () Get the application instance associated with this live stream packetizer.
int	getLiveStreamPacketizerId () Get the live stream packetizer id
WMSProperties	getProperties () Get properties
long	getRepeaterLastSequence () Get the sequence number of the last added repeater item
void	handlePacket (IMediaStream stream, AMFPacket packet) Called to handle an incoming packet
void	init (String streamName, String packetizerName, IApplicationInstance appInstance, LiveStreamPacketizerItem liveStreamPacketizerItem) Initialize live stream packetizer
boolean	isActive () Is the live stream packetizer active
boolean	isPacketizeAudio () True if audio is being packetized
boolean	isPacketizeData () True if data is being packetized
boolean	isPacketizeVideo () True if video is being packetized
boolean	isRepeaterEdge () Is this packetizer a live repeater edge
void	resetStream (IMediaStream stream) Called when something happens that forces the stream to reset
void	setLiveStreamPacketizerId (int id) Set the live stream packetizer id

void	<u>setPacketizeAudio</u> (boolean packetizeAudio) Set to true to packetize audio
void	<u>setPacketizeData</u> (boolean packetizeVideo) Set to true to packetize data
void	<u>setPacketizeVideo</u> (boolean packetizeVideo) Set to true to packetize video
void	<u>setRepeaterEdge</u> (boolean isRepeaterEdge) Set is live repeater edge
void	<u>shutdown</u> () Called to shutdown the live stream packetizer
void	<u>startStream</u> (<u>IMediaStream</u> stream) Called when the stream starts
void	<u>touch</u> (long timecode) Touch the stream to keep it active

Methods

init

```
public void init(String streamName,
String packetizerName,
IApplicationInstance appInstance,
LiveStreamPacketizerItem liveStreamPacketizerItem)
```

Initialize live stream packetizer

Parameters:

streamName - stream name
packetizerName - packetizer name
appInstance - application instance
liveStreamPacketizerItem - live stream packetizer

shutdown

```
public void shutdown( )
```

Called to shutdown the live stream packetizer

isActive

```
public boolean isActive( )
```

Is the live stream packetizer active

Returns:

true if active

getProperties

```
public WMSProperties getProperties( )
```


(continued from last page)

Get properties

Returns:
properties

getRepeaterLastSequence

```
public long getRepeaterLastSequence()
```

Get the sequence number of the last added repeater item

Returns:
sequence number

getLiveStreamPacketizerId

```
public int getLiveStreamPacketizerId()
```

Get the live stream packetizer id

Returns:
live stream packetizer id

setLiveStreamPacketizerId

```
public void setLiveStreamPacketizerId(int id)
```

Set the live stream packetizer id

Parameters:
id

handlePacket

```
public void handlePacket(IMediaStream stream,  
    AMFPacket packet)
```

Called to handle an incoming packet

Parameters:
stream - stream
packet - packet

startStream

```
public void startStream(IMediaStream stream)
```

Called when the stream starts

Parameters:
stream - stream

resetStream

```
public void resetStream(IMediaStream stream)
```

Called when something happens that forces the stream to reset

Parameters:

(continued from last page)

stream - stream

touch

```
public void touch(long timecode)
```

Touch the stream to keep it active

Parameters:

timecode - timecode of touch in milliseconds

isRepeaterEdge

```
public boolean isRepeaterEdge()
```

Is this packetizer a live repeater edge

Returns:

true if live repeater edge

setRepeaterEdge

```
public void setRepeaterEdge(boolean isRepeaterEdge)
```

Set is live repeater edge

Parameters:

isRepeaterEdge - is live repeater edge

isPacketizeAudio

```
public boolean isPacketizeAudio()
```

True if audio is being packetized

Returns:

True if audio is being packetized

setPacketizeAudio

```
public void setPacketizeAudio(boolean packetizeAudio)
```

Set to true to packetize audio

Parameters:

packetizeAudio - true to packetize audio

isPacketizeVideo

```
public boolean isPacketizeVideo()
```

True if video is being packetized

Returns:

True if video is being packetized

(continued from last page)

setPacketizeVideo

```
public void setPacketizeVideo(boolean packetizeVideo)
```

Set to true to packetize video

Parameters:

packetizeVideo - true to packetize video

isPacketizeData

```
public boolean isPacketizeData()
```

True if data is being packetized

Returns:

True if data is being packetized

setPacketizeData

```
public void setPacketizeData(boolean packetizeVideo)
```

Set to true to packetize data

Parameters:

packetizeVideo - true to packetize data

getAndSetStartStream

```
public IMediaStream getAndSetStartStream(IMediaStream stream)
```

Get the current stream that is being packetized

Returns:

stream

getApplicationInstance

```
public IApplicationInstance getApplicationInstance()
```

Get the application instance associated with this live stream packetizer.

Returns:

application instance

com.wowza.wms.stream.livepacketizer Interface **ILiveStreamPacketizerActionNotify**

public interface **ILiveStreamPacketizerActionNotify**
extends

ILiveStreamPacketizerActionNotify: notify interface for live stream packetizers. See
IApplicationInstance.addLiveStreamPacketizerListener(ILiveStreamPacketizerActionNotify liveStreamPacketizerListener)

Method Summary

void	onLiveStreamPacketizerCreate (ILiveStreamPacketizer liveStreamPacketizer, String streamName) Called when packetizer created
void	onLiveStreamPacketizerDestroy (ILiveStreamPacketizer liveStreamPacketizer) Called when packetizer destroyed
void	onLiveStreamPacketizerInit (ILiveStreamPacketizer liveStreamPacketizer, String streamName) Called after packetizer is initialized

Methods

onLiveStreamPacketizerCreate

```
public void onLiveStreamPacketizerCreate(ILiveStreamPacketizer liveStreamPacketizer,  
String streamName)
```

Called when packetizer created

Parameters:

liveStreamPacketizer - packetizer
streamName - stream name

onLiveStreamPacketizerDestroy

```
public void onLiveStreamPacketizerDestroy(ILiveStreamPacketizer liveStreamPacketizer)
```

Called when packetizer destroyed

Parameters:

liveStreamPacketizer - packetizer

onLiveStreamPacketizerInit

```
public void onLiveStreamPacketizerInit(ILiveStreamPacketizer liveStreamPacketizer,  
String streamName)
```

Called after packetizer is initialized

Parameters:

(continued from last page)

liveStreamPacketizer - packetizer
streamName - stream name

com.wowza.wms.stream.livepacketizer Interface ILiveStreamPacketizerControl

public interface **ILiveStreamPacketizerControl**
extends

ILiveStreamPacketizerControl: interface to control which packetizers are run for which streams. See:
IApplicationInstance.setLiveStreamPacketizerControl(ILiveStreamPacketizerControl liveStreamPacketizerControl)

Method Summary

boolean	isLiveStreamPacketize (String packetizer, IMediaStream stream) Return true to packetize stream
---------	---

Methods

isLiveStreamPacketize

public boolean **isLiveStreamPacketize**(String packetizer,
[IMediaStream](#) stream)

Return true to packetize stream

Parameters:

packetizer - packetizer name
stream - stream

Returns:

true to packetize

Package

com.wowza.wms.stream.livetranscoder

com.wowza.wms.stream.livetranscoder

Interface ILiveStreamTranscoder

public interface **ILiveStreamTranscoder**
 extends

ILiveStreamTranscoder: Interface to live stream transcoder.

Field Summary

public static final	TRANSCODERTYPE_LIVE Value: 1
public static final	TRANSCODERTYPE_VOD Value: 2

Method Summary

void	close (IMediaStream stream) Called when live stream transcoder is stream is closed
IApplicationInstance	getAppInstance () Get the application instance associated with this live stream transcoder.
String	getContextStr () Get the streaming context for this live stream transcoder.
LicenseHolder	getLicenseHolder () Get license holder.
LiveStreamTranscoderItem	getLiveStreamTranscoderItem () Get the definition for live stream transcoder.
WMSProperties	getProperties () Get the user properties
java.util.List	getPublishers () Get a list of publishers for this transcoder (resultant streams)
String	getStreamName () Get the stream name of the source stream.
String	getTranscoderName () Get the live stream transcoder name
int	getTranscoderType () Get the live stream transcoder type.
void	handleOnMetadata (IMediaStream stream, AMFPacket packet, long timecode, boolean isSetDataFrame) Called for each new onMetaData packet

void	<u>handlePacket</u> (<u>IMediaStream</u> stream, <u>AMFPacket</u> packet) Called for each new source packet
void	<u>init</u> (String streamName, <u>IMediaStream</u> stream, String transcoderName, <u>IApplicationInstance</u> appInstance, <u>LiveStreamTranscoderItem</u> liveStreamTranscoderItem) Called when live stream transcoder interface is initialized.
boolean	<u>isTemplateLoaded</u> () Is the transcoder template loaded.
boolean	<u>isTranscoderActive</u> (long currTime) Returns true if the transcoder is actively receiving packets
void	<u>resetStream</u> (<u>IMediaStream</u> stream) Called when source stream changes.
void	<u>setAppInstance</u> (<u>IApplicationInstance</u> appInstance) Set the application instance associated with this live stream transcoder.
void	<u>setLiveStreamTranscoderItem</u> (<u>LiveStreamTranscoderItem</u> liveStreamTranscoderItem) Set the definition for live stream transcoder.
void	<u>setStreamName</u> (String streamName) Set source stream name.
void	<u>setTranscoderName</u> (String transcoderName) Get the live stream transcoder name
void	<u>setTranscoderType</u> (int transcoderType) Set the live stream transcoder type.
void	<u>shutdown</u> (<u>IMediaStream</u> stream) Called when live stream transcoder is shutdown

Fields

TRANSCODERTYPE_LIVE

public static final int **TRANSCODERTYPE_LIVE**

Constant value: **1**

TRANSCODERTYPE_VOD

public static final int **TRANSCODERTYPE_VOD**

Constant value: **2**

Methods

(continued from last page)

init

```
public void init(String streamName,  
    IMediaStream stream,  
    String transcoderName,  
    IApplicationInstance appInstance,  
    LiveStreamTranscoderItem liveStreamTranscoderItem)
```

Called when live stream transcoder interface is initialized.

Parameters:

streamName - stream name
transcoderName - transcoder name
appInstance - application instance
liveStreamTranscoderItem - live stream transcoder config item

handlePacket

```
public void handlePacket(IMediaStream stream,  
    AMFPacket packet)
```

Called for each new source packet

Parameters:

stream - stream
packet - packet

handleOnMetadata

```
public void handleOnMetadata(IMediaStream stream,  
    AMFPacket packet,  
    long timecode,  
    boolean isSetDataFrame)
```

Called for each new onMetaData packet

Parameters:

stream - stream
packet - packet
timecode - timecode (milliseconds)
isSetDataFrame - is SetDataFrame call

resetStream

```
public void resetStream(IMediaStream stream)
```

Called when source stream changes.

Parameters:

stream - stream interface

close

```
public void close(IMediaStream stream)
```

Called when live stream transcoder is stream is closed

Parameters:

stream - stream interface

shutdown

```
public void shutdown(IMediaStream stream)
```

Called when live stream transcoder is shutdown

Parameters:

stream - stream interface

getStreamName

```
public String getStreamName()
```

Get the stream name of the source stream.

Returns:

stream name of the source stream

setStreamName

```
public void setStreamName(String streamName)
```

Set source stream name.

Parameters:

streamName - source stream name

getTranscoderName

```
public String getTranscoderName()
```

Get the live stream transcoder name

Returns:

live stream transcoder name

setTranscoderName

```
public void setTranscoderName(String transcoderName)
```

Get the live stream transcoder name

Parameters:

transcoderName - live stream transcoder name

getAppInstance

```
public IApplicationInstance getAppInstance()
```

Get the application instance associated with this live stream transcoder.

Returns:

application instance associated with this live stream transcoder

setAppInstance

```
public void setAppInstance(IApplicationInstance appInstance)
```

(continued from last page)

Set the application instance associated with this live stream transcoder.

Parameters:

appInstance - application instance associated with this live stream transcoder

getLiveStreamTranscoderItem

```
public LiveStreamTranscoderItem getLiveStreamTranscoderItem()
```

Get the definition for live stream transcoder.

Returns:

definition for live stream transcoder

setLiveStreamTranscoderItem

```
public void setLiveStreamTranscoderItem(LiveStreamTranscoderItem  
liveStreamTranscoderItem)
```

Set the definition for live stream transcoder.

Parameters:

liveStreamTranscoderItem - definition for live stream transcoder

getProperties

```
public WMSProperties getProperties()
```

Get the user properties

Returns:

user properties

getLicenseHolder

```
public LicenseHolder getLicenseHolder()
```

Get license holder.

Returns:

license holder

isTranscoderActive

```
public boolean isTranscoderActive(long currTime)
```

Returns true if the transcoder is actively receiving packets

Parameters:

currTime - current timecode in milliseconds

Returns:

true if the transcoder is actively receiving packets

isTemplateLoaded

```
public boolean isTemplateLoaded()
```

(continued from last page)

Is the transcoder template loaded.

Returns:

true if transcoder template is loaded.

getContextStr

```
public String getContextStr()
```

Get the streaming context for this live stream transcoder.

Returns:

streaming context for this live stream transcoder

getTranscoderType

```
public int getTranscoderType()
```

Get the live stream transcoder type. See `ILiveStreamTranscoder.TRANSCODERTYPE_*`

Returns:

live stream transcoder type

setTranscoderType

```
public void setTranscoderType(int transcoderType)
```

Set the live stream transcoder type. See `ILiveStreamTranscoder.TRANSCODERTYPE_*`

Parameters:

transcoderType - live stream transcoder type

getPublishers

```
public java.util.List getPublishers()
```

Get a list of publishers for this transcoder (resultant streams)

Returns:

list of publishers for this transcoder

com.wowza.wms.stream.livetranscoder Interface **ILiveStreamTranscoderControl**

public interface **ILiveStreamTranscoderControl**
extends

Interface used to control if stream is transcoded. See
`IApplicationInstance.setLiveStreamTranscoderControl(ILiveStreamTranscoderControl)`

Method Summary

boolean	isLiveStreamTranscode (String transcoder, IMediaStream stream) Called each time a new publishing stream is started.
---------	--

Methods

isLiveStreamTranscode

```
public boolean isLiveStreamTranscode(String transcoder,  
    IMediaStream stream)
```

Called each time a new publishing stream is started. Return true if you wish stream to be transcoded. See
`IApplicationInstance.setLiveStreamTranscoderControl(ILiveStreamTranscoderControl)`

Parameters:

transcoder - name of transcoder
stream - stream interface

Returns:

true if wish stream to be transcoded

com.wowza.wms.stream.livetranscoder Interface `ILiveStreamTranscoderNotify`

public interface `ILiveStreamTranscoderNotify`
extends

`ILiveStreamTranscoderNotify`: Listener interface for listening for new live stream transcoders. See `IApplicationInstance.addLiveStreamTranscoderListener(ILiveStreamTranscoderNotify)`

Method Summary

void	<code>onLiveStreamTranscoderCreate</code> (<code>ILiveStreamTranscoder</code> liveStreamTranscoder, <code>IMediaStream</code> stream) Triggered when live stream transcoder is created.
void	<code>onLiveStreamTranscoderDestroy</code> (<code>ILiveStreamTranscoder</code> liveStreamTranscoder, <code>IMediaStream</code> stream) Triggered when live stream transcoder is destroyed.
void	<code>onLiveStreamTranscoderInit</code> (<code>ILiveStreamTranscoder</code> liveStreamTranscoder, <code>IMediaStream</code> stream) Triggered after live stream transcoder is initialized.

Methods

`onLiveStreamTranscoderCreate`

```
public void onLiveStreamTranscoderCreate(ILiveStreamTranscoder liveStreamTranscoder,  
    IMediaStream stream)
```

Triggered when live stream transcoder is created.

Parameters:

liveStreamTranscoder - live stream transcoder
stream - source stream

`onLiveStreamTranscoderDestroy`

```
public void onLiveStreamTranscoderDestroy(ILiveStreamTranscoder liveStreamTranscoder,  
    IMediaStream stream)
```

Triggered when live stream transcoder is destroyed.

Parameters:

liveStreamTranscoder - live stream transcoder
stream - source stream

`onLiveStreamTranscoderInit`

```
public void onLiveStreamTranscoderInit(ILiveStreamTranscoder liveStreamTranscoder,  
    IMediaStream stream)
```

Triggered after live stream transcoder is initialized.

(continued from last page)

Parameters:

liveStreamTranscoder - live stream transcoder

stream - source stream

Package

com.wowza.wms.stream.publish

com.wowza.wms.stream.publish Interface IPublishingProvider

All Known Implementing Classes:

[PublishingProviderMediaReader](#), [PublishingProviderLive](#)

public interface **IPublishingProvider**
extends

IPublishingProvider: publishing provider interface.

Method Summary

void	close() Invoked on stream close
boolean	isSendOnMetadata() Get to send onMetadata event when stream starts
boolean	play(Publisher publisher) Invoked on play
boolean	seek(long timecode) Invoked on seek
boolean	seek(long timecode, int seekType) Invoked on seek
void	setDuration(long duration) Set target duration for playback (milliseconds)
void	setRealTimeStartTime(long realTimeStartTime) Set real start time (milliseconds)
void	setSendOnMetadata(boolean sendOnMetadata) Set to send onMetadata event when stream starts

Methods

play

public boolean **play**([Publisher](#) publisher)

Invoked on play

Parameters:

publisher - publisher

Returns:

true if successful

(continued from last page)

close

```
public void close()
```

Invoked on stream close

seek

```
public boolean seek(long timecode,  
                    int seekType)
```

Invoked on seek

Parameters:

timecode - target timecode

seekType - seek type, see IMediaReader.SEEKTARGET_*

Returns:

true, if seek successful

seek

```
public boolean seek(long timecode)
```

Invoked on seek

Parameters:

timecode - target timecode

Returns:

true, if seek successful

setDuration

```
public void setDuration(long duration)
```

Set target duration for playback (milliseconds)

Parameters:

duration - duration for playback (milliseconds)

setRealTimeStartTime

```
public void setRealTimeStartTime(long realTimeStartTime)
```

Set real start time (milliseconds)

Parameters:

realTimeStartTime - real start time (milliseconds)

setSendOnMetadata

```
public void setSendOnMetadata(boolean sendOnMetadata)
```

Set to send onMetadata event when stream starts

Parameters:

sendOnMetadata - true to send onMetadata event

isSendOnMetadata

```
public boolean isSendOnMetadata()
```

Get to send onMetadata event when stream starts

Returns:

true to send onMetadata event

com.wowza.wms.stream.publish Interface IStreamActionNotify

public interface **IStreamActionNotify**
extends

IStreamActionNotify: listener interface to Stream class for playlist items. See Stream.addListener(IStreamActionNotify listener)

Method Summary

void	onPlaylistItemStart (Stream stream, PlaylistItem playlistItem) Invoked when playlist item playback is started
void	onPlaylistItemStop (Stream stream, PlaylistItem playlistItem) Invoked when playlist item playback has ended

Methods

onPlaylistItemStart

```
public void onPlaylistItemStart(Stream stream,  
    PlaylistItem playlistItem)
```

Invoked when playlist item playback is started

Parameters:

stream - stream

playlistItem - playlist item

onPlaylistItemStop

```
public void onPlaylistItemStop(Stream stream,  
    PlaylistItem playlistItem)
```

Invoked when playlist item playback has ended

Parameters:

stream - stream

playlistItem - playlist item

com.wowza.wms.stream.publish Class Playlist

java.lang.Object

└─com.wowza.wms.stream.publish.Playlist

public class **Playlist**
extends Object

Constructor Summary

public	<code>Playlist(String sName)</code> Class constructor - A simple structure that maintains a list of playlist items.
--------	--

Method Summary

void	<code>addItem(String sName, int start, int length)</code> Appends an item to this playlist
java.util.List	<code>getItems()</code> Get the items in the playlist (returns a copy of the list)
String	<code>getName()</code> Returns the name of this playlist as defined in the XML definition file
boolean	<code>getRepeat()</code>
boolean	<code>open(Stream s)</code> Opens this playlist on the given stream...
void	<code>removeItem(int index)</code> Remove an item from the list
void	<code>setRepeat(boolean repeat)</code>

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

Playlist

public **Playlist**(String sName)

Class constructor - A simple structure that maintains a list of playlist items.

Parameters:

sName -- name of playlist - as defined in the XML definition file

Methods

getItems

```
public java.util.List getItems()
```

Get the items in the playlist (returns a copy of the list)

Returns:

items in the playlist (returns a copy of the list)

getRepeat

```
public boolean getRepeat()
```

setRepeat

```
public void setRepeat(boolean repeat)
```

addItem

```
public void addItem(String sName,  
                    int start,  
                    int length)
```

Appends an item to this playlist

Parameters:

sName -- item name

start -- start seconds

length -- playlength seconds

removeItem

```
public void removeItem(int index)
```

Remove an item from the list

Parameters:

index - index of item to remove

open

```
public boolean open(Stream s)
```

Opens this playlist on the given stream... stopping anything currently playing on that stream and switching over.

Parameters:

s -- stream to play on

Returns:

- true if successful

getName

```
public String getName()
```

Returns the name of this playlist as defined in the XML definition file

Returns:

name of this playlist

com.wowza.wms.stream.publish Class PlaylistItem

java.lang.Object

└─com.wowza.wms.stream.publish.PlaylistItem

public class **PlaylistItem**
extends Object

Constructor Summary

public	PlaylistItem (String name, int start, int length, int index) Class constructor - A simple structure to define parameters associated with a playlist item
--------	---

Method Summary

int	getIndex()
int	getLength() Number of seconds of track to play
String	getName() Name of playlist stream
int	getStart() Number of seconds into track to start from
void	setIndex (int index)
String	toString()

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

PlaylistItem

```
public PlaylistItem(String name,
                    int start,
                    int length,
                    int index)
```

Class constructor - A simple structure to define parameters associated with a playlist item

Parameters:

name - - name of the playlist stream

start - - number of seconds into the track to start from (-2 means live stream)

(continued from last page)

length -- number of seconds of track to play

index -- index in the playlist

Methods

getName

```
public String getName()
```

Name of playlist stream

Returns:- name of stream

getStart

```
public int getStart()
```

Number of seconds into track to start from

Returns:- number of seconds

getLength

```
public int getLength()
```

Number of seconds of track to play

Returns:- number of seconds

getIndex

```
public int getIndex()
```

setIndex

```
public void setIndex(int index)
```

toString

```
public String toString()
```

com.wowza.wms.stream.publish Class Publisher

java.lang.Object

└─com.wowza.wms.stream.publish.Publisher

```
public class Publisher  
extends Object
```

Publisher: clientless stream publisher. This class can be used to publish raw video, audio and metadata packets to the Wowza Pro server. Here is a quick snippet of code that illustrates how to use it.

This code below will publish data the stream named "myStream". It will be streamed to the default virtual host and available at the rtmp address rtmp://[server-ip-address]/streamtest.

```
IVHost vhost = VHostSingleton.getInstance(VHost.VHOST_DEFAULT);
Publisher publisher = Publisher.createInstance(vhost, "streamtest");

publisher.setFileExtension("flv");
publisher.setStreamType("live");

publisher.publish("myStream", "live");

// sit in a loop adding data
boolean done = false;
while(true)
{
    AMFPacket amfPacket;

    // read packet from audio, video, data source
    // amfPacket = readPacketFromSomewhere();

    switch (amfPacket.getType())
    {
        case IVHost.CONTENTTYPE_AUDIO:
            publisher.addAudioData(amfPacket.getData(), amfPacket.getSize(),
amfPacket.getTimecode());
            break;
        case IVHost.CONTENTTYPE_VIDEO:
            publisher.addVideoData(amfPacket.getData(), amfPacket.getSize(),
amfPacket.getTimecode());
            break;
        case IVHost.CONTENTTYPE_DATA:
            publisher.addDataData(amfPacket.getData(), amfPacket.getSize(),
amfPacket.getTimecode());
            break;
    }
    if (done)
        break;
}

publisher.unpublish();
publisher.close();
```

Basic packet format:

Audio:

AAC

[1-byte header]

[1-byte codec config indicator (1 - audio data, 0 - codec config packet)]

[n-bytes audio content or codec config data]

All others

[1-byte header]

[n-bytes audio content]

Below is the bit

layout of the header byte of data (table goes from least significant bit to most significant bit):

1 bit Number of channels:

- 0 mono
- 1 stereo

1 bit Sample size:

- 0 8 bits per sample
- 1 16 bits per sample

2 bits Sample rate:

- 0 special or 8KHz
- 1 11KHz
- 2 22KHz
- 3 44KHz

4 bits Audio type:

- 0 PCM (big endian)
- 1 PCM (swf - ADPCM)
- 2 MP3
- 3 PCM (little endian)
- 4 Nelly Moser ASAO 16KHz Mono
- 5 Nelly Moser ASAO 8KHz Mono
- 6 Nelly Moser ASAO
- 7 G.711 ALaw
- 8 G.711 MULaw
- 9 Reserved
- a AAC
- b Speex
- f MP3 8Khz

Note: For AAC the codec config data is generally a two byte packet that describes the stream. It must

be published first. Here is the basic code to fill in the codec config data.

```
AACFrame frame = new AACFrame();
```

```
int sampleRate = 22100;
int channels = 2;
frame.setSampleRate(sampleRate);
frame.setRateIndex(AACUtils.sampleRateToIndex(sampleRate));
frame.setChannels(channels);
frame.setChannelIndex(AACUtils.channelCountToIndex(sampleRate));
byte[] codecConfig = new byte[2];
AACUtils.encodeAACCodecConfig(frame, codecConfig, 0);
```

Note: For AAC the header byte is always 0xaf

Note: For Speex the audio data must be encoded as 16000Hz wide band

Video:

H.264

```
[1-byte header]
[1-byte codec config indicator (1 - video data, 0 - codec config packet)]
[3-byte time difference between dts and pts in milliseconds]
[n-bytes video content or codec config data]
```

All others

```
[1-byte header]
[n-bytes audio content]
```

Below is the bit layout of the header byte of data (table goes from least significant bit to most significant bit):

4 bits Video type:

- | | |
|---|------------------------|
| 2 | Sorenson Spark (H.263) |
| 3 | Screen |
| 4 | On2 VP6 |
| 5 | On2 VP6A |
| 6 | Screen2 |
| 7 | H.264 |

2 bit Frame type:

- | | |
|---|---------------------|
| 1 | K frame (key frame) |
| 2 | P frame |
| 3 | B frame |

Note: H.264 codec config data is the same as the AVCc packet in a QuickTime container.

Note: All timecode data is in milliseconds

Method Summary

void	addAudioData (byte[] data, int offset, int len, long timecode)
------	--

void	<code>addAudioData</code> (byte[] data, int offset, int len, long timecode, java.util.Map extraData) Add audio data
void	<code>addAudioData</code> (byte[] data, int len, long timecode) Add audio data
void	<code>addAudioData</code> (byte[] data, long timecode) Add audio data
void	<code>addAudioDataInc</code> (byte[] data, int offset, int len)
void	<code>addDataData</code> (byte[] data, int offset, int len, long timecode)
void	<code>addDataData</code> (byte[] data, int offset, int len, long timecode, java.util.Map extraData) Add metadata
void	<code>addDataData</code> (byte[] data, int len, long timecode) Add metadata
void	<code>addDataData</code> (byte[] data, long timecode) Add metadata
void	<code>addDataDataInc</code> (byte[] data, int offset, int len)
void	<code>addVideoData</code> (byte[] data, int offset, int len, long timecode) Add video data
void	<code>addVideoData</code> (byte[] data, int offset, int len, long timecode, java.util.Map extraData) Add video data
void	<code>addVideoData</code> (byte[] data, int len, long timecode) Add video data
void	<code>addVideoData</code> (byte[] data, long timecode) Add video data
void	<code>addVideoDataInc</code> (byte[] data, int offset, int len)
void	<code>close</code> () Close the publisher
static <code>Publisher</code>	<code>createInstance</code> (<code>IApplicationInstance</code> appInstance)
static <code>Publisher</code>	<code>createInstance</code> (<code>IVHost</code> vhost, String applicationName)
static <code>Publisher</code>	<code>createInstance</code> (<code>IVHost</code> vhost, String applicationName, String appInstanceName)
void	<code>createStream</code> () Create underlying IMediaStream object if not already created
void	<code>flush</code> () Flush the packets from the input buffer to the output buffer

<u>IApplicationInstance</u>	<u>getAppInstance()</u>
String	<u>getFileExtension()</u> Get the file extension (default flv)
long	<u>getLastAudioTimecode()</u> Get last audio timecode written through this publisher (milliseconds).
long	<u>getLastDataTimecode()</u> Get last data timecode written through this publisher (milliseconds).
long	<u>getLastVideoTimecode()</u> Get last video timecode written through this publisher (milliseconds).
long	<u>getMaxTimecode()</u> Highest timecode written through this publisher (milliseconds).
<u>IMediaStream</u>	<u>getStream()</u> Get the media stream object
String	<u>getStreamType()</u>
boolean	<u>isPublishDataEvents()</u>
void	<u>publish(String streamName)</u> Publish a stream (null to stop publishing)
void	<u>publish(String streamName, String howToPublish)</u> Start publishing a stream (streamName = null to stop).
void	<u>setFileExtension(String fileExtension)</u> Set the file extension
void	<u>setPublishDataEvents(boolean publishDataEvents)</u>
void	<u>setStreamType(String streamType)</u> Set the stream type (default live)
void	<u>startAudioData(int len, long timecode)</u>
void	<u>startAudioData(int len, long timecode, java.util.Map extraData)</u>
void	<u>startDataData(int len, long timecode)</u>
void	<u>startDataData(int len, long timecode, java.util.Map extraData)</u>
void	<u>startVideoData(int len, long timecode)</u>
void	<u>startVideoData(int len, long timecode, java.util.Map extraData)</u>
void	<u>unpublish()</u>

Methods inherited from class `java.lang.Object`


```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

Methods

createInstance

```
public static Publisher createInstance(IVHost vhost,
                                         String applicationName)
```

createInstance

```
public static Publisher createInstance(IVHost vhost,
                                         String applicationName,
                                         String appInstanceName)
```

createInstance

```
public static Publisher createInstance(IApplicationInstance appInstance)
```

getStream

```
public IMediaStream getStream()
```

Get the media stream object

Returns:

media stream object

getStreamType

```
public String getStreamType()
```

setStreamType

```
public void setStreamType(String streamType)
```

Set the stream type (default live)

Parameters:

streamType - stream type

publish

```
public void publish(String streamName)
```

Publish a stream (null to stop publishing)

(continued from last page)

Parameters:

streamName - stream name

getFileExtension

```
public String getFileExtension()
```

Get the file extension (default flv)

Returns:

file extension

setFileExtension

```
public void setFileExtension(String fileExtension)
```

Set the file extension

Parameters:

fileExtension - file extension

getAppInstance

```
public IApplicationInstance getAppInstance()
```

unpublish

```
public void unpublish()
```

createStream

```
public void createStream()
```

Create underlying IMediaStream object if not already created

publish

```
public void publish(String streamName,  
String howToPublish)
```

Start publishing a stream (streamName = null to stop). Valid howToPublish values are (live, record, append)

Parameters:

streamName - stream name

howToPublish - publish method (live, record, append)

addVideoData

```
public void addVideoData(byte[] data,  
long timecode)
```

Add video data

Parameters:

data - data

(continued from last page)

timecode - absolute timecode (milliseconds)

addVideoData

```
public void addVideoData(byte[] data,  
    int len,  
    long timecode)
```

Add video data

Parameters:

data - data
len - data length
timecode - absolute timecode (milliseconds)

addVideoData

```
public void addVideoData(byte[] data,  
    int offset,  
    int len,  
    long timecode)
```

Add video data

Parameters:

data - data
offset - offset
len - data length
timecode - absolute timecode (milliseconds)

addVideoData

```
public void addVideoData(byte[] data,  
    int offset,  
    int len,  
    long timecode,  
    java.util.Map extraData)
```

Add video data

Parameters:

data - data
offset - offset
len - data length
timecode - absolute timecode (milliseconds)
extraData - extra data

startVideoData

```
public void startVideoData(int len,  
    long timecode)
```

startVideoData

```
public void startVideoData(int len,  
    long timecode,  
    java.util.Map extraData)
```

(continued from last page)

addVideoDataInc

```
public void addVideoDataInc(byte[] data,  
    int offset,  
    int len)
```

startAudioData

```
public void startAudioData(int len,  
    long timecode)
```

startAudioData

```
public void startAudioData(int len,  
    long timecode,  
    java.util.Map extraData)
```

addAudioDataInc

```
public void addAudioDataInc(byte[] data,  
    int offset,  
    int len)
```

addAudioData

```
public void addAudioData(byte[] data,  
    long timecode)
```

Add audio data

Parameters:

data - data

timecode - absolute timecode (milliseconds)

addAudioData

```
public void addAudioData(byte[] data,  
    int len,  
    long timecode)
```

Add audio data

Parameters:

data - data

len - data length

timecode - absolute timecode (milliseconds)

(continued from last page)

addAudioData

```
public void addAudioData(byte[] data,  
    int offset,  
    int len,  
    long timecode)
```

addAudioData

```
public void addAudioData(byte[] data,  
    int offset,  
    int len,  
    long timecode,  
    java.util.Map extraData)
```

Add audio data

Parameters:

data - data
len - data length
offset - offset
timecode - absolute timecode (milliseconds)

addDataData

```
public void addDataData(byte[] data,  
    long timecode)
```

Add metadata

Parameters:

data - data
timecode - absolute timecode (milliseconds)

addDataData

```
public void addDataData(byte[] data,  
    int len,  
    long timecode)
```

Add metadata

Parameters:

data - data
len - data length
timecode - absolute timecode (milliseconds)

addDataData

```
public void addDataData(byte[] data,  
    int offset,  
    int len,  
    long timecode)
```

(continued from last page)

addDataData

```
public void addDataData(byte[] data,  
    int offset,  
    int len,  
    long timecode,  
    java.util.Map extraData)
```

Add metadata

Parameters:

data - data
offset - offset
len - data length
timecode - absolute timecode (milliseconds)

startDataData

```
public void startDataData(int len,  
    long timecode)
```

startDataData

```
public void startDataData(int len,  
    long timecode,  
    java.util.Map extraData)
```

addDataDataInc

```
public void addDataDataInc(byte[] data,  
    int offset,  
    int len)
```

flush

```
public void flush()
```

Flush the packets from the input buffer to the output buffer

close

```
public void close()
```

Close the publisher

getMaxTimecode

```
public long getMaxTimecode()
```

Highest timecode written through this publisher (milliseconds).

Returns:

highest timecode written through this publisher (milliseconds)

getLastAudioTimecode

```
public long getLastAudioTimecode()
```

Get last audio timecode written through this publisher (milliseconds).

Returns:

last audio timecode written through this publisher (milliseconds).

getLastVideoTimecode

```
public long getLastVideoTimecode()
```

Get last video timecode written through this publisher (milliseconds).

Returns:

last video timecode written through this publisher (milliseconds).

getLastDataTimecode

```
public long getLastDataTimecode()
```

Get last data timecode written through this publisher (milliseconds).

Returns:

last data timecode written through this publisher (milliseconds).

isPublishDataEvents

```
public boolean isPublishDataEvents()
```

setPublishDataEvents

```
public void setPublishDataEvents(boolean publishDataEvents)
```

com.wowza.wms.stream.publish Class PublishingProviderBase

java.lang.Object

└-com.wowza.wms.stream.publish.PublishingProviderBase

Direct Known Subclasses:

[PublishingProviderMediaReader](#), [PublishingProviderLive](#)

```
public class PublishingProviderBase
extends Object
```

PublishingProviderBase: Base class for publishing providers.

Field Summary

protected	sendOnMetadata
-----------	--------------------------------

Constructor Summary

public	PublishingProviderBase()
--------	--

Method Summary

boolean	isSendOnMetadata() True to send onMetadata event on stream start
void	setSendOnMetadata (boolean sendOnMetadata) True to send onMetadata event on stream start

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

sendOnMetadata

protected boolean **sendOnMetadata**

Constructors

PublishingProviderBase

```
public PublishingProviderBase()
```

(continued from last page)

Methods

isSendOnMetadata

```
public boolean isSendOnMetadata()
```

True to send onMetadata event on stream start

Returns:

True to send onMetadata event on stream start

setSendOnMetadata

```
public void setSendOnMetadata(boolean sendOnMetadata)
```

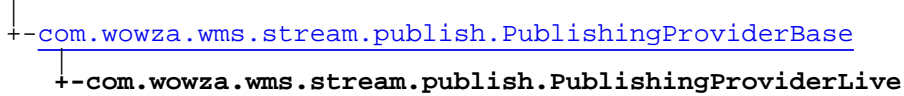
True to send onMetadata event on stream start

Parameters:

sendOnMetadata - True to send onMetadata event on stream start

com.wowza.wms.stream.publish Class PublishingProviderLive

java.lang.Object



All Implemented Interfaces:

[IPublishingProvider](#)

public class **PublishingProviderLive**

extends [PublishingProviderBase](#)

implements [IPublishingProvider](#)

Fields inherited from class [com.wowza.wms.stream.publish.PublishingProviderBase](#)

[sendOnMetadata](#)

Constructor Summary

public	PublishingProviderLive (Publisher publisher, long timeoffset, String streamName)
--------	---

Method Summary

void	close ()
long	getDuration ()
long	getStartOnPreviousBufferTime ()
boolean	isStartOnPreviousKeyFrame ()
boolean	isStopIfStreamMissing ()
boolean	play (Publisher publisher)
boolean	seek (long timecode)
boolean	seek (long timecode, int seekType)
void	setDuration (long duration)
void	setRealTimeStartTime (long realTimeStartTime)
void	setStartOnPreviousBufferTime (long startOnPreviousBufferTime)

void	setStartOnPreviousKeyFrame (boolean startOnPreviousKeyFrame)
void	setStopIfStreamMissing (boolean stopIfStreamMissing)

Methods inherited from class [com.wowza.wms.stream.publish.PublishingProviderBase](#)

[isSendOnMetadata](#), [setSendOnMetadata](#)

Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Methods inherited from interface [com.wowza.wms.stream.publish.IPublishingProvider](#)

[close](#), [isSendOnMetadata](#), [play](#), [seek](#), [seek](#), [setDuration](#), [setRealTimeStartTime](#), [setSendOnMetadata](#)

Constructors

PublishingProviderLive

```
public PublishingProviderLive(Publisher publisher,  
                             long timeoffset,  
                             String streamName)
```

Methods

close

```
public void close()
```

play

```
public boolean play(Publisher publisher)
```

seek

```
public boolean seek(long timecode)
```

seek

```
public boolean seek(long timecode,  
                   int seekType)
```

getDuration

```
public long getDuration()
```

setDuration

```
public void setDuration(long duration)
```

setRealTimeStartTime

```
public void setRealTimeStartTime(long realTimeStartTime)
```

isStartOnPreviousKeyFrame

```
public boolean isStartOnPreviousKeyFrame()
```

setStartOnPreviousKeyFrame

```
public void setStartOnPreviousKeyFrame(boolean startOnPreviousKeyFrame)
```

getStartOnPreviousBufferTime

```
public long getStartOnPreviousBufferTime()
```

setStartOnPreviousBufferTime

```
public void setStartOnPreviousBufferTime(long startOnPreviousBufferTime)
```

isStopIfStreamMissing

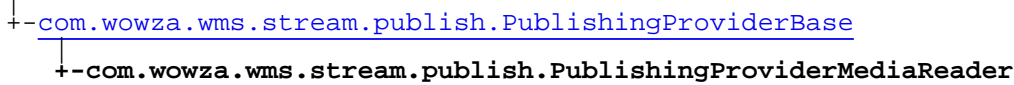
```
public boolean isStopIfStreamMissing()
```

setStopIfStreamMissing

```
public void setStopIfStreamMissing(boolean stopIfStreamMissing)
```

com.wowza.wms.stream.publish Class PublishingProviderMediaReader

java.lang.Object



All Implemented Interfaces:

[IPublishingProvider](#)

public class **PublishingProviderMediaReader**

extends [PublishingProviderBase](#)

implements [IPublishingProvider](#)

PublishingProviderMediaReader: Publishing provider for media reader based stream (vod).

Field Summary

public static final	ONMETADATA_FILTER_DURATION Value: 1
public static final	ONMETADATA_FILTER_NONE Value: 0

Fields inherited from class [com.wowza.wms.stream.publish.PublishingProviderBase](#)

[sendOnMetadata](#)

Constructor Summary

public	PublishingProviderMediaReader (Publisher publisher, long timeoffset, String streamName) Constructor
--------	---

Method Summary

void	close ()
byte[]	filterMetadata (byte[] dataDataBytes)
long	getDuration () Get the target playback duration (milliseconds)
long	getOnMetadataFilter () Filter that control what metadata items are removed from onMetaData event.
double	getSpeed ()
boolean	play (Publisher publisher)

boolean	seek (long timecode)
boolean	seek (long timecode, int seekType)
void	setDuration (long duration)
void	setOnMetadataFilter (long onMetadataFilter) Filter that control what metadata items are removed from onMetaData event.
void	setRealTimeStartTime (long realTimeStartTime)
void	setSpeed (double speed)

Methods inherited from class [com.wowza.wms.stream.publish.PublishingProviderBase](#)

[isSendOnMetadata](#), [setSendOnMetadata](#)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface [com.wowza.wms.stream.publish.IPublishingProvider](#)

[close](#), [isSendOnMetadata](#), [play](#), [seek](#), [seek](#), [setDuration](#), [setRealTimeStartTime](#), [setSendOnMetadata](#)

Fields

ONMETADATA_FILTER_NONE

public static final long **ONMETADATA_FILTER_NONE**

Constant value: **0**

ONMETADATA_FILTER_DURATION

public static final long **ONMETADATA_FILTER_DURATION**

Constant value: **1**

Constructors

PublishingProviderMediaReader

```
public PublishingProviderMediaReader(Publisher publisher,
                                     long timeoffset,
                                     String streamName)
```

Constructor

Parameters:

(continued from last page)

`publisher` - publisher
`timeoffset` - timeoffset (milliseconds)
`streamName` - stream name

Methods

seek

```
public boolean seek(long timecode)
```

seek

```
public boolean seek(long timecode,  
    int seekType)
```

filterMetadata

```
public byte[] filterMetadata(byte[] dataDataBytes)
```

play

```
public boolean play(Publisher publisher)
```

close

```
public void close()
```

getDuration

```
public long getDuration()
```

Get the target playback duration (milliseconds)

Returns:

playback duration (milliseconds)

setDuration

```
public void setDuration(long duration)
```

setRealTimeStartTime

```
public void setRealTimeStartTime(long realTimeStartTime)
```

getSpeed

```
public double getSpeed()
```

setSpeed

```
public void setSpeed(double speed)
```

getOnMetadataFilter

```
public long getOnMetadataFilter()
```

Filter that control what metadata items are removed from onMetaData event. Default is ONMETADATA_FILTER_DURATION. Value is a bit mask of ONMETADATA_FILTER_* value.

Returns:

control what metadata items are removed from onMetaData event

setOnMetadataFilter

```
public void setOnMetadataFilter(long onMetadataFilter)
```

Filter that control what metadata items are removed from onMetaData event. Default is ONMETADATA_FILTER_DURATION. Value is a bit mask of ONMETADATA_FILTER_* value.

Parameters:

onMetadataFilter - control what metadata items are removed from onMetaData event

com.wowza.wms.stream.publish Class Stream

java.lang.Object

└─com.wowza.wms.stream.publish.Stream

All Implemented Interfaces:

Runnable

public class **Stream**
extends Object
implements Runnable

Constructor Summary

public	Stream()
--------	--------------------------

Method Summary

void	addListener(IStreamActionNotify listener) Add a listener
boolean	addToPlaylist(int index, String name, int start, int length) Inserts a media source item to this playlist at a particular index, without interruption.
boolean	addToPlaylist(String existing, String name, int start, int length) Insert item into playlist just after the first item in the playlist with a given name.
void	close() Call this method when you have finished with the playlist object.
void	closeAndWait() Call this method when you have finished with the playlist object.
static Stream	createInstance(IApplicationInstance appInstance, String sName) Use this to create a named Stream on an application instance.
static Stream	createInstance(IVHost vhost, String applicationName, String sName) Use this to create a named Stream on the default instance (_definst_) of an application on a particular VHost.
static Stream	createInstance(IVHost vhost, String applicationName, String appInstanceName, String sName) Use this to create a named Stream on the non-default instance of an application on a particular VHost.
PlaylistItem	getCurrentItem() Get the currently playing playlist item
Object	getLock() Get the synchronization lock for this interface.

String	<u>getName()</u> Returns the name of the playlist stream - the client would play this stream by this name.
java.util.List	<u>getPlaylist()</u> Get the current playlist
int	<u>getPollingInterval()</u> Get the polling interval (milliseconds)
<u>Publisher</u>	<u>getPublisher()</u>
boolean	<u>getRepeat()</u> Use this to determine if the playlist is auto-repeating
long	<u>getStartLiveOnPreviousBufferTime()</u> Get time in milliseconds to go back in live stream buffer to get previous key frame
int	<u>getTimeOffsetBetweenItems()</u> Get time in milliseconds to add to stream time between playlist items (default is zero)
double	<u>getVODSpeed()</u>
boolean	<u>isMoveToNextIfLiveStreamMissing()</u> If true, will move to next playlist item if live stream is missing or is unpublished.
boolean	<u>isSendOnMetadata()</u> True if sending onMetadata events
boolean	<u>isStartLiveOnPreviousKeyFrame()</u> Set to true to start live streams on most recent key frame (smoother switching)
boolean	<u>isSwitchLog()</u> Log when a playlist switch occurs
boolean	<u>isTimesInMilliseconds()</u> If true start time and duration and are milliseconds.
boolean	<u>isUnpublishOnEnd()</u> If true, stop publishing if end of playlist and repeat is false
void	<u>next()</u>
void	<u>next(int n)</u>
void	<u>play(int n)</u>
boolean	<u>play(String sPlaylist)</u> Add a media item to the playlist as defined by an XML file ..
boolean	<u>play(String name, int start, int length, boolean reset)</u> Adds a media source item to this playlist -
void	<u>previous()</u>
void	<u>previous(int n)</u>

boolean	<code>removeFromPlaylist(int index)</code> Remove item from playlist based on index.
boolean	<code>removeFromPlaylist(String name)</code> Remove all items matching the given stream name from the playlist.
void	<code>removeListener(IStreamActionNotify listener)</code> Remove a listener
void	<code>run()</code> Overridden from class Runnable
void	<code>setMoveToNextIfLiveStreamMissing(boolean moveToNextIfLiveStreamMissing)</code> If true, will move to next playlist item if live stream is missing or is unpublished.
void	<code>setPollingInterval(int pollingInterval)</code> Set the polling interval (milliseconds)
void	<code>setRepeat(boolean repeat)</code> Use this to make the playlist repeat or not...
void	<code>setSendOnMetadata(boolean sendOnMetadata)</code> True if sending onMetadata events
void	<code>setStartLiveOnPreviousBufferTime(long startLiveOnPreviousBufferTime)</code> Set time in milliseconds to go back in live stream buffer to get previous key frame
void	<code>setStartLiveOnPreviousKeyFrame(boolean startLiveOnPreviousKeyFrame)</code> Set to true to start live streams on most recent key frame (smoother switching)
void	<code>setSwitchLog(boolean switchLog)</code> Log when a playlist switch occurs
void	<code>setTimeOffsetBetweenItems(int timeOffsetBetweenItems)</code> Set time in milliseconds to add to stream time between playlist items (default is zero)
void	<code>setTimesInMilliseconds(boolean timesInMilliseconds)</code> If true start time and duration and are milliseconds.
void	<code>setUnpublishOnEnd(boolean stopOnEnd)</code> If true, stop publishing if end of playlist and repeat is false
void	<code>setVODSpeed(double vodSpeed)</code>

Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Methods inherited from interface `java.lang.Runnable`

`run`

Constructors

(continued from last page)

Stream

```
public Stream()
```

Methods

createInstance

```
public static Stream createInstance(IVHost vhost,  
    String applicationName,  
    String sName)
```

Use this to create a named Stream on the default instance (_definst_) of an application on a particular VHost.

Parameters:

vhost - - Virtual Host
applicationName - - Application name
sName - - Name of Stream

Returns:

stream interface

createInstance

```
public static Stream createInstance(IVHost vhost,  
    String applicationName,  
    String appInstanceName,  
    String sName)
```

Use this to create a named Stream on the non-default instance of an application on a particular VHost.

Parameters:

vhost - - Virtual Host
applicationName - - Application name
appInstanceName - - Instance name
sName - - Name of Stream

Returns:

stream interface

getLock

```
public Object getLock()
```

Get the synchronization lock for this interface.

Returns:

synchronization lock for this interface

createInstance

```
public static Stream createInstance(IApplicationInstance appInstance,  
    String sName)
```

Use this to create a named Stream on an application instance.

Parameters:

(continued from last page)

appInstance - - Application instance
sName - - Name of Stream

Returns:

stream interface

run

```
public void run()
```

Overridden from class Runnable do NOT call this directly. This method handles switching between media sources in the playlist.

getPlaylist

```
public java.util.List getPlaylist()
```

Get the current playlist

Returns:

current playlist

getCurrentItem

```
public PlaylistItem getCurrentItem()
```

Get the currently playing playlist item

Returns:

currently playing playlist item

play

```
public boolean play(String name,  
                    int start,  
                    int length,  
                    boolean reset)
```

Adds a media source item to this playlist -

Parameters:

name - - name of media item
start - - where to start playing the item. (-2 implies play a live stream)
length - - how much of the item to play (-1 implies play the entire file or live stream)
reset - - if true, will begin a new playlist, otherwise items are appended

Returns:

- returns true if item was added successfully ...otherwise false.

addToPlaylist

```
public boolean addToPlaylist(int index,  
                             String name,  
                             int start,  
                             int length)
```

Inserts a media source item to this playlist at a particular index, without interruption.

Parameters:

index - - insertion index

(continued from last page)

name -- name of media item being inserted

start -- where to start playing the item. (-2 implies play a live stream)

length -- how much of the item to play (-1 implies play the entire file or live stream)

Returns:

- returns true if item was inserted successfully ...otherwise false. this should not interrupt anything that might be currently playing.... will do nothing if existing item does not exist.. if existing == "" insert at head of list

addToPlaylist

```
public boolean addToPlaylist(String existing,  
    String name,  
    int start,  
    int length)
```

Insert item into playlist just after the first item in the playlist with a given name.

Parameters:

existing - name of playlist item in which to insert the item after

name - name of new item

start - where to start playing the item. (-2 implies play a live stream)

length - how much of the item to play (-1 implies play the entire file or live stream)

Returns:

true if item inserted

removeFromPlaylist

```
public boolean removeFromPlaylist(int index)
```

Remove item from playlist based on index.

Parameters:

index - item index

Returns:

true if at least one item from removed

removeFromPlaylist

```
public boolean removeFromPlaylist(String name)
```

Remove all items matching the given stream name from the playlist. If the item is currently being played, it is not removed.

Parameters:

name - stream name

Returns:

true if at least one item from removed

play

```
public boolean play(String sPlaylist)
```

Add a media item to the playlist as defined by an XML file ..

Parameters:

sPlaylist -- the playlist XML definition file

(continued from last page)

Returns:

- returns true if item was added successfully ...otherwise false.

close

```
public void close()
```

Call this method when you have finished with the playlist object. It will terminate the playlist thread. The playlist object cannot be used after this call.

closeAndWait

```
public void closeAndWait()
```

Call this method when you have finished with the playlist object. It will terminate the playlist thread. The playlist object cannot be used after this call. This method will wait for the thread to exit.

getName

```
public String getName()
```

Returns the name of the playlist stream - the client would play this stream by this name.

Returns:

- the name of the playlist stream

getRepeat

```
public boolean getRepeat()
```

Use this to determine if the playlist is auto-repeating

Returns:

- true if repeating otherwise false.

setRepeat

```
public void setRepeat(boolean repeat)
```

Use this to make the playlist repeat or not...

Parameters:

repeat - - true to repeat otherwise false

play

```
public void play(int n)
```

next

```
public void next(int n)
```

(continued from last page)

next

```
public void next()
```

previous

```
public void previous()
```

previous

```
public void previous(int n)
```

getPollingInterval

```
public int getPollingInterval()
```

Get the polling interval (milliseconds)

Returns:

polling interval (milliseconds)

setPollingInterval

```
public void setPollingInterval(int pollingInterval)
```

Set the polling interval (milliseconds)

Parameters:

pollingInterval - polling interval (milliseconds)

getPublisher

```
public Publisher getPublisher()
```

addListener

```
public void addListener(IStreamActionNotify listener)
```

Add a listener

Parameters:

listener - listener

removeListener

```
public void removeListener(IStreamActionNotify listener)
```

Remove a listener

Parameters:

(continued from last page)

listener - listener

isSwitchLog

```
public boolean isSwitchLog()
```

Log when a playlist switch occurs

Returns:

log when a playlist switch occurs

setSwitchLog

```
public void setSwitchLog(boolean switchLog)
```

Log when a playlist switch occurs

Parameters:

switchLog - log when a playlist switch occurs

isSendOnMetadata

```
public boolean isSendOnMetadata()
```

True if sending onMetadata events

Returns:

True if sending onMetadata events

setSendOnMetadata

```
public void setSendOnMetadata(boolean sendOnMetadata)
```

True if sending onMetadata events

Parameters:

sendOnMetadata - True if sending onMetadata events

isTimesInMilliseconds

```
public boolean isTimesInMilliseconds()
```

If true start time and duration and are milliseconds. If false startTime and duration are in seconds.

Returns:

true start time and duration and are milliseconds

setTimesInMilliseconds

```
public void setTimesInMilliseconds(boolean timesInMilliseconds)
```

If true start time and duration and are milliseconds. If false startTime and duration are in seconds.

Parameters:

timesInMilliseconds - true start time and duration and are milliseconds

(continued from last page)

isStartLiveOnPreviousKeyFrame

```
public boolean isStartLiveOnPreviousKeyFrame( )
```

Set to true to start live streams on most recent key frame (smoother switching)

Returns:

true to start live streams on most recent key frame

setStartLiveOnPreviousKeyFrame

```
public void setStartLiveOnPreviousKeyFrame(boolean startLiveOnPreviousKeyFrame)
```

Set to true to start live streams on most recent key frame (smoother switching)

Parameters:

startLiveOnPreviousKeyFrame - true to start live streams on most recent key frame

getStartLiveOnPreviousBufferTime

```
public long getStartLiveOnPreviousBufferTime( )
```

Get time in milliseconds to go back in live stream buffer to get previous key frame

Returns:

time in milliseconds to go back in live stream buffer to get previous key frame

setStartLiveOnPreviousBufferTime

```
public void setStartLiveOnPreviousBufferTime(long startLiveOnPreviousBufferTime)
```

Set time in milliseconds to go back in live stream buffer to get previous key frame

Parameters:

startLiveOnPreviousBufferTime - time in milliseconds to go back in live stream buffer to get previous key frame

getTimeOffsetBetweenItems

```
public int getTimeOffsetBetweenItems( )
```

Get time in milliseconds to add to stream time between playlist items (default is zero)

Returns:

time in milliseconds to add to stream time between playlist items

setTimeOffsetBetweenItems

```
public void setTimeOffsetBetweenItems(int timeOffsetBetweenItems)
```

Set time in milliseconds to add to stream time between playlist items (default is zero)

Parameters:

timeOffsetBetweenItems - time in milliseconds to add to stream time between playlist items

isMoveToNextIfLiveStreamMissing

```
public boolean isMoveToNextIfLiveStreamMissing( )
```

(continued from last page)

If true, will move to next playlist item if live stream is missing or is unpublished.

Returns:

move to next playlist item if live stream is missing or is unpublished

setMoveToNextIfLiveStreamMissing

```
public void setMoveToNextIfLiveStreamMissing(boolean moveToNextIfLiveStreamMissing)
```

If true, will move to next playlist item if live stream is missing or is unpublished.

Parameters:

moveToNextIfLiveStreamMissing - move to next playlist item if live stream is missing or is unpublished

isUnpublishOnEnd

```
public boolean isUnpublishOnEnd()
```

If true, stop publishing if end of playlist and repeat is false

Returns:

stop publishing if end of playlist and repeat is false

setUnpublishOnEnd

```
public void setUnpublishOnEnd(boolean stopOnEnd)
```

If true, stop publishing if end of playlist and repeat is false

Parameters:

stopOnEnd - stop publishing if end of playlist and repeat is false

getVODSpeed

```
public double getVODSpeed()
```

setVODSpeed

```
public void setVODSpeed(double vodSpeed)
```

Package

com.wowza.wms.timedtext.cea608

com.wowza.wms.timedtext.cea608

Class ClosedCaptionCEA608Utils

java.lang.Object

└-com.wowza.wms.timedtext.cea608.CEA608Constants

└-com.wowza.wms.timedtext.cea608.ClosedCaptionCEA608Utils

Direct Known Subclasses:

[ClosedCaptionCEA608Utils](#)

```
public class ClosedCaptionCEA608Utils
extends CEA608Constants
```

Field Summary

public static final	CEA608_EXTENDED_FR_SP_MAP
public static final	CEA608_EXTENDED_NA_MAP
public static final	CEA608_EXTENDED_PORT_GERMAN_DANISH_MAP
public static	CEA608_MUSICAL_NOTE
public static	CEA608_SOLID_BLOCK
public static final	CHANNEL1 Value: 0
public static final	CHANNEL2 Value: 1
public static final	COL_0 Value: 0
public static final	COL_12 Value: 6
public static final	COL_16 Value: 8
public static final	COL_20 Value: 10
public static final	COL_24 Value: 12

public static final	<u>COL_28</u> Value: 14
public static final	<u>COL_4</u> Value: 2
public static final	<u>COL_5</u> Value: 4
public static final	<u>COL0_BLUE</u> Value: 4
public static final	<u>COL0_CYAN</u> Value: 6
public static final	<u>COL0_GREEN</u> Value: 2
public static final	<u>COL0_HIBYTE</u> Move to set color (high-byte) [channel][row]
public static final	<u>COL0_LOWBYTE</u> Move to set color (low-byte) [color][row]
public static final	<u>COL0_MAGENTA</u> Value: 12
public static final	<u>COL0_RED</u> Value: 8
public static final	<u>COL0_WHITE</u> Value: 0
public static final	<u>COL0_YELLOW</u> Value: 10
public static final	<u>COLN_LOWBYTE</u>
public static final	<u>CONTROLCODES_AOF</u> AOF Reserved (formerly Alarm Off) [channel]
public static final	<u>CONTROLCODES_AON</u> AON Reserved (formerly Alarm On) [channel]
public static final	<u>CONTROLCODES_BAO</u> BAO Background Black, Opaque [channel]
public static final	<u>CONTROLCODES_BAS</u> BAS Background Black, Semi-transparent [channel]
public static final	<u>CONTROLCODES_BBO</u> BBO Background Blue, Opaque [channel]

public static final	CONTROLCODES_BBS BBS Background Blue, Semi-transparent [channel]
public static final	CONTROLCODES_BCO BCO Background Cyan, Opaque [channel]
public static final	CONTROLCODES_BCS BCS Background Cyan, Semi-transparent [channel]
public static final	CONTROLCODES_BGO BGO Background Green, Opaque [channel]
public static final	CONTROLCODES_BGS BGS Background Green, Semi-transparent [channel]
public static final	CONTROLCODES_BMO BMO Background Magenta, Opaque [channel]
public static final	CONTROLCODES_BMS BMS Background Magenta, Semi-transparent [channel]
public static final	CONTROLCODES_BRO BRO Background Red, Opaque [channel]
public static final	CONTROLCODES_BRS BRS Background Red, Semi-transparent [channel]
public static final	CONTROLCODES_BS BS Backspace [channel]
public static final	CONTROLCODES_BT BT Background Transparent [channel]
public static final	CONTROLCODES_BWO BWO Background White, Opaque [channel]
public static final	CONTROLCODES_BWS BWS Background White, Semi-transparent [channel]
public static final	CONTROLCODES_BYO BYO Background Yellow, Opaque [channel]
public static final	CONTROLCODES_BYS BYS Background Yellow, Semi-transparent [channel]
public static final	CONTROLCODES_CR CR Carriage Return [channel]
public static final	CONTROLCODES_DER DER Delete to End of Row [channel]
public static final	CONTROLCODES_EDM EDM Erase Displayed Memory [channel]
public static final	CONTROLCODES_ENM ENM Erase Non-Displayed Memory [channel]
public static final	CONTROLCODES_EOC EOC End of Caption (Flip Memories) [channel]

public static final	<u>CONTROLCODES_FA</u> FA Foreground Black [channel]
public static final	<u>CONTROLCODES_FAU</u> FAU Foreground Black Underline [channel]
public static final	<u>CONTROLCODES_FON</u> FON Flash On [channel]
public static final	<u>CONTROLCODES_RCL</u> RCL Resume caption loading [channel]
public static final	<u>CONTROLCODES_RDC</u> RDC Resume Direct Captioning [channel]
public static final	<u>CONTROLCODES_RTD</u> RTD Resume Text Display [channel]
public static final	<u>CONTROLCODES_RU2</u> RU2 Roll-Up Captions-2 Rows [channel]
public static final	<u>CONTROLCODES_RU3</u> RU3 Roll-Up Captions-3 Rows [channel]
public static final	<u>CONTROLCODES_RU4</u> RU4 Roll-Up Captions-4 Rows [channel]
public static final	<u>CONTROLCODES_TO1</u> TO1 Tab Offset 1 Column [channel]
public static final	<u>CONTROLCODES_TO2</u> TO2 Tab Offset 2 Columns [channel]
public static final	<u>CONTROLCODES_TO3</u> TO3 Tab Offset 3 Columns [channel]
public static final	<u>CONTROLCODES_TR</u> TR Text Restart [channel]
public static final	<u>COUNT_COLS</u> Total number of columns per frame Value: 31
public static final	<u>COUNT_ROWS</u> Total number of rows of characters per frame Value: 15
public static final	<u>DONOTHING_BYTE</u> Value: -128
public static final	<u>DONOTHING_SHORT</u> Value: -32640
public static final	<u>FORMAT_COLOR</u>
public static final	<u>FORMAT_FLASH</u> Value: -27480

public static final	<u>FORMAT_ITALICS</u> Value: -28242
public static final	<u>FORMAT_ITALICS_UNDERLINE</u> Value: -28369
public static final	<u>LANG_ASCII</u> Select the standard line 21 character set in normal size.
public static final	<u>LANG_ASCII_DBL</u> Select the standard line 21 character set in double size.
public static final	<u>LANG_GB231280</u> Select the People's Republic of China character set: GB 2312-80.
public static final	<u>LANG_KSC56011987</u> Select the Korean Standard character set: KSC 5601-1987.
public static final	<u>LANG_PRIV1</u> Select the first private character set.
public static final	<u>LANG_PRIV2</u> Select the second private character set.
public static final	<u>LANG_REG1</u> Select the first registered character set.
public static final	<u>MAX_CCCOUNT</u> Maximum number of closed caption codes per SEI packet Value: 31
public static final	<u>ODDPARITY</u> Odd parity calculator
public static final	<u>SEI_PAYLOADTYPE</u> Value: 4
public static final	<u>SEI_STARTCODE</u> SEI start code
public static final	<u>SEI_USERDATATYPECODE</u> Value: 3
public static final	<u>UTF8_BRITISH_POUND</u> Value: 163
public static final	<u>UTF8_BULLET</u> Value: 8226
public static final	<u>UTF8_CENTS</u> Value: 162
public static final	<u>UTF8_CLOSING_DOUBLE_QUOTE</u> Value: 8221

public static final	<u>UTF8_CLOSING_GUILLEMETS</u> Value: 187
public static final	<u>UTF8_CONNECTING_VERTICAL_BAR</u> Value: 9474
public static final	<u>UTF8_COPYRIGHT</u> Value: 169
public static final	<u>UTF8_CURRENCY</u> Value: 164
public static final	<u>UTF8_DASH</u> Value: 45
public static final	<u>UTF8_DEGREE</u> Value: 176
public static final	<u>UTF8_DOUBLEQUOTE</u> Value: 34
public static final	<u>UTF8_EM_DASH</u> Value: 8212
public static final	<u>UTF8_EXCLAMATION</u> Value: 33
public static final	<u>UTF8_GREATERTHAN</u> Value: 62
public static final	<u>UTF8_INVERTED_EXCLAMATION</u> Value: 161
public static final	<u>UTF8_LEFT_SINGLE_QUOTATION</u> Value: 8216
public static final	<u>UTF8_LEFTPAREN</u> Value: 40
public static final	<u>UTF8_LESSTHAN</u> Value: 60
public static final	<u>UTF8_LOWER_A</u> Value: 97
public static final	<u>UTF8_LOWER_A_WITH_CIRCUMFLEX</u> Value: 226

public static final	<u>UTF8_LOWER_A_WITH_DIAERESIS</u> Value: 228
public static final	<u>UTF8_LOWER_A_WITH_GRAVE</u> Value: 224
public static final	<u>UTF8_LOWER_A_WITH_RING</u> Value: 229
public static final	<u>UTF8_LOWER_A_WITH_TILDE</u> Value: 227
public static final	<u>UTF8_LOWER_B</u> Value: 98
public static final	<u>UTF8_LOWER_C</u> Value: 99
public static final	<u>UTF8_LOWER_D</u> Value: 100
public static final	<u>UTF8_LOWER_E</u> Value: 101
public static final	<u>UTF8_LOWER_E_WITH_CIRCUMFLEX</u> Value: 234
public static final	<u>UTF8_LOWER_E_WITH_DIAERESIS</u> Value: 235
public static final	<u>UTF8_LOWER_E_WITH_GRAVE</u> Value: 232
public static final	<u>UTF8_LOWER_F</u> Value: 102
public static final	<u>UTF8_LOWER_G</u> Value: 103
public static final	<u>UTF8_LOWER_H</u> Value: 104
public static final	<u>UTF8_LOWER_I</u> Value: 105
public static final	<u>UTF8_LOWER_I_WITH_CIRCUMFLEX</u> Value: 238

public static final	<u>UTF8_LOWER_I_WITH_DIAERESIS</u> Value: 239
public static final	<u>UTF8_LOWER_I_WITH_GRAVE</u> Value: 236
public static final	<u>UTF8_LOWER_J</u> Value: 106
public static final	<u>UTF8_LOWER_K</u> Value: 107
public static final	<u>UTF8_LOWER_L</u> Value: 108
public static final	<u>UTF8_LOWER_M</u> Value: 109
public static final	<u>UTF8_LOWER_N</u> Value: 110
public static final	<u>UTF8_LOWER_O</u> Value: 111
public static final	<u>UTF8_LOWER_O_WITH_CIRCUMFLEX</u> Value: 244
public static final	<u>UTF8_LOWER_O_WITH_DIAERESIS</u> Value: 246
public static final	<u>UTF8_LOWER_O_WITH_GRAVE</u> Value: 242
public static final	<u>UTF8_LOWER_O_WITH_STROKE</u> Value: 248
public static final	<u>UTF8_LOWER_O_WITH_TILDE</u> Value: 245
public static final	<u>UTF8_LOWER_P</u> Value: 112
public static final	<u>UTF8_LOWER_Q</u> Value: 113
public static final	<u>UTF8_LOWER_R</u> Value: 114

public static final	<u>UTF8_LOWER_S</u> Value: 115
public static final	<u>UTF8_LOWER_T</u> Value: 116
public static final	<u>UTF8_LOWER_U</u> Value: 117
public static final	<u>UTF8_LOWER_U_WITH_CIRCUMFLEX</u> Value: 251
public static final	<u>UTF8_LOWER_U_WITH_DIAERESIS</u> Value: 252
public static final	<u>UTF8_LOWER_U_WITH_GRAVE</u> Value: 249
public static final	<u>UTF8_LOWER_V</u> Value: 118
public static final	<u>UTF8_LOWER_W</u> Value: 119
public static final	<u>UTF8_LOWER_X</u> Value: 120
public static final	<u>UTF8_LOWER_Y</u> Value: 121
public static final	<u>UTF8_LOWER_Z</u> Value: 122
public static final	<u>UTF8_MINUSHYPHEN</u> Value: 45
public static final	<u>UTF8_MUSICAL_NOTE</u> Value: 9834
public static final	<u>UTF8_NBSP</u> Value: 160
public static final	<u>UTF8_NOTMATCH</u>
public static final	<u>UTF8_ONE_HALF</u> Value: 189

public static final	<u>UTF8_OPENING_DOUBLE_QUOTE</u> Value: 8220
public static final	<u>UTF8_OPENING_GUILLEMETS</u> Value: 171
public static final	<u>UTF8_PERIOD</u> Value: 46
public static final	<u>UTF8_PIPE</u> Value: 124
public static final	<u>UTF8_QUESTION</u> Value: 63
public static final	<u>UTF8_REGISTERED_MARK</u> Value: 174
public static final	<u>UTF8_REVERSE_QUERY</u> Value: 191
public static final	<u>UTF8_RIGHTPAREN</u> Value: 41
public static final	<u>UTF8_SERVICE_MARK</u> Value: 8480
public static final	<u>UTF8_SINGLEQUOTE</u> Value: 39
public static final	<u>UTF8_SPACE</u> Value: 32
public static final	<u>UTF8_TRADEMARK</u> Value: 8482
public static final	<u>UTF8_UNDERSCORE</u> Value: 84
public static final	<u>UTF8_UPPER_A</u> Value: 65
public static final	<u>UTF8_UPPER_A_WITH_ACUTE</u> Value: 193
public static final	<u>UTF8_UPPER_A_WITH_CURCUMFLEX</u> Value: 194

public static final	<u>UTF8_UPPER_A_WITH_DIAERESIS</u> Value: 196
public static final	<u>UTF8_UPPER_A_WITH_GRAVE</u> Value: 192
public static final	<u>UTF8_UPPER_A_WITH_RING</u> Value: 197
public static final	<u>UTF8_UPPER_A_WITH_TILDE</u> Value: 195
public static final	<u>UTF8_UPPER_B</u> Value: 66
public static final	<u>UTF8_UPPER_C</u> Value: 67
public static final	<u>UTF8_UPPER_C_WITH_CEDILLA</u> Value: 199
public static final	<u>UTF8_UPPER_D</u> Value: 68
public static final	<u>UTF8_UPPER_E</u> Value: 69
public static final	<u>UTF8_UPPER_E_WITH_ACUTE</u> Value: 201
public static final	<u>UTF8_UPPER_E_WITH_CIRCUMFLEX</u> Value: 202
public static final	<u>UTF8_UPPER_E_WITH_DIAERESIS</u> Value: 203
public static final	<u>UTF8_UPPER_E_WITH_GRAVE</u> Value: 200
public static final	<u>UTF8_UPPER_F</u> Value: 70
public static final	<u>UTF8_UPPER_G</u> Value: 71
public static final	<u>UTF8_UPPER_H</u> Value: 72

public static final	<u>UTF8_UPPER_I</u> Value: 73
public static final	<u>UTF8_UPPER_I_WITH_CIRCUMFLEX</u> Value: 206
public static final	<u>UTF8_UPPER_I_WITH_DIAERESIS</u> Value: 207
public static final	<u>UTF8_UPPER_I_WITH_GRAVE</u> Value: 204
public static final	<u>UTF8_UPPER_I_WITH_TILDE</u> Value: 205
public static final	<u>UTF8_UPPER_J</u> Value: 74
public static final	<u>UTF8_UPPER_K</u> Value: 75
public static final	<u>UTF8_UPPER_L</u> Value: 76
public static final	<u>UTF8_UPPER_M</u> Value: 77
public static final	<u>UTF8_UPPER_N</u> Value: 78
public static final	<u>UTF8_UPPER_O</u> Value: 79
public static final	<u>UTF8_UPPER_O_WITH_ACUTE</u> Value: 211
public static final	<u>UTF8_UPPER_O_WITH_CIRCUMFLEX</u> Value: 212
public static final	<u>UTF8_UPPER_O_WITH_DIAERESIS</u> Value: 214
public static final	<u>UTF8_UPPER_O_WITH_GRAVE</u> Value: 210
public static final	<u>UTF8_UPPER_O_WITH_STROKE</u> Value: 216

public static final	<u>UTF8_UPPER_O_WITH_TILDE</u> Value: 213
public static final	<u>UTF8_UPPER_P</u> Value: 80
public static final	<u>UTF8_UPPER_Q</u> Value: 81
public static final	<u>UTF8_UPPER_R</u> Value: 82
public static final	<u>UTF8_UPPER_S</u> Value: 83
public static final	<u>UTF8_UPPER_T</u> Value: 84
public static final	<u>UTF8_UPPER_U</u> Value: 85
public static final	<u>UTF8_UPPER_U_WITH_ACUTE</u> Value: 218
public static final	<u>UTF8_UPPER_U_WITH_CIRCUMFLEX</u> Value: 219
public static final	<u>UTF8_UPPER_U_WITH_DIAERESIS</u> Value: 220
public static final	<u>UTF8_UPPER_U_WITH_GRAVE</u> Value: 217
public static final	<u>UTF8_UPPER_V</u> Value: 86
public static final	<u>UTF8_UPPER_W</u> Value: 87
public static final	<u>UTF8_UPPER_X</u> Value: 88
public static final	<u>UTF8_UPPER_Y</u> Value: 89
public static final	<u>UTF8_UPPER_Z</u> Value: 90

public static final	UTF8_YEN Value: 165
public static final	UTF8MAP
public static final	UTF8MAP_INDEX_OFFSET Value: 32

Fields inherited from class com.wowza.wms.timedtext.cea608.CEA608Constants

CC_CHANNEL_CC1, CC_CHANNEL_CC2, CC_CHANNEL_CC3, CC_CHANNEL_CC4, CC_TYPE_DTVCC_PACKET_DATA, CC_TYPE_DTVCC_PACKET_START, CC_TYPE_NTSC_FIELD_1, CC_TYPE_NTSC_FIELD_2

Constructor Summary

public	ClosedCaptionCEA608Utils()
--------	--

Method Summary

static byte	calcParity (byte inByte)
static short	calcParity (short inShort)
static char	cea608CharToCharUTF8 (short b)
static Character	cea608ExtendedCharacterToUTF8 (short command)
static byte[]	charUTF8ToCAE608 (CEACaptionConversionContext ceaContext, String text, int pos, String characterSet, int channel)
static byte[]	charUTF8ToCAE608 (String text, int pos, String characterSet, int channel)
static boolean	checkParity (byte inByte)
static boolean	checkParity (short inShort)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

SEI_PAYLOADTYPE

public static final byte **SEI_PAYLOADTYPE**

Constant value: **4**

SEI_USERDATATYPECODE

```
public static final byte SEI_USERDATATYPECODE
```

Constant value: **3**

UTF8_BULLET

```
public static final char UTF8_BULLET
```

Constant value: **8226**

UTF8_CLOSING_DOUBLE_QUOTE

```
public static final char UTF8_CLOSING_DOUBLE_QUOTE
```

Constant value: **8221**

UTF8_CLOSING_GUILLEMETS

```
public static final char UTF8_CLOSING_GUILLEMETS
```

Constant value: **187**

UTF8_CONNECTING_VERTICAL_BAR

```
public static final char UTF8_CONNECTING_VERTICAL_BAR
```

Constant value: **9474**

UTF8_COPYRIGHT

```
public static final char UTF8_COPYRIGHT
```

Constant value: **169**

UTF8_CURRENCY

```
public static final char UTF8_CURRENCY
```

Constant value: **164**

UTF8_DEGREE

```
public static final char UTF8_DEGREE
```

Constant value: **176**

(continued from last page)

UTF8_EM_DASH

```
public static final char UTF8_EM_DASH
```

Constant value: **8212**

UTF8_INVERTED_EXCLAMATION

```
public static final char UTF8_INVERTED_EXCLAMATION
```

Constant value: **161**

UTF8_LEFT_SINGLE_QUOTATION

```
public static final char UTF8_LEFT_SINGLE_QUOTATION
```

Constant value: **8216**

UTF8_OPENING_DOUBLE_QUOTE

```
public static final char UTF8_OPENING_DOUBLE_QUOTE
```

Constant value: **8220**

UTF8_OPENING_GUILLEMETS

```
public static final char UTF8_OPENING_GUILLEMETS
```

Constant value: **171**

UTF8_REGISTERED_MARK

```
public static final char UTF8_REGISTERED_MARK
```

Constant value: **174**

UTF8_SERVICE_MARK

```
public static final char UTF8_SERVICE_MARK
```

Constant value: **8480**

UTF8_YEN

```
public static final char UTF8_YEN
```

Constant value: **165**

UTF8_LOWER_A_WITH_DIAERESIS

```
public static final char UTF8_LOWER_A_WITH_DIAERESIS
```

(continued from last page)

Constant value: **228**

UTF8_LOWER_A_WITH_RING

```
public static final char UTF8_LOWER_A_WITH_RING
```

Constant value: **229**

UTF8_LOWER_A_WITH_TILDE

```
public static final char UTF8_LOWER_A_WITH_TILDE
```

Constant value: **227**

UTF8_LOWER_E_WITH_DIAERESIS

```
public static final char UTF8_LOWER_E_WITH_DIAERESIS
```

Constant value: **235**

UTF8_LOWER_I_WITH_DIAERESIS

```
public static final char UTF8_LOWER_I_WITH_DIAERESIS
```

Constant value: **239**

UTF8_LOWER_I_WITH_GRAVE

```
public static final char UTF8_LOWER_I_WITH_GRAVE
```

Constant value: **236**

UTF8_LOWER_O_WITH_DIAERESIS

```
public static final char UTF8_LOWER_O_WITH_DIAERESIS
```

Constant value: **246**

UTF8_LOWER_O_WITH_GRAVE

```
public static final char UTF8_LOWER_O_WITH_GRAVE
```

Constant value: **242**

UTF8_LOWER_O_WITH_STROKE

```
public static final char UTF8_LOWER_O_WITH_STROKE
```

Constant value: **248**

UTF8_LOWER_O_WITH_TILDE

```
public static final char UTF8_LOWER_O_WITH_TILDE
```

Constant value: **245**

UTF8_LOWER_U_WITH_DIAERESIS

```
public static final char UTF8_LOWER_U_WITH_DIAERESIS
```

Constant value: **252**

UTF8_LOWER_U_WITH_GRAVE

```
public static final char UTF8_LOWER_U_WITH_GRAVE
```

Constant value: **249**

UTF8_UPPER_A_WITH_ACUTE

```
public static final char UTF8_UPPER_A_WITH_ACUTE
```

Constant value: **193**

UTF8_UPPER_A_WITH_CURCUMFLEX

```
public static final char UTF8_UPPER_A_WITH_CURCUMFLEX
```

Constant value: **194**

UTF8_UPPER_A_WITH_DIAERESIS

```
public static final char UTF8_UPPER_A_WITH_DIAERESIS
```

Constant value: **196**

UTF8_UPPER_A_WITH_GRAVE

```
public static final char UTF8_UPPER_A_WITH_GRAVE
```

Constant value: **192**

UTF8_UPPER_A_WITH_RING

```
public static final char UTF8_UPPER_A_WITH_RING
```

Constant value: **197**

(continued from last page)

UTF8_UPPER_A_WITH_TILDE

```
public static final char UTF8_UPPER_A_WITH_TILDE
```

Constant value: **195**

UTF8_UPPER_C_WITH_CEDILLA

```
public static final char UTF8_UPPER_C_WITH_CEDILLA
```

Constant value: **199**

UTF8_UPPER_E_WITH_ACUTE

```
public static final char UTF8_UPPER_E_WITH_ACUTE
```

Constant value: **201**

UTF8_UPPER_E_WITH_CIRCUMFLEX

```
public static final char UTF8_UPPER_E_WITH_CIRCUMFLEX
```

Constant value: **202**

UTF8_UPPER_E_WITH_DIAERESIS

```
public static final char UTF8_UPPER_E_WITH_DIAERESIS
```

Constant value: **203**

UTF8_UPPER_E_WITH_GRAVE

```
public static final char UTF8_UPPER_E_WITH_GRAVE
```

Constant value: **200**

UTF8_UPPER_I_WITH_CIRCUMFLEX

```
public static final char UTF8_UPPER_I_WITH_CIRCUMFLEX
```

Constant value: **206**

UTF8_UPPER_I_WITH_DIAERESIS

```
public static final char UTF8_UPPER_I_WITH_DIAERESIS
```

Constant value: **207**

UTF8_UPPER_I_WITH_GRAVE

```
public static final char UTF8_UPPER_I_WITH_GRAVE
```

(continued from last page)

Constant value: **204**

UTF8_UPPER_I_WITH_TILDE

```
public static final char UTF8_UPPER_I_WITH_TILDE
```

Constant value: **205**

UTF8_UPPER_O_WITH_ACUTE

```
public static final char UTF8_UPPER_O_WITH_ACUTE
```

Constant value: **211**

UTF8_UPPER_O_WITH_CIRCUMFLEX

```
public static final char UTF8_UPPER_O_WITH_CIRCUMFLEX
```

Constant value: **212**

UTF8_UPPER_O_WITH_DIAERESIS

```
public static final char UTF8_UPPER_O_WITH_DIAERESIS
```

Constant value: **214**

UTF8_UPPER_O_WITH_GRAVE

```
public static final char UTF8_UPPER_O_WITH_GRAVE
```

Constant value: **210**

UTF8_UPPER_O_WITH_STROKE

```
public static final char UTF8_UPPER_O_WITH_STROKE
```

Constant value: **216**

UTF8_UPPER_O_WITH_TILDE

```
public static final char UTF8_UPPER_O_WITH_TILDE
```

Constant value: **213**

UTF8_UPPER_U_WITH_ACUTE

```
public static final char UTF8_UPPER_U_WITH_ACUTE
```

Constant value: **218**

UTF8_UPPER_U_WITH_CIRCUMFLEX

```
public static final char UTF8_UPPER_U_WITH_CIRCUMFLEX
```

Constant value: **219**

UTF8_UPPER_U_WITH_DIAERESIS

```
public static final char UTF8_UPPER_U_WITH_DIAERESIS
```

Constant value: **220**

UTF8_UPPER_U_WITH_GRAVE

```
public static final char UTF8_UPPER_U_WITH_GRAVE
```

Constant value: **217**

MAX_CCCOUNT

```
public static final int MAX_CCCOUNT
```

Maximum number of closed caption codes per SEI packet
Constant value: **31**

SEI_STARTCODE

```
public static final byte SEI_STARTCODE
```

SEI start code

ODDPARITY

```
public static final byte ODDPARITY
```

Odd parity calculator

COL0_WHITE

```
public static final int COL0_WHITE
```

Constant value: **0**

COL0_GREEN

```
public static final int COL0_GREEN
```

Constant value: **2**

COL0_BLUE

```
public static final int COL0_BLUE
```

(continued from last page)

Constant value: **4**

COL0_CYAN

```
public static final int COL0_CYAN
```

Constant value: **6**

COL0_RED

```
public static final int COL0_RED
```

Constant value: **8**

COL0_YELLOW

```
public static final int COL0_YELLOW
```

Constant value: **10**

COL0_MAGENTA

```
public static final int COL0_MAGENTA
```

Constant value: **12**

CHANNEL1

```
public static final int CHANNEL1
```

Constant value: **0**

CHANNEL2

```
public static final int CHANNEL2
```

Constant value: **1**

DONOTHING_SHORT

```
public static final short DONOTHING_SHORT
```

Constant value: **-32640**

DONOTHING_BYTE

```
public static final byte DONOTHING_BYTE
```

Constant value: **-128**

(continued from last page)

CONTROLCODES_RCL

```
public static final short CONTROLCODES_RCL
```

RCL Resume caption loading [channel]

CONTROLCODES_BS

```
public static final short CONTROLCODES_BS
```

BS Backspace [channel]

CONTROLCODES_AOF

```
public static final short CONTROLCODES_AOF
```

AOF Reserved (formerly Alarm Off) [channel]

CONTROLCODES_AON

```
public static final short CONTROLCODES_AON
```

AON Reserved (formerly Alarm On) [channel]

CONTROLCODES_DER

```
public static final short CONTROLCODES_DER
```

DER Delete to End of Row [channel]

CONTROLCODES_RU2

```
public static final short CONTROLCODES_RU2
```

RU2 Roll-Up Captions-2 Rows [channel]

CONTROLCODES_RU3

```
public static final short CONTROLCODES_RU3
```

RU3 Roll-Up Captions-3 Rows [channel]

CONTROLCODES_RU4

```
public static final short CONTROLCODES_RU4
```

RU4 Roll-Up Captions-4 Rows [channel]

CONTROLCODES_FON

```
public static final short CONTROLCODES_FON
```

FON Flash On [channel]

CONTROLCODES_RDC

```
public static final short CONTROLCODES_RDC
```

(continued from last page)

RDC Resume Direct Captioning [channel]

CONTROLCODES_TR

public static final short **CONTROLCODES_TR**

TR Text Restart [channel]

CONTROLCODES_RTD

public static final short **CONTROLCODES_RTD**

RTD Resume Text Display [channel]

CONTROLCODES_EDM

public static final short **CONTROLCODES_EDM**

EDM Erase Displayed Memory [channel]

CONTROLCODES_CR

public static final short **CONTROLCODES_CR**

CR Carriage Return [channel]

CONTROLCODES_ENM

public static final short **CONTROLCODES_ENM**

ENM Erase Non-Displayed Memory [channel]

CONTROLCODES_EOC

public static final short **CONTROLCODES_EOC**

EOC End of Caption (Flip Memories) [channel]

CONTROLCODES_TO1

public static final short **CONTROLCODES_TO1**

TO1 Tab Offset 1 Column [channel]

CONTROLCODES_TO2

public static final short **CONTROLCODES_TO2**

TO2 Tab Offset 2 Columns [channel]

CONTROLCODES_TO3

public static final short **CONTROLCODES_TO3**

TO3 Tab Offset 3 Columns [channel]

(continued from last page)

CONTROLCODES_BWO

```
public static final short CONTROLCODES_BWO
```

BWO Background White, Opaque [channel]

CONTROLCODES_BWS

```
public static final short CONTROLCODES_BWS
```

BWS Background White, Semi-transparent [channel]

CONTROLCODES_BGO

```
public static final short CONTROLCODES_BGO
```

BGO Background Green, Opaque [channel]

CONTROLCODES_BGS

```
public static final short CONTROLCODES_BGS
```

BGS Background Green, Semi-transparent [channel]

CONTROLCODES_BBO

```
public static final short CONTROLCODES_BBO
```

BBO Background Blue, Opaque [channel]

CONTROLCODES_BBS

```
public static final short CONTROLCODES_BBS
```

BBS Background Blue, Semi-transparent [channel]

CONTROLCODES_BCO

```
public static final short CONTROLCODES_BCO
```

BCO Background Cyan, Opaque [channel]

CONTROLCODES_BCS

```
public static final short CONTROLCODES_BCS
```

BCS Background Cyan, Semi-transparent [channel]

CONTROLCODES_BRO

```
public static final short CONTROLCODES_BRO
```

BRO Background Red, Opaque [channel]

CONTROLCODES_BRS

```
public static final short CONTROLCODES_BRS
```

(continued from last page)

BRS Background Red, Semi-transparent [channel]

CONTROLCODES_BYO

public static final short **CONTROLCODES_BYO**

BYO Background Yellow, Opaque [channel]

CONTROLCODES_BYS

public static final short **CONTROLCODES_BYS**

BYS Background Yellow, Semi-transparent [channel]

CONTROLCODES_BMO

public static final short **CONTROLCODES_BMO**

BMO Background Magenta, Opaque [channel]

CONTROLCODES_BMS

public static final short **CONTROLCODES_BMS**

BMS Background Magenta, Semi-transparent [channel]

CONTROLCODES_BAO

public static final short **CONTROLCODES_BAO**

BAO Background Black, Opaque [channel]

CONTROLCODES_BAS

public static final short **CONTROLCODES_BAS**

BAS Background Black, Semi-transparent [channel]

CONTROLCODES_BT

public static final short **CONTROLCODES_BT**

BT Background Transparent [channel]

CONTROLCODES_FA

public static final short **CONTROLCODES_FA**

FA Foreground Black [channel]

CONTROLCODES_FAU

public static final short **CONTROLCODES_FAU**

FAU Foreground Black Underline [channel]

(continued from last page)

LANG_ASCII

public static final short **LANG_ASCII**

Select the standard line 21 character set in normal size. [channel]

LANG_ASCII_DBL

public static final short **LANG_ASCII_DBL**

Select the standard line 21 character set in double size. [channel]

LANG_PRIV1

public static final short **LANG_PRIV1**

Select the first private character set. [channel]

LANG_PRIV2

public static final short **LANG_PRIV2**

Select the second private character set. [channel]

LANG_GB231280

public static final short **LANG_GB231280**

Select the People's Republic of China character set: GB 2312-80. [channel]

LANG_KSC56011987

public static final short **LANG_KSC56011987**

Select the Korean Standard character set: KSC 5601-1987. [channel]

LANG_REG1

public static final short **LANG_REG1**

Select the first registered character set. [channel]

COUNT_ROWS

public static final int **COUNT_ROWS**

Total number of rows of characters per frame
Constant value: **15**

COUNT_COLS

public static final int **COUNT_COLS**

Total number of columns per frame
Constant value: **31**

(continued from last page)

COL0_HIBYTE

```
public static final short COL0_HIBYTE
```

Move to set color (high-byte) [channel][row]

COL0_LOWBYTE

```
public static final short COL0_LOWBYTE
```

Move to set color (low-byte) [color][row]

COL_0

```
public static final int COL_0
```

Constant value: **0**

COL_4

```
public static final int COL_4
```

Constant value: **2**

COL_5

```
public static final int COL_5
```

Constant value: **4**

COL_12

```
public static final int COL_12
```

Constant value: **6**

COL_16

```
public static final int COL_16
```

Constant value: **8**

COL_20

```
public static final int COL_20
```

Constant value: **10**

COL_24

```
public static final int COL_24
```

(continued from last page)

Constant value: **12**

COL_28

```
public static final int COL_28
```

Constant value: **14**

COLN_LOWBYTE

```
public static final short COLN_LOWBYTE
```

FORMAT_COLOR

```
public static final short FORMAT_COLOR
```

FORMAT_ITALICS

```
public static final short FORMAT_ITALICS
```

Constant value: **-28242**

FORMAT_ITALICS_UNDERLINE

```
public static final short FORMAT_ITALICS_UNDERLINE
```

Constant value: **-28369**

FORMAT_FLASH

```
public static final short FORMAT_FLASH
```

Constant value: **-27480**

UTF8_NOTMATCH

```
public static final byte UTF8_NOTMATCH
```

UTF8_PERIOD

```
public static final byte UTF8_PERIOD
```

Constant value: **46**

UTF8_SINGLEQUOTE

```
public static final byte UTF8_SINGLEQUOTE
```

(continued from last page)

Constant value: **39**

UTF8_DOUBLEQUOTE

```
public static final byte UTF8_DOUBLEQUOTE
```

Constant value: **34**

UTF8_DASH

```
public static final byte UTF8_DASH
```

Constant value: **45**

UTF8_UNDERSCORE

```
public static final byte UTF8_UNDERSCORE
```

Constant value: **84**

UTF8_LEFTPAREN

```
public static final byte UTF8_LEFTPAREN
```

Constant value: **40**

UTF8_RIGHTPAREN

```
public static final byte UTF8_RIGHTPAREN
```

Constant value: **41**

UTF8_SPACE

```
public static final byte UTF8_SPACE
```

Constant value: **32**

UTF8_EXCLAMATION

```
public static final byte UTF8_EXCLAMATION
```

Constant value: **33**

UTF8_QUESTION

```
public static final byte UTF8_QUESTION
```

Constant value: **63**

(continued from last page)

UTF8_LESSTHAN

public static final byte **UTF8_LESSTHAN**

Constant value: **60**

UTF8_GREATERTHAN

public static final byte **UTF8_GREATERTHAN**

Constant value: **62**

UTF8_MINUSHYPHEN

public static final byte **UTF8_MINUSHYPHEN**

Constant value: **45**

UTF8_PIPE

public static final byte **UTF8_PIPE**

Constant value: **124**

UTF8_LOWER_A

public static final byte **UTF8_LOWER_A**

Constant value: **97**

UTF8_LOWER_B

public static final byte **UTF8_LOWER_B**

Constant value: **98**

UTF8_LOWER_C

public static final byte **UTF8_LOWER_C**

Constant value: **99**

UTF8_LOWER_D

public static final byte **UTF8_LOWER_D**

Constant value: **100**

UTF8_LOWER_E

public static final byte **UTF8_LOWER_E**

(continued from last page)

Constant value: **101**

UTF8_LOWER_Fpublic static final byte **UTF8_LOWER_F**Constant value: **102**

UTF8_LOWER_Gpublic static final byte **UTF8_LOWER_G**Constant value: **103**

UTF8_LOWER_Hpublic static final byte **UTF8_LOWER_H**Constant value: **104**

UTF8_LOWER_Ipublic static final byte **UTF8_LOWER_I**Constant value: **105**

UTF8_LOWER_Jpublic static final byte **UTF8_LOWER_J**Constant value: **106**

UTF8_LOWER_Kpublic static final byte **UTF8_LOWER_K**Constant value: **107**

UTF8_LOWER_Lpublic static final byte **UTF8_LOWER_L**Constant value: **108**

UTF8_LOWER_Mpublic static final byte **UTF8_LOWER_M**Constant value: **109**

UTF8_LOWER_N

public static final byte **UTF8_LOWER_N**

Constant value: **110**

UTF8_LOWER_O

public static final byte **UTF8_LOWER_O**

Constant value: **111**

UTF8_LOWER_P

public static final byte **UTF8_LOWER_P**

Constant value: **112**

UTF8_LOWER_Q

public static final byte **UTF8_LOWER_Q**

Constant value: **113**

UTF8_LOWER_R

public static final byte **UTF8_LOWER_R**

Constant value: **114**

UTF8_LOWER_S

public static final byte **UTF8_LOWER_S**

Constant value: **115**

UTF8_LOWER_T

public static final byte **UTF8_LOWER_T**

Constant value: **116**

UTF8_LOWER_U

public static final byte **UTF8_LOWER_U**

Constant value: **117**

(continued from last page)

UTF8_LOWER_V

```
public static final byte UTF8_LOWER_V
```

Constant value: **118**

UTF8_LOWER_W

```
public static final byte UTF8_LOWER_W
```

Constant value: **119**

UTF8_LOWER_X

```
public static final byte UTF8_LOWER_X
```

Constant value: **120**

UTF8_LOWER_Y

```
public static final byte UTF8_LOWER_Y
```

Constant value: **121**

UTF8_LOWER_Z

```
public static final byte UTF8_LOWER_Z
```

Constant value: **122**

UTF8_UPPER_A

```
public static final byte UTF8_UPPER_A
```

Constant value: **65**

UTF8_UPPER_B

```
public static final byte UTF8_UPPER_B
```

Constant value: **66**

UTF8_UPPER_C

```
public static final byte UTF8_UPPER_C
```

Constant value: **67**

UTF8_UPPER_D

```
public static final byte UTF8_UPPER_D
```

(continued from last page)

Constant value: **68**

UTF8_UPPER_Epublic static final byte **UTF8_UPPER_E**Constant value: **69**

UTF8_UPPER_Fpublic static final byte **UTF8_UPPER_F**Constant value: **70**

UTF8_UPPER_Gpublic static final byte **UTF8_UPPER_G**Constant value: **71**

UTF8_UPPER_Hpublic static final byte **UTF8_UPPER_H**Constant value: **72**

UTF8_UPPER_Ipublic static final byte **UTF8_UPPER_I**Constant value: **73**

UTF8_UPPER_Jpublic static final byte **UTF8_UPPER_J**Constant value: **74**

UTF8_UPPER_Kpublic static final byte **UTF8_UPPER_K**Constant value: **75**

UTF8_UPPER_Lpublic static final byte **UTF8_UPPER_L**Constant value: **76**

UTF8_UPPER_M

public static final byte **UTF8_UPPER_M**

Constant value: **77**

UTF8_UPPER_N

public static final byte **UTF8_UPPER_N**

Constant value: **78**

UTF8_UPPER_O

public static final byte **UTF8_UPPER_O**

Constant value: **79**

UTF8_UPPER_P

public static final byte **UTF8_UPPER_P**

Constant value: **80**

UTF8_UPPER_Q

public static final byte **UTF8_UPPER_Q**

Constant value: **81**

UTF8_UPPER_R

public static final byte **UTF8_UPPER_R**

Constant value: **82**

UTF8_UPPER_S

public static final byte **UTF8_UPPER_S**

Constant value: **83**

UTF8_UPPER_T

public static final byte **UTF8_UPPER_T**

Constant value: **84**

(continued from last page)

UTF8_UPPER_U

```
public static final byte UTF8_UPPER_U
```

Constant value: **85**

UTF8_UPPER_V

```
public static final byte UTF8_UPPER_V
```

Constant value: **86**

UTF8_UPPER_W

```
public static final byte UTF8_UPPER_W
```

Constant value: **87**

UTF8_UPPER_X

```
public static final byte UTF8_UPPER_X
```

Constant value: **88**

UTF8_UPPER_Y

```
public static final byte UTF8_UPPER_Y
```

Constant value: **89**

UTF8_UPPER_Z

```
public static final byte UTF8_UPPER_Z
```

Constant value: **90**

UTF8_LOWER_A_WITH_CIRCUMFLEX

```
public static final char UTF8_LOWER_A_WITH_CIRCUMFLEX
```

Constant value: **226**

UTF8_LOWER_E_WITH_CIRCUMFLEX

```
public static final char UTF8_LOWER_E_WITH_CIRCUMFLEX
```

Constant value: **234**

UTF8_LOWER_I_WITH_CIRCUMFLEX

```
public static final char UTF8_LOWER_I_WITH_CIRCUMFLEX
```

(continued from last page)

Constant value: **238**

UTF8_LOWER_O_WITH_CIRCUMFLEX

```
public static final char UTF8_LOWER_O_WITH_CIRCUMFLEX
```

Constant value: **244**

UTF8_LOWER_U_WITH_CIRCUMFLEX

```
public static final char UTF8_LOWER_U_WITH_CIRCUMFLEX
```

Constant value: **251**

UTF8_LOWER_A_WITH_GRAVE

```
public static final char UTF8_LOWER_A_WITH_GRAVE
```

Constant value: **224**

UTF8_LOWER_E_WITH_GRAVE

```
public static final char UTF8_LOWER_E_WITH_GRAVE
```

Constant value: **232**

UTF8_NBSP

```
public static final char UTF8_NBSP
```

Constant value: **160**

UTF8_MUSICAL_NOTE

```
public static final char UTF8_MUSICAL_NOTE
```

Constant value: **9834**

UTF8_BRITISH_POUND

```
public static final char UTF8_BRITISH_POUND
```

Constant value: **163**

UTF8_CENTS

```
public static final char UTF8_CENTS
```

Constant value: **162**

UTF8_TRADEMARK

public static final char **UTF8_TRADEMARK**

Constant value: **8482**

UTF8_REVERSE_QUERY

public static final char **UTF8_REVERSE_QUERY**

Constant value: **191**

UTF8_ONE_HALF

public static final char **UTF8_ONE_HALF**

Constant value: **189**

CEA608_MUSICAL_NOTE

public static byte **CEA608_MUSICAL_NOTE**

CEA608_SOLID_BLOCK

public static byte **CEA608_SOLID_BLOCK**

UTF8MAP_INDEX_OFFSET

public static final int **UTF8MAP_INDEX_OFFSET**

Constant value: **32**

UTF8MAP

public static final byte **UTF8MAP**

CEA608_EXTENDED_NA_MAP

public static final char **CEA608_EXTENDED_NA_MAP**

CEA608_EXTENDED_FR_SP_MAP

public static final char **CEA608_EXTENDED_FR_SP_MAP**

CEA608_EXTENDED_PORT_GERMAN_DANISH_MAP

```
public static final char CEA608_EXTENDED_PORT_GERMAN_DANISH_MAP
```

Constructors

ClosedCaptionCEA608Utils

```
public ClosedCaptionCEA608Utils()
```

Methods

charUTF8ToCAE608

```
public static byte[] charUTF8ToCAE608(CEACaptionConversionContext ceaContext,  
    String text,  
    int pos,  
    String characterSet,  
    int channel)
```

charUTF8ToCAE608

```
public static byte[] charUTF8ToCAE608(String text,  
    int pos,  
    String characterSet,  
    int channel)
```

cea608CharToCharUTF8

```
public static char cea608CharToCharUTF8(short b)
```

calcParity

```
public static byte calcParity(byte inByte)
```

calcParity

```
public static short calcParity(short inShort)
```

checkParity

```
public static boolean checkParity(byte inByte)
```

checkParity

```
public static boolean checkParity(short inShort)
```

cea608ExtendedCharacterToUTF8

```
public static Character cea608ExtendedCharacterToUTF8(short command)
```

Package

com.wowza.wms.timedtext.model

com.wowza.wms.timedtext.model Interface ITimedTextConstants

public interface **ITimedTextConstants**
extends **ITimedTextPrivateConstants**

Field Summary

public static final	AMF_TEXTTYPE_ON_TEXT_DATA Value: onTextData
public static final	CEA608 Value: CEA608
public static final	DEFAULT_AMF_CONVERTER_TEXT_TYPE Value: onTextData
public static final	DEFAULT_AMF_CONVERTER_TRACK_INDEX Value: 99
public static final	DEFAULT_CAPTION_CHARACTER_SET Default value for Timed Text Property "captionCharacterSet". Value: UTF-8
public static final	DEFAULT_MAXIMUM_CAPTION_DURATION Default value for Timed Text Property "maximumCaptionDuration". Value: 5000
public static final	DEFAULT_PROPERTY_CAPTION_LANGUAGES Default value for Timed Text Property "captionLanguages". Value: *
public static final	DEFAULT_PROPERTY_CAPTION_FILE_NAMING_RULE Default value for Timed Text Property "captionFileNamingRule". Value: \${SourceStreamName}.\${FileExtension}
public static final	DEFAULT_PROPERTY_CAPTION_PATH_NAMING_RULE Default value for Timed Text Property "captionFileNamingRule". Value: \${SourceBasePath}
public static final	LANGUAGE_ID_ENGLISH Value: eng
public static final	LANGUAGE_LOCALE Value: locale
public static final	ON_CAPTION Value: onCaption

public static final	ON_CAPTION_INFO Value: onCaptionInfo
public static final	ON_TEXT_DATA Value: onTextData
public static final	OUTPUT_TYPE_AMF Value: AMF
public static final	OUTPUT_TYPE_CEA608 Value: CEA608
public static final	OUTPUT_TYPE_TTML Value: TTML
public static final	PROP_3GPP_CAPTION_CHARACTER_SET Timed Text Property "reader3GPPCaptionCharacterSet": used to determine character set when ingesting TTML captions Overrides general property "captionCharacterSet" Value: reader3GPPCaptionCharacterSet
public static final	PROP_AMF_CONVERTER_INSERT_ERASES_IN_GAPS Timed Text Property "amfCaptionConverterInsertErases": insert erase captions if gaps between captions are greater than PROP_MAXIMUM_CAPTION_DURATION Value: amfCaptionConverterInsertErases
public static final	PROP_AMF_CONVERTER_TEXT_TYPE Timed Text Property "amfCaptionConverterTextType": AMF text type used in conversion Value: amfCaptionConverterTextType
public static final	PROP_AMF_CONVERTER_TRACK_INDEX Timed Text Property "amfCaptionConverterTrackIndex": track index used for captions Value: amfCaptionConverterTrackIndex
public static final	PROP_CAPTION_CHARACTER_SET Timed Text Property "captionCharacterSet": used to determine character set when ingesting captions Value: captionCharacterSet
public static final	PROP_CAPTION_FILE_NAMING_RULE Timed Text Property "captionFileNamingRule": to change logic determining the caption companion file naming scheme Value: captionFileNamingRule
public static final	PROP_CAPTION_FILENAME_QUERY_PARAMETER HTTP Streamer Property "captionFilenameQueryParameter": used for changing query string param used for caption file selection. Value: captionFilenameQueryParameter
public static final	PROP_CAPTION_LANGUAGE_QUERY_PARAMETER HTTP Streamer Property "captionLanguageQueryParameter": used for changing query string param used for caption language selection. Value: captionLanguageQueryParameter

public static final	<u>PROP_CAPTION_LANGUAGES</u> Timed Text Property "captionLanguages": used to select presented caption Language. Value: captionLanguages
public static final	<u>PROP_CAPTION_PATH_NAMING_RULE</u> Timed Text Property "captionPathNamingRule": to change logic determining the caption companion path Value: captionPathNamingRule
public static final	<u>PROP_CEA608_CONVERTER_CHARACTER_SET</u> Timed Text Property "cea608CaptionConverterCharacterSet": character set in default CEA caption converter. Value: cea608CaptionConverterCharacterSet
public static final	<u>PROP_CEA608_CONVERTER_CHARS_PER_LINE</u> Timed Text Property "cea608CaptionConverterCharsPerLine": max number characters per line in default CEA caption converter. Value: cea608CaptionConverterCharsPerLine
public static final	<u>PROP_CEA608_CONVERTER_COLOR</u> Timed Text Property "cea608CaptionConverterColor": color of text in default CEA caption converter. Value: cea608CaptionConverterColor
public static final	<u>PROP_CUPERTINO_CAPTION_LANGUAGES</u> Timed Text Property "cupertinoCaptionLanguages": to enabled captionLanguage selection for Cupertino streaming. Value: cupertinoCaptionLanguages
public static final	<u>PROP_CUPERTINO_LIVE_USE_WEBVTT</u> Timed Text Property "cupertinoLiveCaptionsUseWebVTT" Value: cupertinoLiveCaptionsUseWebVTT
public static final	<u>PROP_CUPERTINO_VOD_CAPTIONS_ENABLED</u> Timed Text Property "cupertinoVODCaptionsEnabled": to enabled VOD captions for Cupertino streaming. Value: cupertinoVODCaptionsEnabled
public static final	<u>PROP_CUPERTINO_VOD_REMOVE_EXISTING_CEA608</u> Timed Text Property "cupertinoVODCaptionsRemoveExistingCEA608": if existing VOD has 608, remove them Value: cupertinoVODCaptionsRemoveExistingCEA608
public static final	<u>PROP_CUPERTINO_VOD_USE_WEBVTT</u> Timed Text Property "cupertinoVODCaptionsUseWebVTT" Value: cupertinoVODCaptionsUseWebVTT
public static final	<u>PROP_DEBUG_3GPP_CAPTION_PARSER</u> Timed Text Property "debug3GPPCaptionParser": adds info level logging regarding the 3GPP parser Value: debug3GPPCaptionParser
public static final	<u>PROP_DEBUG_3GPP_CAPTION_PARSER_TIME</u> Timed Text Property "debug3GPPCaptionParserTime": adds info level logging regarding how long it took parser to index Value: debug3GPPCaptionParserTime

public static final	<u>PROP_DEBUG_CUPERTINO_VOD_CAPTION_LANGUAGE_SELECTION</u> Timed Text Property "cupertinoDebugVODCaptionLanguageSelection": adds info level logging regarding language selection for Cupertino Streaming Value: cupertinoDebugVODCaptionLanguageSelection
public static final	<u>PROP_DEBUG_CUPERTINO_VOD_CAPTION_PROVIDER_DETERMINATION</u> Timed Text Property "cupertinoDebugVODCaptionProviderDetermination": adds info level logging regarding caption provider determination for a given VOD asset for Cupertino Streaming. Value: cupertinoDebugVODCaptionProviderDetermination
public static final	<u>PROP_DEBUG_LIVE_CAPTION_INGEST</u> Timed Text Property "debugCaptionLiveIngest" Value: debugCaptionLiveIngest
public static final	<u>PROP_DEBUG_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608_LOG_AS_RAW</u> Timed Text Property "debugCaptionLiveIngestOnCaptionCEA608LogAsRaw" : Adds additional logging of decoding AMF onCaption events of type "708" (CEA-608) such that they are logged in a raw binary state Value: debugCaptionLiveIngestOnCaptionCEA608LogAsRaw
public static final	<u>PROP_DEBUG_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608_LOG_AS_SCC</u> Timed Text Property "debugCaptionLiveIngestOnCaptionCEA608LogAsSCC" : Adds additional logging of decoding AMF onCaption events of type "708" (CEA-608) such that they are logged in SCC format Value: debugCaptionLiveIngestOnCaptionCEA608LogAsSCC
public static final	<u>PROP_DEBUG_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608_VERBOSE</u> Timed Text Property "debugCaptionLiveIngestOnCaptionCEA608Verbose" : Adds verbose logging to AMF onCaption events of type "708" (CEA-608) Value: debugCaptionLiveIngestOnCaptionCEA608Verbose
public static final	<u>PROP_DEBUG_LIVE_CAPTION_ONCAPTION_INGEST</u> Timed Text Property "debugCaptionLiveOnCaptionIngest" Value: debugCaptionLiveOnCaptionIngest
public static final	<u>PROP_DEBUG_LIVE_CAPTION_ONTEXTDATA_INGEST</u> Timed Text Property "debugCaptionLiveOnTextDataIngest" Value: debugCaptionLiveOnTextDataIngest
public static final	<u>PROP_DEBUG_RTMP_VOD_CAPTION_LANGUAGE_SELECTION</u> Timed Text Property "rtmpDebugVODCaptionLanguageSelection": adds info level logging regarding language selection for RTMP Streaming Value: rtmpDebugVODCaptionLanguageSelection
public static final	<u>PROP_DEBUG_RTMP_VOD_CAPTION_PROVIDER_DETERMINATION</u> Timed Text Property "rtmpDebugVODCaptionProviderDetermination": adds info level logging regarding caption provider determination for a given VOD asset for RTMP Streaming. Value: rtmpDebugVODCaptionProviderDetermination
public static final	<u>PROP_DEBUG_SANJOSE_VOD_CAPTION_LANGUAGE_SELECTION</u> Timed Text Property "sanJoseDebugVODCaptionLanguageSelection": adds info level logging regarding language selection for San Jose Streaming Value: sanJoseDebugVODCaptionLanguageSelection

public static final	<u>PROP_DEBUG_SANJOSE_VOD_CAPTION_PROVIDER_DETERMINATION</u> Timed Text Property "sanJoseVODDebugCaptionProviderDetermination": adds info level logging regarding caption provider determination for a given VOD asset for San Jose Streaming. Value: sanJoseVODDebugCaptionProviderDetermination
public static final	<u>PROP_DEBUG_SCC_CAPTION_PARSER</u> Timed Text Property "debugSCCCaptionParser": adds info level logging regarding the SCC caption parser Value: debugSCCCaptionParser
public static final	<u>PROP_DEBUG_SCC_CAPTION_PARSER_TIME</u> Timed Text Property "debugSCCCaptionParserTime": adds info level logging regarding how long it took parser to index Value: debugSCCCaptionParserTime
public static final	<u>PROP_DEBUG_SCC_VOD_CAPTION_COMPANION_FILES</u> Timed Text Property "debugSCCVODCaptionFileDetermination": adds info level logging regarding caption companion file determination for a given VOD asset for SCC provider Value: debugSCCVODCaptionFileDetermination
public static final	<u>PROP_DEBUG_SRT_CAPTION_PARSER</u> Timed Text Property "debugSRTCaptionParser": adds info level logging regarding the SRT caption parser Value: debugSRTCaptionParser
public static final	<u>PROP_DEBUG_SRT_CAPTION_PARSER_TIME</u> Timed Text Property "debugSRTCaptionParserTime": adds info level logging regarding how long it took parser to index Value: debugSRTCaptionParserTime
public static final	<u>PROP_DEBUG_SRT_VOD_CAPTION_COMPANION_FILES</u> Timed Text Property "debugSRTVODCaptionFileDetermination": adds info level logging regarding caption companion file determination for a given VOD asset for SRT provider Value: debugSRTVODCaptionFileDetermination
public static final	<u>PROP_DEBUG_TTML_CAPTION_PARSER</u> Timed Text Property "debugTTMLCaptionParser": adds info level logging regarding the TTML caption parser Value: debugTTMLCaptionParser
public static final	<u>PROP_DEBUG_TTML_CAPTION_PARSER_TIME</u> Timed Text Property "debugTTMLCaptionParserTime": adds info level logging regarding how long it took parser to index Value: debugTTMLCaptionParserTime
public static final	<u>PROP_DEBUG_TTML_VOD_CAPTION_COMPANION_FILES</u> Timed Text Property "debugTTMLVODCaptionFileDetermination": adds info level logging regarding caption companion file determination for a given VOD asset for TTML Streaming. Value: debugTTMLVODCaptionFileDetermination
public static final	<u>PROP_DEBUG_VOD_CAPTION_COMPANION_FILES</u> Timed Text Property "debugVODCaptionFileDetermination": adds info level logging regarding caption companion file determination for a given VOD asset Value: debugVODCaptionFileDetermination

public static final	<u>PROP_DEBUG_VOD_CAPTION_LANGUAGE_SELECTION</u> Timed Text Property "debugVODCaptionLanguageSelection": adds info level logging regarding language selection Value: debugVODCaptionLanguageSelection
public static final	<u>PROP_DEBUG_VOD_CAPTION_PROVIDER_DETERMINATION</u> Timed Text Property "debugVODCaptionProviderDetermination": adds info level logging regarding caption provider determination for a given VOD asset Value: debugVODCaptionProviderDetermination
public static final	<u>PROP_DEBUG_WEBVTT_CAPTION_PARSER</u> Timed Text Property "debugWebVTTCaptionParser": adds info level logging regarding the WebVTT caption parser Value: debugWebVTTCaptionParser
public static final	<u>PROP_DEBUG_WEBVTT_CAPTION_PARSER_TIME</u> Timed Text Property "debugWebVTTCaptionParserTime": adds info level logging regarding how long it took parser to index Value: debugWebVTTCaptionParserTime
public static final	<u>PROP_DEBUG_WEBVTT_VOD_CAPTION_COMPANION_FILES</u> Timed Text Property "debugWebVTTVODCaptionFileDetermination": adds info level logging regarding caption companion file determination for a given VOD asset for WebVTT provider Value: debugWebVTTVODCaptionFileDetermination
public static final	<u>PROP_LIVE_CAPTION_DEFAULT_LANGUAGES</u> Timed Text Property "captionLiveIngestLanguages": comma separated ISO-639 language codes determining what languages are expected in the live stream Value: captionLiveIngestLanguages
public static final	<u>PROP_LIVE_CAPTION_INGEST_AMF_ID_STRING</u> Timed Text Property "captionLiveIngestAMFIDString": to change AMF identifier from default "onTextData" to something else Value: captionLiveIngestAMFIDString
public static final	<u>PROP_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608_ENABLE_FIELD1</u> Timed Text Property "captionLiveIngestOnCaptionCEA608EnableField2": When ingesting AMF onCaption 708 data events (actually CEA-608), controls whether CEA-608 field 1 is enabled for decoding Value: captionLiveIngestOnCaptionCEA608EnableField1
public static final	<u>PROP_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608_ENABLE_FIELD2</u> Timed Text Property "captionLiveIngestOnCaptionCEA608EnableField2": When ingesting AMF onCaption 708 data events (actually CEA-608), controls whether CEA-608 field 2 is enabled for decoding Value: captionLiveIngestOnCaptionCEA608EnableField2
public static final	<u>PROP_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608_IDLE_TIME_TO_FLUSH_DECODER</u> Timed Text Property "captionLiveIngestOnCaptionCEA608IdleTimeToFlushDecoder" Controls allowed amount of idle time when parsing CEA-608 captions when ingested from AMF 708 data events Value: captionLiveIngestOnCaptionCEA608IdleTimeToFlushDecoder
public static final	<u>PROP_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608_LANGUAGE_ID</u> Timed Text Property "captionLiveIngestOnCaptionCEA608LanguageID": When ingesting AMF onCaption 708 data events (actually CEA-608), controls the language ID used to add captions as onTextData Value: captionLiveIngestOnCaptionCEA608LanguageID

public static final	<u>PROP_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608_MAX_WARN_MESSAGES</u> Timed Text Property "captionLiveIngestOnCaptionCEA608MaxWarnMessages": When ingesting AMF onCaption 708 data events (actually CEA-608), controls the maximum number of warning messages that are logged by the decoder Value: captionLiveIngestOnCaptionCEA608MaxWarnMessages
public static final	<u>PROP_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608_TRACK_NUMBER</u> Timed Text Property "captionLiveIngestOnCaptionCEA608TrackNumber": When ingesting AMF onCaption 708 data events (actually CEA-608), controls the track number used to add captions as onTextData Value: captionLiveIngestOnCaptionCEA608TrackNumber
public static final	<u>PROP_LIVE_CAPTION_INGEST_ON_CAPTION_INFO_STRING</u> Timed Text Property "captionLiveIngestOnCaptionInfoString": to change AMF identifier from default "onCaptionInfo" to something else Value: captionLiveIngestOnCaptionInfoString
public static final	<u>PROP_LIVE_CAPTION_INGEST_ON_CAPTION_STRING</u> Timed Text Property "captionLiveIngestOnCaptionString": to change AMF identifier from default "onCaption" to something else Value: captionLiveIngestOnCaptionString
public static final	<u>PROP_LIVE_CAPTION_INGEST_TYPE</u> Timed Text Property "captionLiveIngestType": type of live caption ingestion. Value: captionLiveIngestType
public static final	<u>PROP_LIVE_CAPTION_USE_STREAM_NAME_GROUPS</u> Timed Text Property "captionLiveIngestUseStreamNameGroups": to enable use of smil files to determine available languages and ingest type Value: captionLiveIngestUseStreamNameGroups
public static final	<u>PROP_MAXIMUM_CAPTION_DURATION</u> Timed Text Property "maximumCaptionDuration": used to determine maximum caption duration Value: maximumCaptionDuration
public static final	<u>PROP_RTMP_CAPTION_LANGUAGES</u> Timed Text Property "rtmpCaptionLanguages": to enabled captionLanguage selection for RTMP Type String. Value: rtmpCaptionLanguages
public static final	<u>PROP_RTMP_VOD_CAPTIONS_ENABLED</u> Timed Text Property "rtmpVODCaptionsEnabled": to enabled VOD captions for RTMP streaming. Value: rtmpVODCaptionsEnabled
public static final	<u>PROP_SANJOSE_CAPTION_LANGUAGES</u> Timed Text Property "sanJoseCaptionLanguages": to enabled captionLanguage selection for San Jose streaming. Value: sanJoseCaptionLanguages
public static final	<u>PROP_SANJOSE_VOD_CAPTIONS_ENABLED</u> Timed Text Property "sanJoseVODCaptionsEnabled": to enabled VOD captions for San Jose streaming. Value: sanJoseVODCaptionsEnabled

public static final	PROP_SRT_CAPTION_CHARACTER_SET Timed Text Property "srtReaderCaptionCharacterSet": used to determine character set when ingesting SRT captions Overrides general property "captionCharacterSet" Value: srtReaderCaptionCharacterSet
public static final	PROP_TTML_CAPTION_CHARACTER_SET Timed Text Property "ttmlReaderCaptionCharacterSet": used to determine character set when ingesting TTML captions Overrides general property "captionCharacterSet" Value: ttmlReaderCaptionCharacterSet
public static final	PROP_TTML_INLINE_TAGS_IN_TEXT Timed Text Property "ttmlInlineTagsInText ": TTML parser includes in-line styling and line-break tags in the sample text Value: ttmlInlineTagsInText
public static final	PROP_UNDEFINED_LANG_CODE Timed Text Property "captionUndefinedLanguageId": if incoming caption language is undefined, use this language ID. Value: captionUndefinedLanguageId
public static final	PROP_VOD_CAPTIONS_ENABLED Timed Text Property "vodCaptionsEnabled": to enabled VOD captions for all supported protocols Value: vodCaptionsEnabled
public static final	PROP_WEBVTT_CONVERTER_INSERT_ERASES_IN_GAPS Timed Text Property "webVTTCaptionConverterInsertErases": insert erase captions if gaps between captions are greater than PROP_MAXIMUM_CAPTION_DURATION Value: webVTTCaptionConverterInsertErases
public static final	QUERYSTR_CAPTION_FILENAME Query string used for choosing caption languages. Value: wowzacaptionfile
public static final	QUERYSTR_CAPTIONLANGUAGES Query string used for choosing caption languages. Value: wowzacaptionlanguages
public static final	SMIL_PARAM_IS_WOWZA_CAPTION_STREAM Value: iswowzacaptionstream
public static final	SMIL_PARAM_WOWZA_CAPTION_INGEST_TYPE Value: wowzacaptioningesttype
public static final	SOURCE_TYPE_3GPP Value: 3GPP
public static final	SOURCE_TYPE_AMF Value: AMF
public static final	SOURCE_TYPE_SCC Value: SCC
public static final	SOURCE_TYPE_SRT Value: SRT

public static final	SOURCE_TYPE_TTML Value: TTML
public static final	SOURCE_TYPE_WEBVTT Value: WebVTT
public static final	TIMED_TEXT_TYPE_CAPTION Value: Caption

Fields inherited from interface com.wowza.wms.timedtext.model.ITimedTextPrivateConstants

```

DEFAULT_AMF_CONVERTER_CLASS, DEFAULT_CEA608_CONVERTER_CLASS,
DEFAULT_CEA608_IDENTITY_CONVERTER_CLASS,
DEFAULT_PROPERTY_CUPERTINO_VOD_CAPTIONS_DELEGATE_CLASS,
DEFAULT_PROPERTY_RTMP_VOD_CAPTIONS_DELEGATE_CLASS,
DEFAULT_PROPERTY_SANJOSE_VOD_CAPTIONS_DELEGATE_CLASS, DEFAULT_TIMED_TEXT_READER_SCC,
DEFAULT_TIMED_TEXT_READER_SRT, DEFAULT_TIMED_TEXT_READER_TTML,
DEFAULT_TIMED_TEXT_READER_WEBVTT, DEFAULT_WEBVTT_CONVERTER_CLASS,
PROP_AMF_CONVERTER_CLASS, PROP_CEA608_CONVERTER_CLASS,
PROP_CEA608_MAX_COMMANDS_PER_FRAME, PROP_CUPERTINO_VOD_CAPTIONS_DELEGATE_CLASS,
PROP_LIVE_ONCAPTION_TO_AMF_CONVERTER_CLASS,
PROP_LIVE_ONCAPTION_TO_WEBVTT_CONVERTER_CLASS,
PROP_LIVE_ONTEXTDATA_TO_AMF_CONVERTER_CLASS,
PROP_LIVE_ONTEXTDATA_TO_WEBVTT_CONVERTER_CLASS, PROP_MP4_TO_AMF_CONVERTER_CLASS,
PROP_MP4_TO_CEA608_CONVERTER_CLASS, PROP_MP4_TO_WEBVTT_CONVERTER_CLASS,
PROP_RTMP_VOD_CAPTIONS_DELEGATE_CLASS, PROP_SANJOSE_VOD_CAPTIONS_DELEGATE_CLASS,
PROP_SCC_TO_AMF_CONVERTER_CLASS, PROP_SCC_TO_CEA608_CONVERTER_CLASS,
PROP_SCC_TO_WEBVTT_CONVERTER_CLASS, PROP_SRT_TO_AMF_CONVERTER_CLASS,
PROP_SRT_TO_CEA608_CONVERTER_CLASS, PROP_SRT_TO_WEBVTT_CONVERTER_CLASS,
PROP_TTML_TO_AMF_CONVERTER_CLASS, PROP_TTML_TO_CEA608_CONVERTER_CLASS,
PROP_TTML_TO_WEBVTT_CONVERTER_CLASS, PROP_WEBVTT_CONVERTER_CLASS,
PROP_WEBVTT_TO_AMF_CONVERTER_CLASS, PROP_WEBVTT_TO_CEA608_CONVERTER_CLASS,
PROP_WEBVTT_TO_WEBVTT_CONVERTER_CLASS, PROPERTY_SOURCE_DURATION,
PROPERTY_SOURCE_TIME_SCALE, PROPERTY_SOURCE_TRACK_INDEX, TIMED_TEXT_READER_EXTENSION_SCC,
TIMED_TEXT_READER_EXTENSION_SRT, TIMED_TEXT_READER_EXTENSION_TTML,
TIMED_TEXT_READER_EXTENSION_WEBVTT

```

Fields

TIMED_TEXT_TYPE_CAPTION

```
public static final java.lang.String TIMED_TEXT_TYPE_CAPTION
```

Constant value: **Caption**

SOURCE_TYPE_AMF

```
public static final java.lang.String SOURCE_TYPE_AMF
```

Constant value: **AMF**

(continued from last page)

SOURCE_TYPE_TTML

```
public static final java.lang.String SOURCE_TYPE_TTML
```

Constant value: **TTML**

SOURCE_TYPE_3GPP

```
public static final java.lang.String SOURCE_TYPE_3GPP
```

Constant value: **3GPP**

SOURCE_TYPE_SRT

```
public static final java.lang.String SOURCE_TYPE_SRT
```

Constant value: **SRT**

SOURCE_TYPE_SCC

```
public static final java.lang.String SOURCE_TYPE_SCC
```

Constant value: **SCC**

SOURCE_TYPE_WEBVTT

```
public static final java.lang.String SOURCE_TYPE_WEBVTT
```

Constant value: **WebVTT**

OUTPUT_TYPE_CEA608

```
public static final java.lang.String OUTPUT_TYPE_CEA608
```

Constant value: **CEA608**

OUTPUT_TYPE_AMF

```
public static final java.lang.String OUTPUT_TYPE_AMF
```

Constant value: **AMF**

OUTPUT_TYPE_TTML

```
public static final java.lang.String OUTPUT_TYPE_TTML
```

Constant value: **TTML**

AMF_TEXTTYPE_ON_TEXT_DATA

```
public static final java.lang.String AMF_TEXTTYPE_ON_TEXT_DATA
```


(continued from last page)

Constant value: **onTextData**

LANGUAGE_LOCALE

```
public static final java.lang.String LANGUAGE_LOCALE
```

Constant value: **locale**

LANGUAGE_ID_ENGLISH

```
public static final java.lang.String LANGUAGE_ID_ENGLISH
```

Constant value: **eng**

PROP_VOD_CAPTIONS_ENABLED

```
public static final java.lang.String PROP_VOD_CAPTIONS_ENABLED
```

Timed Text Property "vodCaptionsEnabled": to enabled VOD captions for all supported protocols

Type boolean. The default is true.

Add the property to Application/TimedText/Properties in Application.xml

VOD Captions can be enabled on a per protocol basis using [PROP_CUPERTINO_VOD_CAPTIONS_ENABLED](#), [PROP_RTMP_VOD_CAPTIONS_ENABLED](#), [PROP_SANJOSE_VOD_CAPTIONS_ENABLED](#).

Constant value: **vodCaptionsEnabled**

See Also:

[PROP_CUPERTINO_VOD_CAPTIONS_ENABLED](#)

[PROP_RTMP_VOD_CAPTIONS_ENABLED](#)

[PROP_SANJOSE_VOD_CAPTIONS_ENABLED](#)

PROP_CUPERTINO_VOD_CAPTIONS_ENABLED

```
public static final java.lang.String PROP_CUPERTINO_VOD_CAPTIONS_ENABLED
```

Timed Text Property "cupertinoVODCaptionsEnabled": to enabled VOD captions for Cupertino streaming.

Type boolean. The default is true.

Add the property to Application/TimedText/Properties in Application.xml

VOD Captions can be enabled for all protocols using [PROP_VOD_CAPTIONS_ENABLED](#).

Constant value: **cupertinoVODCaptionsEnabled**

See Also:

[PROP_VOD_CAPTIONS_ENABLED](#)

PROP_RTMP_VOD_CAPTIONS_ENABLED

```
public static final java.lang.String PROP_RTMP_VOD_CAPTIONS_ENABLED
```

(continued from last page)

Timed Text Property "rtmpVODCaptionsEnabled": to enabled VOD captions for RTMP streaming.

Type boolean. The default is true.

Add the property to Application/TimedText/Properties in Application.xml

VOD Captions can be enabled for all protocols using [PROP_VOD_CAPTIONS_ENABLED](#).
Constant value: **rtmpVODCaptionsEnabled**

See Also:

[PROP_VOD_CAPTIONS_ENABLED](#)

PROP_SANJOSE_VOD_CAPTIONS_ENABLED

```
public static final java.lang.String PROP_SANJOSE_VOD_CAPTIONS_ENABLED
```

Timed Text Property "sanJoseVODCaptionsEnabled": to enabled VOD captions for San Jose streaming.

Type boolean. The default is true.

Add the property to Application/TimedText/Properties in Application.xml

VOD Captions can be enabled for all protocols using [PROP_VOD_CAPTIONS_ENABLED](#).
Constant value: **sanJoseVODCaptionsEnabled**

See Also:

[PROP_VOD_CAPTIONS_ENABLED](#)

PROP_CAPTION_LANGUAGES

```
public static final java.lang.String PROP_CAPTION_LANGUAGES
```

Timed Text Property "captionLanguages": used to select presented caption Language.

Type String. The default is [DEFAULT_PROPERTY_CAPION_LANGUAGES](#)

Add the property to Application/TimedText/Properties in Application.xml

VOD Caption languages can be specified on a per protocol basis using [PROP_CUPERTINO_CAPTION_LANGUAGES](#), [PROP_RTMP_CAPTION_LANGUAGES](#), [PROP_SANJOSE_CAPTION_LANGUAGES](#).
Constant value: **captionLanguages**

See Also:

[PROP_CUPERTINO_CAPTION_LANGUAGES](#)

[PROP_RTMP_CAPTION_LANGUAGES](#)

[PROP_SANJOSE_CAPTION_LANGUAGES](#)

DEFAULT_PROPERTY_CAPION_LANGUAGES

```
public static final java.lang.String DEFAULT_PROPERTY_CAPION_LANGUAGES
```

Default value for Timed Text Property "captionLanguages".

Default value is "*", meaning present all found languages.
Constant value: *****

See Also:

[PROP_CAPTION_LANGUAGES](#)

PROP_CUPERTINO_CAPTION_LANGUAGES

```
public static final java.lang.String PROP_CUPERTINO_CAPTION_LANGUAGES
```

(continued from last page)

Timed Text Property "cupertinoCaptionLanguages": to enabled captionLanguage selection for Cupertino streaming.

Type String. The default is [DEFAULT_PROPERTY_CAPION_LANGUAGES](#)

Add the property to Application/TimedText/Properties in Application.xml

VOD Captions can be enabled for all protocols using [PROP_CAPTION_LANGUAGES](#).

Constant value: **cupertinoCaptionLanguages**

See Also:

[PROP_CAPTION_LANGUAGES](#)

PROP_RTMP_CAPTION_LANGUAGES

```
public static final java.lang.String PROP_RTMP_CAPTION_LANGUAGES
```

Timed Text Property "rtmpCaptionLanguages": to enabled captionLanguage selection for RTMP Type String. The default is [DEFAULT_PROPERTY_CAPION_LANGUAGES](#)

Add the property to Application/TimedText/Properties in Application.xml

VOD Captions can be enabled for all protocols using [PROP_CAPTION_LANGUAGES](#).

Constant value: **rtmpCaptionLanguages**

See Also:

[PROP_CAPTION_LANGUAGES](#)

PROP_SANJOSE_CAPTION_LANGUAGES

```
public static final java.lang.String PROP_SANJOSE_CAPTION_LANGUAGES
```

Timed Text Property "sanJoseCaptionLanguages": to enabled captionLanguage selection for San Jose streaming.

Type String. The default is [DEFAULT_PROPERTY_CAPION_LANGUAGES](#)

Add the property to Application/TimedText/Properties in Application.xml

VOD Captions can be enabled for all protocols using [PROP_CAPTION_LANGUAGES](#).

Constant value: **sanJoseCaptionLanguages**

See Also:

[PROP_CAPTION_LANGUAGES](#)

PROP_DEBUG_VOD_CAPTION_LANGUAGE_SELECTION

```
public static final java.lang.String PROP_DEBUG_VOD_CAPTION_LANGUAGE_SELECTION
```

Timed Text Property "debugVODCaptionLanguageSelection": adds info level logging regarding language selection

Type boolean. The default is false.

Add the property to Application/TimedText/Properties in Application.xml

Can be enabled on a per protocol basis using [PROP_DEBUG_CUPERTINO_VOD_CAPTION_LANGUAGE_SELECTION](#), [PROP_DEBUG_RTMP_VOD_CAPTION_LANGUAGE_SELECTION](#), [PROP_DEBUG_SANJOSE_VOD_CAPTION_LANGUAGE_SELECTION](#).

Constant value: **debugVODCaptionLanguageSelection**

PROP_DEBUG_CUPERTINO_VOD_CAPTION_LANGUAGE_SELECTION

```
public static final java.lang.String
PROP_DEBUG_CUPERTINO_VOD_CAPTION_LANGUAGE_SELECTION
```

(continued from last page)

Timed Text Property "cupertinoDebugVODCaptionLanguageSelection": adds info level logging regarding language selection for Cupertino Streaming

Type boolean. The default is false.

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **cupertinoDebugVODCaptionLanguageSelection**

See Also:

[PROP_DEBUG_VOD_CAPTION_LANGUAGE_SELECTION](#)

PROP_DEBUG_RTMP_VOD_CAPTION_LANGUAGE_SELECTION

```
public static final java.lang.String PROP_DEBUG_RTMP_VOD_CAPTION_LANGUAGE_SELECTION
```

Timed Text Property "rtmpDebugVODCaptionLanguageSelection": adds info level logging regarding language selection for RTMP Streaming

Type boolean. The default is false.

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **rtmpDebugVODCaptionLanguageSelection**

See Also:

[PROP_DEBUG_VOD_CAPTION_LANGUAGE_SELECTION](#)

PROP_DEBUG_SANJOSE_VOD_CAPTION_LANGUAGE_SELECTION

```
public static final java.lang.String PROP_DEBUG_SANJOSE_VOD_CAPTION_LANGUAGE_SELECTION
```

Timed Text Property "sanJoseDebugVODCaptionLanguageSelection": adds info level logging regarding language selection for San Jose Streaming

Type boolean. The default is false.

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **sanJoseDebugVODCaptionLanguageSelection**

See Also:

[PROP_DEBUG_VOD_CAPTION_LANGUAGE_SELECTION](#)

PROP_DEBUG_VOD_CAPTION_PROVIDER_DETERMINATION

```
public static final java.lang.String PROP_DEBUG_VOD_CAPTION_PROVIDER_DETERMINATION
```

Timed Text Property "debugVODCaptionProviderDetermination": adds info level logging regarding caption provider determination for a given VOD asset

Type boolean. The default is false.

Add the property to Application/TimedText/Properties in Application.xml

Can be enabled on a per protocol basis using [PROP_DEBUG_CUPERTINO_VOD_CAPTION_PROVIDER_DETERMINATION](#), [PROP_DEBUG_RTMP_VOD_CAPTION_PROVIDER_DETERMINATION](#), [PROP_DEBUG_SANJOSE_VOD_CAPTION_PROVIDER_DETERMINATION](#).

Constant value: **debugVODCaptionProviderDetermination**

(continued from last page)

PROP_DEBUG_CUPERTINO_VOD_CAPTION_PROVIDER_DETERMINATION

```
public static final java.lang.String  
PROP_DEBUG_CUPERTINO_VOD_CAPTION_PROVIDER_DETERMINATION
```

Timed Text Property "cupertinoDebugVODCaptionProviderDetermination": adds info level logging regarding caption provider determination for a given VOD asset for Cupertino Streaming.

Type boolean. The default is false.

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **cupertinoDebugVODCaptionProviderDetermination**

See Also:

[PROP_DEBUG_VOD_CAPTION_PROVIDER_DETERMINATION](#)

PROP_DEBUG_RTMP_VOD_CAPTION_PROVIDER_DETERMINATION

```
public static final java.lang.String  
PROP_DEBUG_RTMP_VOD_CAPTION_PROVIDER_DETERMINATION
```

Timed Text Property "rtmpDebugVODCaptionProviderDetermination": adds info level logging regarding caption provider determination for a given VOD asset for RTMP Streaming.

Type boolean. The default is false.

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **rtmpDebugVODCaptionProviderDetermination**

See Also:

[PROP_DEBUG_VOD_CAPTION_PROVIDER_DETERMINATION](#)

PROP_DEBUG_SANJOSE_VOD_CAPTION_PROVIDER_DETERMINATION

```
public static final java.lang.String  
PROP_DEBUG_SANJOSE_VOD_CAPTION_PROVIDER_DETERMINATION
```

Timed Text Property "sanJoseVODDebugCaptionProviderDetermination": adds info level logging regarding caption provider determination for a given VOD asset for San Jose Streaming.

Type boolean. The default is false.

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **sanJoseVODDebugCaptionProviderDetermination**

See Also:

[PROP_DEBUG_VOD_CAPTION_PROVIDER_DETERMINATION](#)

PROP_DEBUG_VOD_CAPTION_COMPANION_FILES

```
public static final java.lang.String PROP_DEBUG_VOD_CAPTION_COMPANION_FILES
```

(continued from last page)

Timed Text Property "debugVODCaptionFileDetermination": adds info level logging regarding caption companion file determination for a given VOD asset

Type boolean. The default is false.

Add the property to Application/TimedText/Properties in Application.xml

Can be enabled on a per provider basis as well using [PROP_DEBUG_TTML_VOD_CAPTION_COMPANION_FILES](#)
Constant value: **debugVODCaptionFileDetermination**

PROP_UNDEFINED_LANG_CODE

```
public static final java.lang.String PROP_UNDEFINED_LANG_CODE
```

Timed Text Property "captionUndefinedLanguageId": if incoming caption language is undefined, use this language ID.

Type String. If not defined, the Locale's 3 character language code is used. See `Locale.getDefault().getISO3Language()`

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **captionUndefinedLanguageId**

PROP_CAPTION_FILE_NAMING_RULE

```
public static final java.lang.String PROP_CAPTION_FILE_NAMING_RULE
```

Timed Text Property "captionFileNamingRule": to change logic determining the caption companion file naming scheme

Type String. The default value is [DEFAULT_PROPERTY_CAPTION_FILE_NAMING_RULE](#).

This string is used to find the caption file for a given VOD asset. It is used in conjunction with [PROP_CAPTION_PATH_NAMING_RULE](#), which determines the file location.

It is comprised of variables preceded by a dollar sign and enclosed in squiggly brackets. For example: "\${var1}.\${var2}"

Valid variables include these:

- `${SourceStreamName}` : The baseName of the VOD file
- `${SourceBasePath}` : The base path of the VOD asset
- `${FileExtension}` : Registered file extensions with the Timed Text Provider

and these standard ones:

- `${com.wowza.wms.context.VHost}`
- `${com.wowza.wms.context.VHostConfigHome, appInstance.getVHost().getHomePath()};`
- `${com.wowza.wms.context.Application, appInstance.getApplication().getName()};`
- `${com.wowza.wms.context.ApplicationInstance, appInstance.getName()};`

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **captionFileNamingRule**

See Also:

[DEFAULT_PROPERTY_CAPTION_FILE_NAMING_RULE](#)

DEFAULT_PROPERTY_CAPTION_FILE_NAMING_RULE

```
public static final java.lang.String DEFAULT_PROPERTY_CAPTION_FILE_NAMING_RULE
```

Default value for Timed Text Property "captionFileNamingRule".

Default value is `"${SourceStreamName}.${FileExtension}"`

Constant value: **`${SourceStreamName}.${FileExtension}`**

(continued from last page)

See Also:[PROP_CAPTION_FILE_NAMING_RULE](#)

PROP_CAPTION_PATH_NAMING_RULE

```
public static final java.lang.String PROP_CAPTION_PATH_NAMING_RULE
```

Timed Text Property "captionPathNamingRule": to change logic determining the caption companion path

Type String. The default value is [DEFAULT_PROPERTY_CAPTION_PATH_NAMING_RULE](#).

This string is used to find the caption file location for a given VOD asset. It is used in conjunction with [PROP_CAPTION_FILE_NAMING_RULE](#), which determines the file name.

It is comprised of variables preceded by a dollar sign and enclosed in squiggly brackets. For example: "\${var1}.{var2}"

Valid variables include these:

- \${SourceStreamName} : The baseName of the VOD file
- \${SourceBasePath} : The base path of the VOD asset
- \${FileExtension} : Registered file extensions with the Timed Text Provider

and these standard ones:

- \${com.wowza.wms.context.VHost}
- \${com.wowza.wms.context.VHostConfigHome", appInstance.getVHost().getHomePath()};
- \${com.wowza.wms.context.Application", appInstance.getApplication().getName()};
- \${com.wowza.wms.context.ApplicationInstance", appInstance.getName()};

Add the property to Application/TimedText/Properties in Application.xml
Constant value: **captionPathNamingRule**

See Also:[DEFAULT_PROPERTY_CAPTION_PATH_NAMING_RULE](#)

DEFAULT_PROPERTY_CAPTION_PATH_NAMING_RULE

```
public static final java.lang.String DEFAULT_PROPERTY_CAPTION_PATH_NAMING_RULE
```

Default value for Timed Text Property "captionFileNamingRule". Default value is "\${SourceBasePath}"
Constant value: **\${SourceBasePath}**

See Also:[PROP_CAPTION_PATH_NAMING_RULE](#)

PROP_MAXIMUM_CAPTION_DURATION

```
public static final java.lang.String PROP_MAXIMUM_CAPTION_DURATION
```

Timed Text Property "maximumCaptionDuration": used to determine maximum caption duration

Type Integer. In ms. The default is [DEFAULT_MAXIMUM_CAPTION_DURATION](#)

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **maximumCaptionDuration**

(continued from last page)

DEFAULT_MAXIMUM_CAPTION_DURATION

```
public static final int DEFAULT_MAXIMUM_CAPTION_DURATION
```

Default value for Timed Text Property "maximumCaptionDuration".

Default value is 10000.

Constant value: **5000**

See Also:

[PROP_MAXIMUM_CAPTION_DURATION](#)

PROP_CAPTION_CHARACTER_SET

```
public static final java.lang.String PROP_CAPTION_CHARACTER_SET
```

Timed Text Property "captionCharacterSet": used to determine character set when ingesting captions

Type String. The default is [DEFAULT_CAPTION_CHARACTER_SET](#)

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **captionCharacterSet**

DEFAULT_CAPTION_CHARACTER_SET

```
public static final java.lang.String DEFAULT_CAPTION_CHARACTER_SET
```

Default value for Timed Text Property "captionCharacterSet".

Default value is "UTF-8"

Constant value: **UTF-8**

See Also:

[PROP_CAPTION_CHARACTER_SET](#)

PROP_SRT_CAPTION_CHARACTER_SET

```
public static final java.lang.String PROP_SRT_CAPTION_CHARACTER_SET
```

Timed Text Property "srtReaderCaptionCharacterSet": used to determine character set when ingesting SRT captions
Overrides general property "captionCharacterSet"

Type String. The default is [DEFAULT_CAPTION_CHARACTER_SET](#)

Constant value: **srtReaderCaptionCharacterSet**

See Also:

Add the property to Application/TimedText/Properties in Application.xml

PROP_TTML_CAPTION_CHARACTER_SET

```
public static final java.lang.String PROP_TTML_CAPTION_CHARACTER_SET
```

Timed Text Property "ttmlReaderCaptionCharacterSet": used to determine character set when ingesting TTML captions
Overrides general property "captionCharacterSet"

Type String. The default is [DEFAULT_CAPTION_CHARACTER_SET](#)

(continued from last page)

Constant value: **ttmlReaderCaptionCharacterSet**

See Also:

Add the property to Application/TimedText/Properties in Application.xml

PROP_3GPP_CAPTION_CHARACTER_SET

```
public static final java.lang.String PROP_3GPP_CAPTION_CHARACTER_SET
```

Timed Text Property "reader3GPPCaptionCharacterSet": used to determine character set when ingesting TTML captions Overrides general property "captionCharacterSet"

Type String. The default is [DEFAULT_CAPTION_CHARACTER_SET](#)

Constant value: **reader3GPPCaptionCharacterSet**

See Also:

Add the property to Application/TimedText/Properties in Application.xml

PROP_CEA608_CONVERTER_COLOR

```
public static final java.lang.String PROP_CEA608_CONVERTER_COLOR
```

Timed Text Property "cea608CaptionConverterColor": color of text in default CEA caption converter.

Type Integer. Default is [ClosedCaptionCEA608Utils.COL0_WHITE](#)

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **cea608CaptionConverterColor**

See Also:

[ClosedCaptionCEA608Utils.COL0_WHITE](#)

[ClosedCaptionCEA608Utils.COL0_GREEN](#)

[ClosedCaptionCEA608Utils.COL0_BLUE](#)

[ClosedCaptionCEA608Utils.COL0_CYAN](#)

[ClosedCaptionCEA608Utils.COL0_RED](#)

[ClosedCaptionCEA608Utils.COL0_YELLOW](#)

[ClosedCaptionCEA608Utils.COL0_MAGENTA](#)

PROP_CEA608_CONVERTER_CHARS_PER_LINE

```
public static final java.lang.String PROP_CEA608_CONVERTER_CHARS_PER_LINE
```

Timed Text Property "cea608CaptionConverterCharsPerLine": max number characters per line in default CEA caption converter.

Type Integer. Default is [ClosedCaptionCEA608Utils.COUNT_COLS](#)

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **cea608CaptionConverterCharsPerLine**

PROP_CEA608_CONVERTER_CHARACTER_SET

```
public static final java.lang.String PROP_CEA608_CONVERTER_CHARACTER_SET
```

(continued from last page)

Timed Text Property "cea608CaptionConverterCharacterSet": character set in default CEA caption converter.

Type String. Default is "UTF-8"

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **cea608CaptionConverterCharacterSet**

PROP_CUPERTINO_VOD_REMOVE_EXISTING_CEA608

```
public static final java.lang.String PROP_CUPERTINO_VOD_REMOVE_EXISTING_CEA608
```

Timed Text Property "cupertinoVODCaptionsRemoveExistingCEA608": if existing VOD has 608, remove them

Type Boolean. Default is false.

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **cupertinoVODCaptionsRemoveExistingCEA608**

PROP_CUPERTINO_VOD_USE_WEBVTT

```
public static final java.lang.String PROP_CUPERTINO_VOD_USE_WEBVTT
```

Timed Text Property "cupertinoVODCaptionsUseWebVTT"

Type Boolean. Default is false.

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **cupertinoVODCaptionsUseWebVTT**

PROP_CUPERTINO_LIVE_USE_WEBVTT

```
public static final java.lang.String PROP_CUPERTINO_LIVE_USE_WEBVTT
```

Timed Text Property "cupertinoLiveCaptionsUseWebVTT"

Type Boolean. Default is false.

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **cupertinoLiveCaptionsUseWebVTT**

PROP_AMF_CONVERTER_TEXT_TYPE

```
public static final java.lang.String PROP_AMF_CONVERTER_TEXT_TYPE
```

Timed Text Property "amfCaptionConverterTextType": AMF text type used in conversion

Type String. Default is [AMF_TEXTTYPE_ON_TEXT_DATA](#)

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **amfCaptionConverterTextType**

DEFAULT_AMF_CONVERTER_TEXT_TYPE

```
public static final java.lang.String DEFAULT_AMF_CONVERTER_TEXT_TYPE
```

Constant value: **onTextData**

PROP_AMF_CONVERTER_TRACK_INDEX

```
public static final java.lang.String PROP_AMF_CONVERTER_TRACK_INDEX
```

(continued from last page)

Timed Text Property "amfCaptionConverterTrackIndex": track index used for captions

Type Integer.

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **amfCaptionConverterTrackIndex**

DEFAULT_AMF_CONVERTER_TRACK_INDEX

```
public static final int DEFAULT_AMF_CONVERTER_TRACK_INDEX
```

Constant value: **99**

PROP_AMF_CONVERTER_INSERT_ERASES_IN_GAPS

```
public static final java.lang.String PROP_AMF_CONVERTER_INSERT_ERASES_IN_GAPS
```

Timed Text Property "amfCaptionConverterInsertErases": insert erase captions if gaps between captions are greater than PROP_MAXIMUM_CAPTION_DURATION

Type boolean. The default is true.

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **amfCaptionConverterInsertErases**

PROP_WEBVTT_CONVERTER_INSERT_ERASES_IN_GAPS

```
public static final java.lang.String PROP_WEBVTT_CONVERTER_INSERT_ERASES_IN_GAPS
```

Timed Text Property "webVTTCaptionConverterInsertErases": insert erase captions if gaps between captions are greater than PROP_MAXIMUM_CAPTION_DURATION

Type boolean. The default is false.

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **webVTTCaptionConverterInsertErases**

PROP_DEBUG_TTML_VOD_CAPTION_COMPANION_FILES

```
public static final java.lang.String PROP_DEBUG_TTML_VOD_CAPTION_COMPANION_FILES
```

Timed Text Property "debugTTMLVODCaptionFileDetermination": adds info level logging regarding caption companion file determination for a given VOD asset for TTML Streaming.

Type boolean. The default is false.

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **debugTTMLVODCaptionFileDetermination**

See Also:

[PROP_DEBUG_VOD_CAPTION_COMPANION_FILES](#)

PROP_DEBUG_TTML_CAPTION_PARSER

```
public static final java.lang.String PROP_DEBUG_TTML_CAPTION_PARSER
```

(continued from last page)

Timed Text Property "debugTTMLCaptionParser": adds info level logging regarding the TTML caption parser

Type boolean. The default is false.

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **debugTTMLCaptionParser**

PROP_DEBUG_TTML_CAPTION_PARSER_TIME

```
public static final java.lang.String PROP_DEBUG_TTML_CAPTION_PARSER_TIME
```

Timed Text Property "debugTTMLCaptionParserTime": adds info level logging regarding how long it took parser to index

Type boolean. The default is false.

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **debugTTMLCaptionParserTime**

PROP_TTML_INLINE_TAGS_IN_TEXT

```
public static final java.lang.String PROP_TTML_INLINE_TAGS_IN_TEXT
```

Timed Text Property "ttmlInlineTagsInText": TTML parser includes in-line styling and line-break tags in the sample text

Type boolean. The default is false.

Add the property to Application/TimedText/Properties in Application.xml

Example sample: "

I can linebreak

or change fontWeight to bold in the middle of a sample." If ttmlInlineTagsInText=true then the parser provides the sample text as: "I can linebreak

or change fontWeight to bold in the middle of a sample." If ttmlInlineTagsInText=false then the parser provides the sample text as: "I can linebreak or change fontWeight to bold in the middle of a sample."

Constant value: **ttmlInlineTagsInText**

PROP_DEBUG_SRT_VOD_CAPTION_COMPANION_FILES

```
public static final java.lang.String PROP_DEBUG_SRT_VOD_CAPTION_COMPANION_FILES
```

Timed Text Property "debugSRTVODCaptionFileDetermination": adds info level logging regarding caption companion file determination for a given VOD asset for SRT provider

Type boolean. The default is false.

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **debugSRTVODCaptionFileDetermination**

See Also:

[PROP_DEBUG_VOD_CAPTION_COMPANION_FILES](#)

PROP_DEBUG_SRT_CAPTION_PARSER

```
public static final java.lang.String PROP_DEBUG_SRT_CAPTION_PARSER
```

(continued from last page)

Timed Text Property "debugSRTCaptionParser": adds info level logging regarding the SRT caption parser

Type boolean. The default is false.

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **debugSRTCaptionParser**

PROP_DEBUG_SRT_CAPTION_PARSER_TIME

```
public static final java.lang.String PROP_DEBUG_SRT_CAPTION_PARSER_TIME
```

Timed Text Property "debugSRTCaptionParserTime": adds info level logging regarding how long it took parser to index

Type boolean. The default is false.

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **debugSRTCaptionParserTime**

PROP_DEBUG_WEBVTT_VOD_CAPTION_COMPANION_FILES

```
public static final java.lang.String PROP_DEBUG_WEBVTT_VOD_CAPTION_COMPANION_FILES
```

Timed Text Property "debugWebVTTVODCaptionFileDetermination": adds info level logging regarding caption companion file determination for a given VOD asset for WebVTT provider

Type boolean. The default is false.

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **debugWebVTTVODCaptionFileDetermination**

See Also:

[PROP_DEBUG_VOD_CAPTION_COMPANION_FILES](#)

PROP_DEBUG_WEBVTT_CAPTION_PARSER

```
public static final java.lang.String PROP_DEBUG_WEBVTT_CAPTION_PARSER
```

Timed Text Property "debugWebVTTCaptionParser": adds info level logging regarding the WebVTT caption parser

Type boolean. The default is false.

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **debugWebVTTCaptionParser**

PROP_DEBUG_WEBVTT_CAPTION_PARSER_TIME

```
public static final java.lang.String PROP_DEBUG_WEBVTT_CAPTION_PARSER_TIME
```

Timed Text Property "debugWebVTTCaptionParserTime": adds info level logging regarding how long it took parser to index

Type boolean. The default is false.

Add the property to Application/TimedText/Properties in Application.xml

(continued from last page)

Constant value: **debugWebVTTCaptionParserTime**

PROP_DEBUG_SCC_VOD_CAPTION_COMPANION_FILES

```
public static final java.lang.String PROP_DEBUG_SCC_VOD_CAPTION_COMPANION_FILES
```

Timed Text Property "debugSCCVODCaptionFileDetermination": adds info level logging regarding caption companion file determination for a given VOD asset for SCC provider

Type boolean. The default is false.

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **debugSCCVODCaptionFileDetermination**

See Also:

[PROP_DEBUG_VOD_CAPTION_COMPANION_FILES](#)

PROP_DEBUG_SCC_CAPTION_PARSER

```
public static final java.lang.String PROP_DEBUG_SCC_CAPTION_PARSER
```

Timed Text Property "debugSCCCaptionParser": adds info level logging regarding the SCC caption parser

Type boolean. The default is false.

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **debugSCCCaptionParser**

PROP_DEBUG_SCC_CAPTION_PARSER_TIME

```
public static final java.lang.String PROP_DEBUG_SCC_CAPTION_PARSER_TIME
```

Timed Text Property "debugSCCCaptionParserTime": adds info level logging regarding how long it took parser to index

Type boolean. The default is false.

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **debugSCCCaptionParserTime**

PROP_DEBUG_3GPP_CAPTION_PARSER

```
public static final java.lang.String PROP_DEBUG_3GPP_CAPTION_PARSER
```

Timed Text Property "debug3GPPCaptionParser": adds info level logging regarding the 3GPP parser

Type boolean. The default is false.

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **debug3GPPCaptionParser**

PROP_DEBUG_3GPP_CAPTION_PARSER_TIME

```
public static final java.lang.String PROP_DEBUG_3GPP_CAPTION_PARSER_TIME
```

(continued from last page)

Timed Text Property "debug3GPPCaptionParserTime": adds info level logging regarding how long it took parser to index

Type boolean. The default is false.

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **debug3GPPCaptionParserTime**

QUERYSTR_CAPTIONLANGUAGES

```
public static final java.lang.String QUERYSTR_CAPTIONLANGUAGES
```

Query string used for choosing caption languages. Language must be presented using ISO 3 letter language codes.

Constant value: **wowzacaptionlanguages**

PROP_CAPTION_LANGUAGE_QUERY_PARAMETER

```
public static final java.lang.String PROP_CAPTION_LANGUAGE_QUERY_PARAMETER
```

HTTP Streamer Property "captionLanguageQueryParameter": used for changing query string param used for caption language selection. Note: Must go into HTTPStreamers.xml under Properties for each streamer type

Type String. The default is "wowzacaptionlanguages"

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **captionLanguageQueryParameter**

QUERYSTR_CAPTION_FILENAME

```
public static final java.lang.String QUERYSTR_CAPTION_FILENAME
```

Query string used for choosing caption languages. Language must be presented using ISO 3 letter language codes.

Constant value: **wowzacaptionfile**

PROP_CAPTION_FILENAME_QUERY_PARAMETER

```
public static final java.lang.String PROP_CAPTION_FILENAME_QUERY_PARAMETER
```

HTTP Streamer Property "captionFilenameQueryParameter": used for changing query string param used for caption file selection. Note: Must go into HTTPStreamers.xml under Properties for each streamer type

Type String. The default is "wowzacaptionfile"

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **captionFilenameQueryParameter**

PROP_LIVE_CAPTION_INGEST_AMF_ID_STRING

```
public static final java.lang.String PROP_LIVE_CAPTION_INGEST_AMF_ID_STRING
```

Timed Text Property "captionLiveIngestAMFIDString": to change AMF identifier from default "onTextData" to something else

Type String. The default value is "onTextData".

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **captionLiveIngestAMFIDString**

PROP_LIVE_CAPTION_USE_STREAM_NAME_GROUPS

```
public static final java.lang.String PROP_LIVE_CAPTION_USE_STREAM_NAME_GROUPS
```

Timed Text Property "captionLiveIngestUseStreamNameGroups": to enable use of smil files to determine available languages and ingest type

Type Boolean. The default value is true. Only affects streams started with .smil files using StartupStreams or stream manager.

Add the property to Application/TimedText/Properties in Application.xml
Constant value: **captionLiveIngestUseStreamNameGroups**

PROP_LIVE_CAPTION_DEFAULT_LANGUAGES

```
public static final java.lang.String PROP_LIVE_CAPTION_DEFAULT_LANGUAGES
```

Timed Text Property "captionLiveIngestLanguages": comma separated ISO-639 language codes determining what languages are expected in the live stream

Type String. The default value is the language returned by Locale.getDefault().getISO3Language().

Add the property to Application/TimedText/Properties in Application.xml
Constant value: **captionLiveIngestLanguages**

PROP_LIVE_CAPTION_INGEST_TYPE

```
public static final java.lang.String PROP_LIVE_CAPTION_INGEST_TYPE
```

Timed Text Property "captionLiveIngestType": type of live caption ingestion.

This corresponds to LiveIngestTypes in TimedTextProviders.

Type String. The default value is "onTextData". "onCaption" is also a valid value. "CEA608" is a valid value since 4.0.4

Add the property to Application/TimedText/Properties in Application.xml
Constant value: **captionLiveIngestType**

PROP_DEBUG_LIVE_CAPTION_INGEST

```
public static final java.lang.String PROP_DEBUG_LIVE_CAPTION_INGEST
```

Timed Text Property "debugCaptionLiveIngest "

Type Boolean. The default value is false.

Add the property to Application/TimedText/Properties in Application.xml
Constant value: **debugCaptionLiveIngest**

PROP_DEBUG_LIVE_CAPTION_ONTEXTDATA_INGEST

```
public static final java.lang.String PROP_DEBUG_LIVE_CAPTION_ONTEXTDATA_INGEST
```

Timed Text Property "debugCaptionLiveOnTextDataIngest "

Type Boolean. The default value is false. Overrides [PROP_LIVE_CAPTION_DEFAULT_LANGUAGES](#)

Add the property to Application/TimedText/Properties in Application.xml
Constant value: **debugCaptionLiveOnTextDataIngest**

PROP_DEBUG_LIVE_CAPTION_ONCAPTION_INGEST

```
public static final java.lang.String PROP_DEBUG_LIVE_CAPTION_ONCAPTION_INGEST
```

(continued from last page)

Timed Text Property "debugCaptionLiveOnCaptionIngest"

Type Boolean. The default value is false. Overrides [PROP_LIVE_CAPTION_DEFAULT_LANGUAGES](#)

Add the property to Application/TimedText/Properties in Application.xml
Constant value: **debugCaptionLiveOnCaptionIngest**

PROP_LIVE_CAPTION_INGEST_ON_CAPTION_STRING

```
public static final java.lang.String PROP_LIVE_CAPTION_INGEST_ON_CAPTION_STRING
```

Timed Text Property "captionLiveIngestOnCaptionString": to change AMF identifier from default "onCaption" to something else

Type String. The default value is "onCaption".

Add the property to Application/TimedText/Properties in Application.xml
Constant value: **captionLiveIngestOnCaptionString**

PROP_LIVE_CAPTION_INGEST_ON_CAPTION_INFO_STRING

```
public static final java.lang.String PROP_LIVE_CAPTION_INGEST_ON_CAPTION_INFO_STRING
```

Timed Text Property "captionLiveIngestOnCaptionInfoString": to change AMF identifier from default "onCaptionInfo" to something else

Type String. The default value is "onCaptionInfo".

Add the property to Application/TimedText/Properties in Application.xml
Constant value: **captionLiveIngestOnCaptionInfoString**

PROP_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608_IDLE_TIME_TO_FLUSH_DECODER

```
public static final java.lang.String  
PROP_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608_IDLE_TIME_TO_FLUSH_DECODER
```

Timed Text Property "captionLiveIngestOnCaptionCEA608IdleTimeToFlushDecoder" Controls allowed amount of idle time when parsing CEA-608 captions when ingested from AMF 708 data events

Type Integer. The default value is 250 ms.

Add the property to Application/TimedText/Properties in Application.xml
Constant value: **captionLiveIngestOnCaptionCEA608IdleTimeToFlushDecoder**

PROP_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608_ENABLE_FIELD1

```
public static final java.lang.String  
PROP_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608_ENABLE_FIELD1
```

Timed Text Property "captionLiveIngestOnCaptionCEA608EnableField2": When ingesting AMF onCaption 708 data events (actually CEA-608), controls whether CEA-608 field 1 is enabled for decoding

Type Boolean. Default is true;

Add the property to Application/TimedText/Properties in Application.xml
Constant value: **captionLiveIngestOnCaptionCEA608EnableField1**

PROP_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608_ENABLE_FIELD2

```
public static final java.lang.String  
PROP_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608_ENABLE_FIELD2
```

(continued from last page)

Timed Text Property "captionLiveIngestOnCaptionCEA608EnableField2": When ingesting AMF onCaption 708 data events (actually CEA-608), controls whether CEA-608 field 2 is enabled for decoding

Type Boolean. Default is false;

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **captionLiveIngestOnCaptionCEA608EnableField2**

PROP_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608_MAX_WARN_MESSAGES

```
public static final java.lang.String
```

```
PROP_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608_MAX_WARN_MESSAGES
```

Timed Text Property "captionLiveIngestOnCaptionCEA608MaxWarnMessages": When ingesting AMF onCaption 708 data events (actually CEA-608), controls the maximum number of warning messages that are logged by the decoder

Type Integer. Default is 25.

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **captionLiveIngestOnCaptionCEA608MaxWarnMessages**

PROP_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608_TRACK_NUMBER

```
public static final java.lang.String
```

```
PROP_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608_TRACK_NUMBER
```

Timed Text Property "captionLiveIngestOnCaptionCEA608TrackNumber": When ingesting AMF onCaption 708 data events (actually CEA-608), controls the track number used to add captions as onTextData

Type Integer. Default is 99.

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **captionLiveIngestOnCaptionCEA608TrackNumber**

PROP_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608_LANGUAGE_ID

```
public static final java.lang.String
```

```
PROP_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608_LANGUAGE_ID
```

Timed Text Property "captionLiveIngestOnCaptionCEA608LanguageID": When ingesting AMF onCaption 708 data events (actually CEA-608), controls the language ID used to add captions as onTextData

Type String. Default is language code of Java's locale.

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **captionLiveIngestOnCaptionCEA608LanguageID**

PROP_DEBUG_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608_VERBOSE

```
public static final java.lang.String
```

```
PROP_DEBUG_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608_VERBOSE
```

Timed Text Property "debugCaptionLiveIngestOnCaptionCEA608Verbose" : Adds verbose logging to AMF onCaption events of type "708" (CEA-608)

Type Boolean. The default value is false.

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **debugCaptionLiveIngestOnCaptionCEA608Verbose**

PROP_DEBUG_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608_LOG_AS_SCC

```
public static final java.lang.String
```

```
PROP_DEBUG_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608_LOG_AS_SCC
```

(continued from last page)

Timed Text Property "debugCaptionLiveIngestOnCaptionCEA608LogAsSCC" : Adds additional logging of decoding AMF onCaption events of type "708" (CEA-608) such that they are logged in SCC format

Type Boolean. The default value is false.

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **debugCaptionLiveIngestOnCaptionCEA608LogAsSCC**

PROP_DEBUG_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608_LOG_AS_RAW

```
public static final java.lang.String
```

```
PROP_DEBUG_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608_LOG_AS_RAW
```

Timed Text Property "debugCaptionLiveIngestOnCaptionCEA608LogAsRaw" : Adds additional logging of decoding AMF onCaption events of type "708" (CEA-608) such that they are logged in a raw binary state

Type Boolean. The default value is false.

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **debugCaptionLiveIngestOnCaptionCEA608LogAsRaw**

ON_TEXT_DATA

```
public static final java.lang.String ON_TEXT_DATA
```

Constant value: **onTextData**

ON_CAPTION_INFO

```
public static final java.lang.String ON_CAPTION_INFO
```

Constant value: **onCaptionInfo**

ON_CAPTION

```
public static final java.lang.String ON_CAPTION
```

Constant value: **onCaption**

CEA608

```
public static final java.lang.String CEA608
```

Constant value: **CEA608**

SMIL_PARAM_WOWZA_CAPTION_INGEST_TYPE

```
public static final java.lang.String SMIL_PARAM_WOWZA_CAPTION_INGEST_TYPE
```

Constant value: **wowzacaptioningesttype**

SMIL_PARAM_IS_WOWZA_CAPTION_STREAM

```
public static final java.lang.String SMIL_PARAM_IS_WOWZA_CAPTION_STREAM
```

(continued from last page)

Constant value: **iswowzacaptionstream**

Package

com.wowza.wms.transcoder.model

com.wowza.wms.transcoder.model Interface ILiveStreamTranscoderActionNotify

public interface **ILiveStreamTranscoderActionNotify**
extends

ILiveStreamTranscoderActionNotify: listener interface for live stream transcoder events.

Method Summary

void	onCalculateSourceAudioBitrate (LiveStreamTranscoder liveStreamTranscoder, long bitrate) Called when the bitrate of the source audio stream is calculated
void	onCalculateSourceVideoBitrate (LiveStreamTranscoder liveStreamTranscoder, long bitrate) Called when the bitrate of the source video stream is calculated
void	onInitAfterLoadTemplate (LiveStreamTranscoder liveStreamTranscoder) Called just after transcoder template is loaded.
void	onInitBeforeLoadTemplate (LiveStreamTranscoder liveStreamTranscoder) Called just before transcoder template is loaded
void	onInitStart (LiveStreamTranscoder liveStreamTranscoder, String streamName, String transcoderName, IApplicationInstance appInstance, LiveStreamTranscoderItem liveStreamTranscoderItem) Call when live stream transcoder is first created
void	onInitStop (LiveStreamTranscoder liveStreamTranscoder) At the end of the initialization process
void	onRegisterStreamNameGroup (LiveStreamTranscoder liveStreamTranscoder, TranscoderStreamNameGroup streamNameGroup) Called after a stream name group is resolved and registered with MediaStreamMap
void	onResetStream (LiveStreamTranscoder liveStreamTranscoder) Called when the stream feeding the live stream transcoder switches.
void	onSessionAudioDecodeCodecInfo (LiveStreamTranscoder liveStreamTranscoder, com.wowza.wms.media.model.MediaCodecInfoAudio codecInfoAudio) Called when audio decoding information is available.
void	onSessionAudioEncodeCodecInfo (LiveStreamTranscoder liveStreamTranscoder, TranscoderSessionAudioEncode sessionAudioEncode, com.wowza.wms.media.model.MediaCodecInfoAudio codecInfoAudio) Called when audio encoding information is available.
void	onSessionAudioEncodeCreate (LiveStreamTranscoder liveStreamTranscoder, TranscoderSessionAudioEncode sessionAudioEncode) Called when audio encoder session is created.
void	onSessionAudioEncodeInit (LiveStreamTranscoder liveStreamTranscoder, TranscoderSessionAudioEncode sessionAudioEncode) Called after audio session is initialized.

void	<u>onSessionAudioEncodeSetup</u> (LiveStreamTranscoder liveStreamTranscoder, TranscoderSessionAudioEncode sessionAudioEncode) Called after native audio encoder is created and initialized.
void	<u>onSessionDataEncodeCreate</u> (LiveStreamTranscoder liveStreamTranscoder, TranscoderSessionDataEncode sessionDataEncode) Called when data encoder session is created.
void	<u>onSessionDataEncodeInit</u> (LiveStreamTranscoder liveStreamTranscoder, TranscoderSessionDataEncode sessionDataEncode) Called after data session is initialized.
void	<u>onSessionDestinationCreate</u> (LiveStreamTranscoder liveStreamTranscoder, TranscoderSessionDestination sessionDestination) Called when transcoding destination is created
void	<u>onSessionVideoDecodeCodecInfo</u> (LiveStreamTranscoder liveStreamTranscoder, com.wowza.wms.media.model.MediaCodecInfoVideo codecInfoVideo) Called when video decoding information is available.
void	<u>onSessionVideoEncodeCodecInfo</u> (LiveStreamTranscoder liveStreamTranscoder, TranscoderSessionVideoEncode sessionVideoEncode, com.wowza.wms.media.model.MediaCodecInfoVideo codecInfoVideo) Called when video encoding information is available.
void	<u>onSessionVideoEncodeCreate</u> (LiveStreamTranscoder liveStreamTranscoder, TranscoderSessionVideoEncode sessionVideoEncode) Called when video encoder session is created.
void	<u>onSessionVideoEncodeInit</u> (LiveStreamTranscoder liveStreamTranscoder, TranscoderSessionVideoEncode sessionVideoEncode) Called after video session is initialized.
void	<u>onSessionVideoEncodeSetup</u> (LiveStreamTranscoder liveStreamTranscoder, TranscoderSessionVideoEncode sessionVideoEncode) Called after native video encoder is created and initialized.
void	<u>onShutdownStart</u> (LiveStreamTranscoder liveStreamTranscoder) Called when the live stream transcoder starts to shutdown.
void	<u>onShutdownStop</u> (LiveStreamTranscoder liveStreamTranscoder) Called when the live stream transcoder is shutdown.
void	<u>onUnregisterStreamNameGroup</u> (LiveStreamTranscoder liveStreamTranscoder, TranscoderStreamNameGroup streamNameGroup) Called after a stream name group is unregistered with MediaStreamMap

Methods

onInitStart

```
public void onInitStart(LiveStreamTranscoder liveStreamTranscoder,
    String streamName,
    String transcoderName,
    IApplicationInstance appInstance,
    LiveStreamTranscoderItem liveStreamTranscoderItem)
```

Call when live stream transcoder is first created

Parameters:

(continued from last page)

liveStreamTranscoder - live stream transcoder
streamName - stream name
transcoderName - transcoder name
appInstance - application instance
liveStreamTranscoderItem - live stream transcoder definition

onInitBeforeLoadTemplate

```
public void onInitBeforeLoadTemplate(LiveStreamTranscoder liveStreamTranscoder)
```

Called just before transcoder template is loaded

Parameters:

liveStreamTranscoder - live stream transcoder

onInitAfterLoadTemplate

```
public void onInitAfterLoadTemplate(LiveStreamTranscoder liveStreamTranscoder)
```

Called just after transcoder template is loaded. Good place to modify values loaded from template.

Parameters:

liveStreamTranscoder - live stream transcoder

onInitStop

```
public void onInitStop(LiveStreamTranscoder liveStreamTranscoder)
```

At the end of the initialization process

Parameters:

liveStreamTranscoder - live stream transcoder

onCalculateSourceVideoBitrate

```
public void onCalculateSourceVideoBitrate(LiveStreamTranscoder liveStreamTranscoder,  
long bitrate)
```

Called when the bitrate of the source video stream is calculated

Parameters:

liveStreamTranscoder - live stream transcoder
bitrate - source bitrate (bytes per second)

onCalculateSourceAudioBitrate

```
public void onCalculateSourceAudioBitrate(LiveStreamTranscoder liveStreamTranscoder,  
long bitrate)
```

Called when the bitrate of the source audio stream is calculated

Parameters:

liveStreamTranscoder - live stream transcoder
bitrate - source bitrate (bytes per second)

(continued from last page)

onSessionDestinationCreate

```
public void onSessionDestinationCreate(LiveStreamTranscoder liveStreamTranscoder,  
    TranscoderSessionDestination sessionDestination)
```

Called when transcoding destination is created

Parameters:

liveStreamTranscoder - live stream transcoder
sessionDestination - destination

onSessionVideoEncodeCreate

```
public void onSessionVideoEncodeCreate(LiveStreamTranscoder liveStreamTranscoder,  
    TranscoderSessionVideoEncode sessionVideoEncode)
```

Called when video encoder session is created.

Parameters:

liveStreamTranscoder - live stream transcoder
sessionVideoEncode - video session

onSessionAudioEncodeCreate

```
public void onSessionAudioEncodeCreate(LiveStreamTranscoder liveStreamTranscoder,  
    TranscoderSessionAudioEncode sessionAudioEncode)
```

Called when audio encoder session is created.

Parameters:

liveStreamTranscoder - live stream transcoder
sessionAudioEncode - audio session

onSessionDataEncodeCreate

```
public void onSessionDataEncodeCreate(LiveStreamTranscoder liveStreamTranscoder,  
    TranscoderSessionDataEncode sessionDataEncode)
```

Called when data encoder session is created.

Parameters:

liveStreamTranscoder - live stream transcoder
sessionDataEncode - data session

onSessionVideoEncodeInit

```
public void onSessionVideoEncodeInit(LiveStreamTranscoder liveStreamTranscoder,  
    TranscoderSessionVideoEncode sessionVideoEncode)
```

Called after video session is initialized.

Parameters:

liveStreamTranscoder - live stream transcoder
sessionVideoEncode - video session

onSessionAudioEncodeInit

```
public void onSessionAudioEncodeInit(LiveStreamTranscoder liveStreamTranscoder,  
    TranscoderSessionAudioEncode sessionAudioEncode)
```

(continued from last page)

Called after audio session is initialized.

Parameters:

liveStreamTranscoder - live stream transcoder
sessionAudioEncode - audio session

onSessionDataEncodeInit

```
public void onSessionDataEncodeInit(LiveStreamTranscoder liveStreamTranscoder,  
    TranscoderSessionDataEncode sessionDataEncode)
```

Called after data session is initialized.

Parameters:

liveStreamTranscoder - live stream transcoder
sessionDataEncode - data session

onSessionVideoEncodeSetup

```
public void onSessionVideoEncodeSetup(LiveStreamTranscoder liveStreamTranscoder,  
    TranscoderSessionVideoEncode sessionVideoEncode)
```

Called after native video encoder is created and initialized.

Parameters:

liveStreamTranscoder - live stream transcoder
sessionVideoEncode - video session

onSessionAudioEncodeSetup

```
public void onSessionAudioEncodeSetup(LiveStreamTranscoder liveStreamTranscoder,  
    TranscoderSessionAudioEncode sessionAudioEncode)
```

Called after native audio encoder is created and initialized.

Parameters:

liveStreamTranscoder - live stream transcoder
sessionAudioEncode - audio session

onSessionVideoEncodeCodecInfo

```
public void onSessionVideoEncodeCodecInfo(LiveStreamTranscoder liveStreamTranscoder,  
    TranscoderSessionVideoEncode sessionVideoEncode,  
    com.wowza.wms.media.model.MediaCodecInfoVideo codecInfoVideo)
```

Called when video encoding information is available.

Parameters:

liveStreamTranscoder - live stream transcoder
sessionVideoEncode - video session
codecInfoVideo - encoding info

onSessionAudioEncodeCodecInfo

```
public void onSessionAudioEncodeCodecInfo(LiveStreamTranscoder liveStreamTranscoder,  
    TranscoderSessionAudioEncode sessionAudioEncode,  
    com.wowza.wms.media.model.MediaCodecInfoAudio codecInfoAudio)
```

Called when audio encoding information is available.

(continued from last page)

Parameters:

liveStreamTranscoder - live stream transcoder
sessionAudioEncode - audio session
codecInfoAudio - encoding info

onSessionVideoDecodeCodecInfo

```
public void onSessionVideoDecodeCodecInfo(LiveStreamTranscoder liveStreamTranscoder,  
com.wowza.wms.media.model.MediaCodecInfoVideo codecInfoVideo)
```

Called when video decoding information is available.

Parameters:

liveStreamTranscoder - live stream transcoder
codecInfoVideo - video info

onSessionAudioDecodeCodecInfo

```
public void onSessionAudioDecodeCodecInfo(LiveStreamTranscoder liveStreamTranscoder,  
com.wowza.wms.media.model.MediaCodecInfoAudio codecInfoAudio)
```

Called when audio decoding information is available.

Parameters:

liveStreamTranscoder - live stream transcoder
codecInfoAudio - audio info

onRegisterStreamNameGroup

```
public void onRegisterStreamNameGroup(LiveStreamTranscoder liveStreamTranscoder,  
TranscoderStreamNameGroup streamNameGroup)
```

Called after a stream name group is resolved and registered with MediaStreamMap

Parameters:

liveStreamTranscoder - live stream transcoder
streamNameGroup - stream name group

onUnregisterStreamNameGroup

```
public void onUnregisterStreamNameGroup(LiveStreamTranscoder liveStreamTranscoder,  
TranscoderStreamNameGroup streamNameGroup)
```

Called after a stream name group is unregistered with MediaStreamMap

Parameters:

liveStreamTranscoder - live stream transcoder
streamNameGroup - stream name group

onShutdownStart

```
public void onShutdownStart(LiveStreamTranscoder liveStreamTranscoder)
```

Called when the live stream transcoder starts to shutdown.

Parameters:

liveStreamTranscoder - live stream transcoder

onShutdownStop

```
public void onShutdownStop(LiveStreamTranscoder liveStreamTranscoder)
```

Called when the live stream transcoder is shutdown.

Parameters:

liveStreamTranscoder - live stream transcoder

onResetStream

```
public void onResetStream(LiveStreamTranscoder liveStreamTranscoder)
```

Called when the stream feeding the live stream transcoder switches.

Parameters:

liveStreamTranscoder - live stream transcoder

com.wowza.wms.transcoder.model Interface ITranscoderFrameGrabProvider

public interface **ITranscoderFrameGrabProvider**
extends

ITranscoderFrameGrabProvider: interface used to grab frames from the live stream transcoder.

Method Summary

java.util.List	getAndClearPendingFrameGrabs() Returns a list of all grab frame requests that are pending and clears the pending list.
void	grabFrame(ITranscoderFrameGrabResult grabResult) Call to grab a frame.
void	grabFrame(ITranscoderFrameGrabResult grabResult, int width, int height) Call to grab a frame.

Methods

grabFrame

public void **grabFrame**([ITranscoderFrameGrabResult](#) grabResult)

Call to grab a frame. The size of the frame will be the full size of the video frame.

Parameters:

grabResult - the class that will be called back when frame is available

grabFrame

public void **grabFrame**([ITranscoderFrameGrabResult](#) grabResult,
int width,
int height)

Call to grab a frame. The size of the frame is controlled by width and height.

Parameters:

grabResult - the class that will be called back when frame is available

width - frame width

height - frame height

getAndClearPendingFrameGrabs

public java.util.List **getAndClearPendingFrameGrabs**()

Returns a list of all grab frame requests that are pending and clears the pending list.

Returns:

list of pending grab frames

com.wowza.wms.transcoder.model

Interface ITranscoderFrameGrabResult

public interface **ITranscoderFrameGrabResult**
extends

ITranscoderFrameGrabResult: Implement this interface when using frame grabber interface to grab transcoder video frames.

Method Summary	
void	onGrabFrame (TranscoderNativeVideoFrame nativeFrame) Trigger when frame is available.

Methods

onGrabFrame

public void **onGrabFrame**(TranscoderNativeVideoFrame nativeFrame)

Trigger when frame is available.

Parameters:

nativeFrame - native video frame.

com.wowza.wms.transcoder.model
Interface ITranscoderOverlayImages

public interface ITranscoderOverlayImages
extends

Method Summary	
void	addOverlayImage (int index, TranscoderVideoOverlayFrame overlay)
java.util.Map	getOverlayImages ()

Methods

addOverlayImage

```
public void addOverlayImage(int index,  
    TranscoderVideoOverlayFrame overlay)
```

getOverlayImages

```
public java.util.Map getOverlayImages()
```

com.wowza.wms.transcoder.model Interface ITranscoderOverlayProvider

public interface **ITranscoderOverlayProvider**
extends

ITranscoderOverlayProvider: interface to add and remove video overlays.

Method Summary

void	<code>addOverlay</code> (int index, TranscoderVideoOverlayFrame overlay) Add an overlay to a video stream.
void	<code>clearOverlay</code> (int index) Clear video overlay
java.util.Map	<code>getAndClearPendingOverlays</code> () Get a list of pending overlay requests and clear the pending overlay queue.
boolean	<code>isOverlayAvailable</code> () Returns true if there is a pending overlay request.

Methods

addOverlay

```
public void addOverlay(int index,  
    TranscoderVideoOverlayFrame overlay)
```

Add an overlay to a video stream.

Parameters:

index - over lay index (zero is bottom in z-order)
overlay - overlay object

clearOverlay

```
public void clearOverlay(int index)
```

Clear video overlay

Parameters:

index - over lay index (zero is bottom in z-order)

isOverlayAvailable

```
public boolean isOverlayAvailable()
```

Returns true if there is a pending overlay request.

Returns:

true if there is a pending overlay request

getAndClearPendingOverlays

```
public java.util.Map getAndClearPendingOverlays()
```

Get a list of pending overlay requests and clear the pending overlay queue.

Returns:

list of pending overlay requests

com.wowza.wms.transcoder.model Interface ITranscoderVideoDecoderNotify

public interface **ITranscoderVideoDecoderNotify**
extends

Method Summary

void	<code>onAfterDecodeFrame</code> (TranscoderSessionVideo sessionVideo, TranscoderStreamSourceVideo sourceVideo, long frameCount)
void	<code>onAfterScaleFrame</code> (TranscoderSessionVideo sessionVideo, TranscoderStreamSourceVideo sourceVideo, long frameCount)
void	<code>onBeforeDecodeFrame</code> (TranscoderSessionVideo sessionVideo, TranscoderStreamSourceVideo sourceVideo, long frameCount)
void	<code>onBeforeScaleFrame</code> (TranscoderSessionVideo sessionVideo, TranscoderStreamSourceVideo sourceVideo, long frameCount)

Methods

onBeforeDecodeFrame

```
public void onBeforeDecodeFrame(TranscoderSessionVideo sessionVideo,  
    TranscoderStreamSourceVideo sourceVideo,  
    long frameCount)
```

onAfterDecodeFrame

```
public void onAfterDecodeFrame(TranscoderSessionVideo sessionVideo,  
    TranscoderStreamSourceVideo sourceVideo,  
    long frameCount)
```

onBeforeScaleFrame

```
public void onBeforeScaleFrame(TranscoderSessionVideo sessionVideo,  
    TranscoderStreamSourceVideo sourceVideo,  
    long frameCount)
```

onAfterScaleFrame

```
public void onAfterScaleFrame(TranscoderSessionVideo sessionVideo,  
    TranscoderStreamSourceVideo sourceVideo,  
    long frameCount)
```

(continued from last page)

com.wowza.wms.transcoder.model Interface ITranscoderVideoEncoderNotify

public interface **ITranscoderVideoEncoderNotify**
extends

Method Summary

void	<u>onAfterEncodeFrame</u> (TranscoderSessionVideoEncode sessionVideoEncode, TranscoderStreamDestinationVideo destinationVideo, long frameCount)
void	<u>onBeforeEncodeFrame</u> (TranscoderSessionVideoEncode sessionVideoEncode, TranscoderStreamDestinationVideo destinationVideo, long frameCount)

Methods

onBeforeEncodeFrame

```
public void onBeforeEncodeFrame(TranscoderSessionVideoEncode sessionVideoEncode,  
    TranscoderStreamDestinationVideo destinationVideo,  
    long frameCount)
```

onAfterEncodeFrame

```
public void onAfterEncodeFrame(TranscoderSessionVideoEncode sessionVideoEncode,  
    TranscoderStreamDestinationVideo destinationVideo,  
    long frameCount)
```

com.wowza.wms.transcoder.model Interface ITranscoderWorker

public interface **ITranscoderWorker**
extends

For internal use.

Method Summary

void	<code>destroy()</code>
String	<code>getContextStr()</code>
long	<code>getFrameCount()</code>
boolean	<code>isEncoderInUse()</code>
boolean	<code>isRunning()</code> Returns true if transcoder worker is running (internal use)
void	<code>setFrameCount(long frameCount)</code>

Methods

isRunning

public boolean **isRunning()**

Returns true if transcoder worker is running (internal use)

Returns:

true if transcoder worker is running (internal use)

getFrameCount

public long **getFrameCount()**

setFrameCount

public void **setFrameCount(long frameCount)**

isEncoderInUse

public boolean **isEncoderInUse()**

(continued from last page)

getContextStr

```
public String getContextStr()
```

destroy

```
public void destroy()
```

com.wowza.wms.transcoder.model

Interface ITranscoderWorkerSorterSender

public interface ITranscoderWorkerSorterSender
extends

For internal use.

Method Summary	
void	addSorterPacket (TranscoderPacketSorterHolder sorterHolder) For internal use.

Methods

addSorterPacket

public void addSorterPacket(TranscoderPacketSorterHolder sorterHolder)

For internal use.

Parameters:
sorterHolder - sorter holder

Package

com.wowza.wms.util

com.wowza.wms.util Class RTPUtils

java.lang.Object

└─com.wowza.wms.util.RTPUtils

public class **RTPUtils**
extends Object

Constructor Summary

public	RTPUtils()
--------	----------------------------

Method Summary

static double[]	decodeRangeHeader (String rangeStr) Decode RTP range header, Internal use.
static RTPStream	decodeStreamInfo (IApplicationInstance appInstance, RTPContext context, String streamId, String streamInfo) Decode SDP info and create RTP stream
static RTPStream	decodeStreamInfo (IApplicationInstance appInstance, RTPStreamContext rtpStreamContext, RTPContext context, String streamId, String streamInfo) Decode SDP info and create RTP stream
static RTPStream	decodeStreamInfo (RTPContext context, String streamId, String streamInfo) Decode SDP info and create RTP stream
static byte[]	formatH264CodecConfig (byte[] sps, java.util.List ppss, byte[] profileLevel) Format codec config info, Internal use.
static byte[]	formatH264CodecConfigPacket (byte[] sps, java.util.List ppss, byte[] profileLevel) Format codec config info, Internal use.
static boolean	loadConfigFile (RTPContext rtpContext, String fileURL) Load config file, Internal use.
static RTTPushPublishSession	startRTTPull (IApplicationInstance appInstance, String streamName, boolean streamPacketizer, String ipAddress, int streamPort) Start pushing an RTP stream
static RTTPushPublishSession	startRTTPull (IApplicationInstance appInstance, String streamName, boolean streamPacketizer, String ipAddress, int streamPort, boolean isRTPWrapped) Start pushing an RTP stream
static RTTPushPublishSession	startRTTPull (IApplicationInstance appInstance, String streamName, boolean streamPacketizer, String ipAddress, int audioPort, int videoPort) Start pushing an RTP stream

static RTPPushPublishSession	startRTPPull (IApplicationInstance appInstance, String streamName, boolean streamPacketizer, String ipAddress, int audioPort, int videoPort, boolean isRTPWrapped) Start pushing an RTP stream
static RTPPushPublishSession	startRTPPull (IApplicationInstance appInstance, String streamName, RTPDestination rtpDestination) Start pushing an RTP stream
static void	stopRTPPull (RTPPushPublishSession rtpPushPublishSession) Stop pushing an RTP stream
static String	updateSDPDestination (RTPDestination rtpDestination, String sdpData) Update SDP data information with RTP destination information
static void	writeCodecConfig (RTPTrack rtpTrack, int codecId, long adjTimecode, byte[] codecConfig)
static void	writeCodecConfig (RTPTrack rtpTrack, long adjTimecode, byte[] codecConfig) Write codec config information, Internal use.
static void	writeCodecConfig (RTPTrack rtpTrack, long adjTimecode, byte[] codecConfig, int codec)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

RTPUtils

```
public RTPUtils()
```

Methods

updateSDPDestination

```
public static String updateSDPDestination(RTPDestination rtpDestination, String sdpData)
```

Update SDP data information with RTP destination information

Parameters:

rtpDestination - RTP destination
sdpData - SDP data

Returns:

modified SDP data

(continued from last page)

decodeRangeHeader

```
public static double[] decodeRangeHeader(String rangeStr)
```

Decode RTP range header, Internal use.

Parameters:

rangeStr

Returns:

range values

stopRTPPull

```
public static void stopRTPPull(RTPPushPublishSession rtpPushPublishSession)
```

Stop pushing an RTP stream

Parameters:

rtpPushPublishSession - RTP push session

startRTPPull

```
public static RTPPushPublishSession startRTPPull(IApplicationInstance appInstance,  
String streamName,  
boolean streamPacketizer,  
String ipAddress,  
int streamPort,  
boolean isRTPWrapped)
```

Start pushing an RTP stream

Parameters:

appInstance - application instance
streamName - stream name
streamPacketizer - stream packetizer
ipAddress - IP address
streamPort - stream port
isRTPWrapped - is RTP wrapped

Returns:

RTP push session

startRTPPull

```
public static RTPPushPublishSession startRTPPull(IApplicationInstance appInstance,  
String streamName,  
boolean streamPacketizer,  
String ipAddress,  
int streamPort)
```

Start pushing an RTP stream

Parameters:

appInstance - application instance
streamName - stream name
streamPacketizer - stream packetizer
ipAddress - IP address
streamPort - stream port

(continued from last page)

Returns:

RTP push session

startRTPPull

```
public static RTPPushPublishSession startRTPPull(IApplicationInstance appInstance,  
    String streamName,  
    boolean streamPacketizer,  
    String ipAddress,  
    int audioPort,  
    int videoPort)
```

Start pushing an RTP stream

Parameters:

appInstance - application instance
streamName - stream name
streamPacketizer - stream packetizer
ipAddress - IP address
audioPort - audio port
videoPort - video port

Returns:

RTP push session

startRTPPull

```
public static RTPPushPublishSession startRTPPull(IApplicationInstance appInstance,  
    String streamName,  
    boolean streamPacketizer,  
    String ipAddress,  
    int audioPort,  
    int videoPort,  
    boolean isRTPWrapped)
```

Start pushing an RTP stream

Parameters:

appInstance - application instance
streamName - stream name
streamPacketizer - stream packetizer
ipAddress - IP address
audioPort - audio port
videoPort - video port
isRTPWrapped - is RTP wrapped

Returns:

RTP push session

startRTPPull

```
public static RTPPushPublishSession startRTPPull(IApplicationInstance appInstance,  
    String streamName,  
    RTPDestination rtpDestination)
```

Start pushing an RTP stream

Parameters:

appInstance - application instance
streamName - stream name
rtpDestination - RTP destination

(continued from last page)

Returns:

RTP push session

writeCodecConfig

```
public static void writeCodecConfig(RTPTrack rtpTrack,
    int codecId,
    long adjTimecode,
    byte[] codecConfig)
```

writeCodecConfig

```
public static void writeCodecConfig(RTPTrack rtpTrack,
    long adjTimecode,
    byte[] codecConfig)
```

Write codec config information, Internal use.

Parameters:

rtpTrack - RTP track
adjTimecode - timecode (milliseconds)
codecConfig - codec config

writeCodecConfig

```
public static void writeCodecConfig(RTPTrack rtpTrack,
    long adjTimecode,
    byte[] codecConfig,
    int codec)
```

loadConfigFile

```
public static boolean loadConfigFile(RTPContext rtpContext,
    String fileURL)
```

Load config file, Internal use.

Parameters:

rtpContext
fileURL

decodeStreamInfo

```
public static RTPStream decodeStreamInfo(RTPContext context,
    String streamId,
    String streamInfo)
```

Decode SDP info and create RTP stream

Parameters:

context - RTP context
streamId - stream id
streamInfo - SDP data

Returns:

(continued from last page)

RTP stream

decodeStreamInfo

```
public static RTPStream decodeStreamInfo(IApplicationInstance appInstance,  
    RTPContext context,  
    String streamId,  
    String streamInfo)
```

Decode SDP info and create RTP stream

Parameters:

appInstance - app instance
context - rtp context
streamId - stream id
streamInfo - stream info (SDP data)

Returns:

rtp stream

decodeStreamInfo

```
public static RTPStream decodeStreamInfo(IApplicationInstance appInstance,  
    RTPStreamContext rtpStreamContext,  
    RTPContext context,  
    String streamId,  
    String streamInfo)
```

Decode SDP info and create RTP stream

Parameters:

appInstance - app instance
rtpStreamContext - rtp stream context
context - rtp context
streamId - stream id
streamInfo - stream info (SDP data)

Returns:

rtp stream

formatH264CodecConfigPacket

```
public static byte[] formatH264CodecConfigPacket(byte[] sps,  
    java.util.List ppss,  
    byte[] profileLevel)
```

Format codec config info, Internal use.

Parameters:

sps
ppss
profileLevel

Returns:

bytes

(continued from last page)

formatH264CodecConfig

```
public static byte[] formatH264CodecConfig(byte[] sps,  
      java.util.List ppss,  
      byte[] profileLevel)
```

Format codec config info, Internal use.

Parameters:

- sps
- ppss
- profileLevel

Returns:

- bytes

com.wowza.wms.util Class StreamUtils

java.lang.Object

└─com.wowza.wms.util.StreamUtils

public class **StreamUtils**
extends Object

Constructor Summary

public	StreamUtils()
--------	-------------------------------

Method Summary

static int	directOutput (byte[] dataBuffer, int[] headerValues, boolean isAbsTimecode, int src, java.io.OutputStream out, AMFObj wmsObj, byte[] workBuffer, int chunkSize) Direct output, Internal use.
static double	getStreamBitrate (IApplicationInstance appInstance, String streamName)
static double	getStreamBitrate (IMediaStream stream) Get the approximate bitrate of a media file in bits/per-second.
static double	getStreamLength (IApplicationInstance appInstance, String streamName) Get the duration of a media file in seconds.
static double	getStreamLength (IMediaStream stream) Get the duration of a media file in seconds.
static boolean	loadConfigFile (StreamList streamDefs, String fileURL) Load Streams.xml, Internal use.
static int	packetOutput (java.io.OutputStream out, IMediaStream stream, AMFPacket packet, long timecode, AMFObj wmsObj, byte[] workBuffer, int chunkSize) Packet output.
static int	packetOutput (java.io.OutputStream out, IMediaStream stream, AMFPacket packet, long timecode, AMFObj wmsObj, byte[] workBuffer, int chunkSize, boolean referenceWrite) Packet output.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

(continued from last page)

StreamUtils

```
public StreamUtils()
```

Methods

getStreamLength

```
public static double getStreamLength(IMediaStream stream)
```

Get the duration of a media file in seconds. This will work correctly with all RandomAccessReaders and caching technology

Parameters:

stream - stream

Returns:

duration in seconds

getStreamBitrate

```
public static double getStreamBitrate(IMediaStream stream)
```

Get the approximate bitrate of a media file in bits/per-second. This will work correctly with all RandomAccessReaders and caching technology.

Parameters:

stream - stream

Returns:

bitrate in bits/per-second

getStreamLength

```
public static double getStreamLength(IApplicationInstance appInstance,  
String streamName)
```

Get the duration of a media file in seconds. This will work correctly with all RandomAccessReaders and caching technology

Parameters:

appInstance - application instance

streamName - stream name

Returns:

duration in seconds

getStreamBitrate

```
public static double getStreamBitrate(IApplicationInstance appInstance,  
String streamName)
```

(continued from last page)

directOutput

```
public static int directOutput(byte[] dataBuffer,  
    int[] headerValues,  
    boolean isAbsTimecode,  
    int src,  
    java.io.OutputStream out,  
    AMFObj wmsObj,  
    byte[] workBuffer,  
    int chunkSize)
```

Direct output. Internal use.

Parameters:

dataBuffer
headerValues
isAbsTimecode
src
out
wmsObj
workBuffer
chunkSize

Returns:

c

packetOutput

```
public static int packetOutput(java.io.OutputStream out,  
    IMediaStream stream,  
    AMFPacket packet,  
    long timecode,  
    AMFObj wmsObj,  
    byte[] workBuffer,  
    int chunkSize)
```

Packet output. Internal use.

Parameters:

out
stream
packet
timecode
wmsObj
workBuffer
chunkSize

Returns:

bytes

packetOutput

```
public static int packetOutput(java.io.OutputStream out,  
    IMediaStream stream,  
    AMFPacket packet,  
    long timecode,  
    AMFObj wmsObj,  
    byte[] workBuffer,  
    int chunkSize,  
    boolean referenceWrite)
```

Packet output. Internal use.

(continued from last page)

Parameters:

out
stream
packet
timecode
wmsObj
workBuffer
chunkSize
referenceWrite

Returns:

bytes

loadConfigFile

```
public static boolean loadConfigFile(StreamList streamDefs,  
    String fileURL)
```

Load Streams.xml, Internal use.

Parameters:

streamDefs
fileURL

Package

com.wowza.wms.util.crontab

com.wowza.wms.util.crontab Class CrontabEvent

java.lang.Object

└─com.wowza.wms.util.crontab.CrontabEvent

All Implemented Interfaces:

Comparable

public class **CrontabEvent**
extends Object
implements Comparable

Holds an event's schedule and is responsible for calculating event activation.

Field Summary

public static final	DAY Value: 2
public static	DT_FORMAT
public static final	HOUR Value: 1
public static final	MINUTE Value: 0
public static final	MONTH Value: 3
public static final	WEEKDAY Value: 5
public static final	YEAR Value: 4

Constructor Summary

public	CrontabEvent ()
--------	----------------------------------

Method Summary

int	compareTo (CrontabEvent o) Allows CronEvents to be sorted via Java Sort Class.
void	dump () dumps this event's information to the console

String	<u>getError()</u>
int	<u>getErrorField()</u>
<u>ICrontabEventHandler</u>	<u>getEventHandler()</u>
String	<u>getEventStr()</u>
CrontabField	<u>getField()</u> (int whichField)
org.joda.time.MutableDateTime	<u>getLastRun()</u>
org.joda.time.MutableDateTime	<u>getNextRun()</u>
String	<u>getTarget()</u>
boolean	<u>isDebugEnabled()</u>
boolean	<u>isExpired()</u> Indicates whether or not all execution dates for this event occur in the past
boolean	<u>isMatch()</u> (org.joda.time.MutableDateTime date) Determines if this event will fire on the date passed in
void	<u>setDebug()</u> (boolean debug) turns on/off debug logging
void	<u>setError()</u> (String error)
void	<u>setErrorField()</u> (int errorField)
void	<u>setEventHandler()</u> (<u>ICrontabEventHandler</u> handler)
void	<u>setEventStr()</u> (String entry)
void	<u>setField()</u> (int whichField, CrontabField field)
void	<u>setLastRun()</u> Sets the last run time to now
void	<u>setLastRun()</u> (org.joda.time.MutableDateTime date)
void	<u>setTarget()</u> (String target)
boolean	<u>start()</u> Starts this events timer
void	<u>stop()</u> stops this events timer

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

Methods inherited from interface `java.lang.Comparable`

```
compareTo
```

Fields

DT_FORMAT

```
public static org.joda.time.format.DateTimeFormatter DT_FORMAT
```

MINUTE

```
public static final int MINUTE
```

Constant value: **0**

HOURL

```
public static final int HOUR
```

Constant value: **1**

DAY

```
public static final int DAY
```

Constant value: **2**

MONTH

```
public static final int MONTH
```

Constant value: **3**

YEAR

```
public static final int YEAR
```

Constant value: **4**

WEEKDAY

```
public static final int WEEKDAY
```

Constant value: **5**

Constructors

CrontabEvent

```
public CrontabEvent()
```

Methods

start

```
public boolean start()
```

Starts this events timer

Returns:

true - if timer is started, else false

stop

```
public void stop()
```

stops this events timer

isMatch

```
public boolean isMatch(org.joda.time.MutableDateTime date)
```

Determines if this event will fire on the date passed in

Parameters:

date

Returns:

boolean

setField

```
public void setField(int whichField,  
    CrontabField field)
```

getField

```
public CrontabField getField(int whichField)
```

setTarget

```
public void setTarget(String target)
```

(continued from last page)

getTarget

```
public String getTarget()
```

setEventStr

```
public void setEventStr(String entry)
```

getEventStr

```
public String getEventStr()
```

getNextRun

```
public org.joda.time.MutableDateTime getNextRun()
```

getLastRun

```
public org.joda.time.MutableDateTime getLastRun()
```

setLastRun

```
public void setLastRun(org.joda.time.MutableDateTime date)
```

getError

```
public String getError()
```

setError

```
public void setError(String error)
```

getErrorField

```
public int getErrorField()
```

setErrorField

```
public void setErrorField(int errorField)
```

(continued from last page)

setLastRun

```
public void setLastRun()
```

Sets the last run time to now

setDebug

```
public void setDebug(boolean debug)
```

turns on/off debug logging

Parameters:

debug - - true/false

isDebugEnabled

```
public boolean isDebugEnabled()
```

setEventHandler

```
public void setEventHandler(ICrontabEventHandler handler)
```

getEventHandler

```
public ICrontabEventHandler getEventHandler()
```

isExpired

```
public boolean isExpired()
```

Indicates whether or not all execution dates for this event occur in the past

Returns:

true/false

compareTo

```
public int compareTo(CrontabEvent o)
```

Allows CronEvents to be sorted via Java Sort Class. Compares nextRun dates of two CronEvents where earlier events are considered "less" than later events, (e.g. today < tomorrow) and nulls are "greater" than events with dates (e.g. tomorrow < null) This provides the logic to sort events by soonest to latest with nulls at the end

Parameters:

o - - CronEvent to compare this event to

Returns:

-1,0,1 - if equal, if this event is greater than

(continued from last page)

dump

```
public void dump()
```

 dumps this event's information to the console

com.wowza.wms.util.crontab Interface ICrontabEventHandler

public interface **ICrontabEventHandler**
extends

EventHandler functions are called from a Java timer task context and therefore should execute very quickly so as not to delay execution of subsequent timers. If there is heavy processing which needs to take place, the implementation for these functions should spawn a thread to do it.

Method Summary

void	onCronEvent (CrontabEvent event)
------	---

Methods

onCronEvent

public void **onCronEvent**([CrontabEvent](#) event)

Package

com.wowza.wms.vhost

com.wowza.wms.vhost Class HostPort

java.lang.Object

└─com.wowza.wms.vhost.HostPort

public class **HostPort**
extends Object

HostPort: data object that describes a socket connection. The address can be defined by ipAddress or by domainName. It can also contain a reference to an SSLFactory class that can be used to create a secure connection to the server.

Field Summary

public static final	TYPE_ADMIN Value: Admin
public static final	TYPE_STREAMING Value: Streaming

Constructor Summary

public	HostPort() Create an empty HostPort object
--------	---

Method Summary

void	addHttpProvider() (IHTTPProvider httpProvider)
void	addHttpProvider() (IHTTPProvider2 httpProvider)
void	addHttpProvider() (int index, IHTTPProvider2 httpProvider)
void	addHTTPStreamerAdapterID() (String ID)
void	configureSocketAcceptor() (org.apache.mina.transport.socket.nio.SocketAcceptorConfig socketConfig) Configure a socketAcceport
java.net.InetAddress	getAddress() Get the ipAddress as an InetAddress object
String	getAddressRawStr()
String	getAddressStr() Get a String representation of the address
HostPortConfig	getConfiguration() Get the socket configuration

String	<u>getHTTPIdent2Response()</u>
String	<u>getHTTPIdentResponse()</u>
java.util.List	<u>getHttpProviders()</u>
java.util.List	<u>getHTTPStreamerAdapterIDs()</u>
String	<u>getName()</u>
int	<u>getPort()</u> Get port
int	<u>getProcessorCount()</u> Get the number of threads to use to service this incoming port
HostPortSSLConfig	<u>getSSLConfig()</u>
String	<u>getSslFactoryClass()</u> Get full class name or SSLFactory class
String	<u>getTypeStr()</u>
boolean	<u>isAdmin()</u>
boolean	<u>isStreaming()</u>
boolean	<u>isSuspended()</u>
void	<u>setDomainName()</u> (String domainName) Set domainName.
void	<u>setHTTPIdent2Response()</u> (String httpIdent2Response)
void	<u>setHTTPIdentResponse()</u> (String httpIdentResponse)
void	<u>setIpAddress()</u> (String ipAddress) Set ipAddress for object.
void	<u>setName()</u> (String name)
void	<u>setPort()</u> (int port) Set port
void	<u>setProcessorCount()</u> (int processorCount) Set the number of threads to use to service this incoming port
void	<u>setSSLConfig()</u> (HostPortSSLConfig sslConfig)
void	<u>setSslFactoryClass()</u> (String sslFactoryClass) Set full class name of SSLFactory class

void	<u>setSuspended</u> (boolean isSuspended)
void	<u>setTypeStr</u> (String typeStr)
int	<u>size</u> (int index, <u>IHTTPProvider2</u> httpProvider)
String	<u>toString</u> () Return object as formatted string
String	<u>toString</u> (boolean mBeanSafe) Return object as formatted string

Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Fields

TYPE_ADMIN

```
public static final java.lang.String TYPE_ADMIN
```

Constant value: **Admin**

TYPE_STREAMING

```
public static final java.lang.String TYPE_STREAMING
```

Constant value: **Streaming**

Constructors

HostPort

```
public HostPort()
```

Create an empty HostPort object

Methods

setDomainName

```
public void setDomainName(String domainName)
```

Set domainName. A domainName of * defines a local connection.

Parameters:

domainName - domainName like www.mycompany.com or * for local

(continued from last page)

getAddress

```
public java.net.InetAddress getAddress()
```

Get the ipAddress as an InetAddress object

Returns:

ipAddress as an InetAddress object. At this point the domainName (if specified) will be resolved.

getAddressStr

```
public String getAddressStr()
```

Get a String representation of the address

Returns:

String representation of the address/domainName. If * it will return [any]

getAddressRawStr

```
public String getAddressRawStr()
```

setIpAddress

```
public void setIpAddress(String ipAddress)
```

Set ipAddress for object. A ipAddress of * defines a local connection.

Parameters:

ipAddress - ipAddress like 127.0.0.1 or * for local

getPort

```
public int getPort()
```

Get port

Returns:

port

setPort

```
public void setPort(int port)
```

Set port

Parameters:

port - port

getSslFactoryClass

```
public String getSslFactoryClass()
```

Get full class name or SSLFactory class

Returns:

(continued from last page)

full class name or SSLFactory or "" if not specified

setSslFactoryClass

```
public void setSslFactoryClass(String sslFactoryClass)
```

Set full class name of SSLFactory class

Parameters:

sslFactoryClass - full class name or SSLFactory or "" if not specified

getProcessorCount

```
public int getProcessorCount()
```

Get the number of threads to use to service this incoming port

Returns:

number of processor threads

setProcessorCount

```
public void setProcessorCount(int processorCount)
```

Set the number of threads to use to service this incoming port

Parameters:

processorCount - number of processor threads

toString

```
public String toString(boolean mBeanSafe)
```

Return object as formatted string

Parameters:

mBeanSafe - make the name safe for JMX management interface

Returns:

formatted string

toString

```
public String toString()
```

Return object as formatted string

Returns:

formatted string

getConfiguration

```
public HostPortConfig getConfiguration()
```

Get the socket configuration

Returns:

socket configuration

configureSocketAcceptor

```
public void  
configureSocketAcceptor(org.apache.mina.transport.socket.nio.SocketAcceptorConfig  
socketConfig)
```

Configure a socketAcceptor

Parameters:

socketConfig - socket acceptor

getHttpProviders

```
public java.util.List getHttpProviders()
```

addHttpProvider

```
public void addHttpProvider(IHTTPProvider httpProvider)
```

addHttpProvider

```
public void addHttpProvider(IHTTPProvider2 httpProvider)
```

addHttpProvider

```
public void addHttpProvider(int index,  
    IHTTPProvider2 httpProvider)
```

size

```
public int size(int index,  
    IHTTPProvider2 httpProvider)
```

isSuspended

```
public boolean isSuspended()
```

setSuspended

```
public void setSuspended(boolean isSuspended)
```

(continued from last page)

getHTTPStreamerAdapterIDs

```
public java.util.List getHTTPStreamerAdapterIDs()
```

addHTTPStreamerAdapterID

```
public void addHTTPStreamerAdapterID(String ID)
```

getSSLConfig

```
public HostPortSSLConfig getSSLConfig()
```

setSSLConfig

```
public void setSSLConfig(HostPortSSLConfig sslConfig)
```

getHTTPIdentResponse

```
public String getHTTPIdentResponse()
```

setHTTPIdentResponse

```
public void setHTTPIdentResponse(String httpIdentResponse)
```

getHTTPIdent2Response

```
public String getHTTPIdent2Response()
```

setHTTPIdent2Response

```
public void setHTTPIdent2Response(String httpIdent2Response)
```

getTypeStr

```
public String getTypeStr()
```

setTypeStr

```
public void setTypeStr(String typeStr)
```

(continued from last page)

getName

```
public String getName()
```

setName

```
public void setName(String name)
```

isAdmin

```
public boolean isAdmin()
```

isStreaming

```
public boolean isStreaming()
```

com.wowza.wms.vhost Class HostPortList

java.lang.Object

└--com.wowza.wms.vhost.HostPortList

```
public class HostPortList
extends Object
```

HostPortList: data object that contains a collection of HostPort objects.

Constructor Summary

public	HostPortList() Create empty HostPortList
--------	---

Method Summary

void	add(HostPort hostPort) Add HostPort object
HostPort	get(int index) Get HostPort object at index, null if out of bounds
int	size() Get number of HostPort objects

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

HostPortList

```
public HostPortList()
```

Create empty HostPortList

Methods

add

```
public void add(HostPort hostPort)
```

Add HostPort object

Parameters:

hostPort

size

```
public int size()
```

Get number of HostPort objects

Returns:

number of HostPort objects

get

```
public HostPort get(int index)
```

Get HostPort object at index, null if out of bounds

Parameters:

index - index

Returns:

HostPort object at index, null if out of bounds

com.wowza.wms.vhost Interface IAcceptorNotify

public interface **IAcceptorNotify**
extends

IAcceptorNotify: listener interface used by IVHost addAcceptorListener

Method Summary

void	onAcceptorCreate (HostPort hostPort, java.util.Map acceptorMap) Triggered when a new acceptor is created
void	onAcceptorDestroy (HostPort hostPort, java.util.Map acceptorMap) Triggered when a new acceptor is deleted

Methods

onAcceptorCreate

```
public void onAcceptorCreate(HostPort hostPort,  
                             java.util.Map acceptorMap)
```

Triggered when a new acceptor is created

Parameters:

hostPort - HostPort object

acceptorMap - Map of acceptors

onAcceptorDestroy

```
public void onAcceptorDestroy(HostPort hostPort,  
                               java.util.Map acceptorMap)
```

Triggered when a new acceptor is deleted

Parameters:

hostPort - HostPort object

acceptorMap - Map of acceptors

com.wowza.wms.vhost Interface IVHost

public interface **IVHost**
extends

IVHost: public interface to VHost object.

Field Summary

public static final	ACCEPTORS_ACCEPTOR Acceptor types: acceptor Value: acceptor
public static final	ACCEPTORS_HANDLERADAPTER Acceptor types: handler Value: handlerAdapter
public static final	CODEC_AUDIO_AAC Value: 10
public static final	CODEC_AUDIO_AC3 Value: 1
public static final	CODEC_AUDIO_G711_ALAW Value: 7
public static final	CODEC_AUDIO_G711_MULAW Value: 8
public static final	CODEC_AUDIO_MP3 Value: 2
public static final	CODEC_AUDIO_MP3_8 Value: 15
public static final	CODEC_AUDIO_NELLYMOSER Value: 6
public static final	CODEC_AUDIO_NELLYMOSER_16MONO Value: 4
public static final	CODEC_AUDIO_NELLYMOSER_8MONO Value: 5
public static final	CODEC_AUDIO_PCM_BE Value: 0

public static final	CODEC_AUDIO_PCM_LE Value: 3
public static final	CODEC_AUDIO_PCM_SWF Value: 1
public static final	CODEC_AUDIO_RESERVED Value: 9
public static final	CODEC_AUDIO_SPEEX Value: 11
public static final	CODEC_AUDIO_UNKNOWN Value: -1
public static final	CODEC_AUDIO_VORBIS Value: 9
public static final	CODEC_STREAM_MP2T Value: 0
public static final	CODEC_STREAM_UNKNOWN Value: -1
public static final	CODEC_VIDEO_H263 Value: 9
public static final	CODEC_VIDEO_H264 Value: 7
public static final	CODEC_VIDEO_H265 Value: 12
public static final	CODEC_VIDEO_MPEG2 Value: 11
public static final	CODEC_VIDEO_MPEG4 Value: 10
public static final	CODEC_VIDEO_SCREEN Value: 3
public static final	CODEC_VIDEO_SCREEN2 Value: 6
public static final	CODEC_VIDEO_SPARK Value: 2

public static final	CODEC_VIDEO_UNKNOWN Value: -1
public static final	CODEC_VIDEO_VP6 Value: 4
public static final	CODEC_VIDEO_VP6A Value: 5
public static final	CODEC_VIDEO_VP8 Value: 8
public static final	CONTENTTYPE_ACKBANDWIDTH AMF Content type: set acknowledge bandwidth size Value: 5
public static final	CONTENTTYPE_AUDIO AMF Content type: audio packet Value: 8
public static final	CONTENTTYPE_BUFFERSIZE AMF Content type: set buffer size Value: 4
public static final	CONTENTTYPE_DATA AMF Content type: data packet Value: 18
public static final	CONTENTTYPE_DATA0 AMF Content type: data packet (AMF0) Value: 18
public static final	CONTENTTYPE_DATA3 AMF Content type: data packet (AMF3) Value: 15
public static final	CONTENTTYPE_FUNCTION AMF Content type: function data (AMF0) Value: 20
public static final	CONTENTTYPE_FUNCTION0 AMF Content type: function data (AMF0) Value: 20
public static final	CONTENTTYPE_FUNCTION3 AMF Content type: function data (AMF3) Value: 17
public static final	CONTENTTYPE_MEDIACHUNK AMF Content type: media chunk Value: 22
public static final	CONTENTTYPE_PLAYCALLBACK AMF Content type: play callback Value: 127

public static final	CONTENTTYPE_SETBANDWIDTH AMF Content type: set bandwidth size Value: 6
public static final	CONTENTTYPE_SETCHUNKSIZE AMF Content type: set packet chunk size Value: 1
public static final	CONTENTTYPE_SHAREDOBJECTS AMF Content type: shared object packet (AMF0) Value: 19
public static final	CONTENTTYPE_SHAREDOBJECTS0 AMF Content type: shared object packet (AMF0) Value: 19
public static final	CONTENTTYPE_SHAREDOBJECTS3 AMF Content type: shared object packet (AMF3) Value: 16
public static final	CONTENTTYPE_UNKNOWN AMF Content type: unknown Value: 0
public static final	CONTENTTYPE_VIDEO AMF Content type: video packet Value: 9
public static final	CONTENTTYPE_WATCHDOG AMF Content type: watch dog Value: 3
public static final	COUNTER_HTTPCUPERTINO Value: 2
public static final	COUNTER_HTTPDVRCHUNKS Value: 7
public static final	COUNTER_HTTPMPEGDASH Value: 6
public static final	COUNTER_HTTPSANJOSE Value: 4
public static final	COUNTER_HTTPSMOOTH Value: 3
public static final	COUNTER_HTTPWEBM Value: 5
public static final	COUNTER_RTMP Value: 0

public static final	<u>COUNTER_RTP</u> Value: 1
public static final	<u>COUNTER_TOTAL</u> Value: 8
public static final	<u>FILEFORMAT_FLV</u> Value: 1
public static final	<u>FILEFORMAT_MP4</u> Value: 2
public static final	<u>FILEFORMAT_UNKNOWN</u> Value: -1
public static final	<u>LICENSECOUNTER_DRM_BUYDRM_LIVE</u> Value: 9
public static final	<u>LICENSECOUNTER_DRM_BUYDRM_VOD</u> Value: 10
public static final	<u>LICENSECOUNTER_DRM_EZDRM_LIVE</u> Value: 4
public static final	<u>LICENSECOUNTER_DRM_EZDRM_VOD</u> Value: 5
public static final	<u>LICENSECOUNTER_DRM_VERIMATRIX_LIVE</u> Value: 6
public static final	<u>LICENSECOUNTER_DRM_VERIMATRIX_VOD</u> Value: 7
public static final	<u>LICENSECOUNTER_NDVR</u> Value: 3
public static final	<u>LICENSECOUNTER_PUBLISHER</u> Value: 0
public static final	<u>LICENSECOUNTER_PUBLISHERTRANSCODER</u> Value: 8
public static final	<u>LICENSECOUNTER_TOTAL</u> Value: 20
public static final	<u>LICENSECOUNTER_TRANSCODE_DECODE</u> Value: 1

public static final	LICENSECOUNTER_TRANSCODE_DECODECOUNTAUDIO Value: 14
public static final	LICENSECOUNTER_TRANSCODE_DECODECOUNTAUDIOVIDEO Value: 16
public static final	LICENSECOUNTER_TRANSCODE_DECODECOUNTVIDEO Value: 15
public static final	LICENSECOUNTER_TRANSCODE_DECODEPOLLING Value: 11
public static final	LICENSECOUNTER_TRANSCODE_ENCODE Value: 2
public static final	LICENSECOUNTER_TRANSCODE_ENCODECOUNTAUDIO Value: 17
public static final	LICENSECOUNTER_TRANSCODE_ENCODECOUNTAUDIOVIDEO Value: 19
public static final	LICENSECOUNTER_TRANSCODE_ENCODECOUNTVIDEO Value: 18
public static final	LICENSECOUNTER_TRANSCODE_ENCODEPOLLING Value: 12
public static final	LICENSECOUNTER_TRANSCODE_STREAMNAMES Value: 13
public static final	LICENSECOUNTERS_DRM
public static final	VHOST_DEFAULT Value: _defaultVHost_

Method Summary

void	addAcceptorListener (IAcceptorNotify acceptorListener) Add acceptor listener.
void	addApplicationListener (IApplicationNotify applicationListener) Add application listener.
void	addIdleWorkerListener (IIdleWorkerNotify idleWorkerListener) Add idleWorker listener.
void	addStartupStream (StartupStream startupStream) Add a stream to the list of streams to start and virtual host startup
boolean	applicationExists (String name) Return true if an application folder exists for this application name

void	<u>closeHostPort</u> (<u>HostPort</u> hostPort, boolean isSuspend) Close an individual HostPort
<u>IApplication</u>	<u>getApplication</u> (String applicationName) Get application by name.
java.util.List	<u>getApplicationFolderNames</u> () Get a list of application folder names
edu.emory.mathcs.backport.java.util.concurrent.locks.WMSReadWriteLock	<u>getApplicationLock</u> () Get the object used in synchronized statements to lock and application.
java.util.List	<u>getApplicationNames</u> () Get a list of application names
int	<u>getApplicationTimeout</u> () Get application time out (milliseconds).
AuthenticationList	<u>getAuthenticationList</u> () Get the list of available authentication methods
CEACaptionConversionContext	<u>getCEACaptionConversionContext</u> () Get CEACaptionConversionContextMP3 tag map attached to vHost (CEACaptionConversion.xml).
<u>IClient</u>	<u>getClient</u> (int clientId) Get client by client id.
<u>IClient</u>	<u>getClient</u> (int clientId, boolean create) Get client by client id and create if does not exist.
int	<u>getClientCount</u> () Get number of clients connected to this vHost.
int	<u>getClientIdleFrequency</u> () Get default client idle frequency (milliseconds)
int	<u>getClientTimeout</u> () Get client timeout.
<u>ConnectionCounter</u>	<u>getConnectionCounter</u> () Get vHost connection counter.
ConnectionCounterSimple	<u>getConnectionCounter</u> (int counterIndex) Get vHost connection counter for a specific technology (see IVHost.COUNTER_*)
int	<u>getConnectionLimit</u> () Get vHost connection limit.
int	<u>getCoreHandlerPoolSize</u> () Get the handler core thread pool size.
int	<u>getCoreTransportPoolSize</u> () Get the transport core thread pool size.
String	<u>getDateStarted</u> () Get date and time the server was started.

DvrRecorderList	<u>getDvrRecorderList()</u> Get the DvrRecorderList
DvrStoreList	<u>getDvrStoreList()</u> Get the list of DVR Stores
java.util.Properties	<u>getDynamicLogProperties()</u> Get the dynamic log properties defined at the vhost level in conf/log4j.properties
int	<u>getFileIOPoolSize()</u> Get the default file io pool size.
<u>ThreadPool</u>	<u>getHandlerThreadPool()</u> Get the VHost handler thread pool.
String	<u>getHomePath()</u> Get vHost configuration path.
<u>HostPortList</u>	<u>getHostPortsList()</u> Get list of host port definitions for vHost.
<u>IHTTPStreamerAdapter</u>	<u>getHTTPStreamerAdapter(String ID)</u> Get an HTTPStreamerAdapter by ID
java.util.List	<u>getHTTPStreamerAdapterIDs()</u> Get a list of HTTPStreamerAdapter IDs
HTTPStreamerContext	<u>getHTTPStreamerContext()</u> Get the HTTPStreamer (Cupertino Streaming and Silverlight Smooth Streaming) host context
HTTPStreamerList	<u>getHTTPStreamerList()</u> Get the list of HTTPStreamers
int	<u>getHTTPStreamerMaxPathLen()</u> Get the max HTTP streamer path length
<u>IVHostHTTPStreamerRequestValidator</u>	<u>getHTTPStreamerRequestValidator()</u> Get the HTTP streamer request validator
int	<u>getIdleCheckFrequency()</u> Get idle check frequency (milliseconds)
int	<u>getIdleMinimumWaitTime()</u> Get the minimum time (milliseconds) the idle worker thread will sleep before generating idle events
int	<u>getIdleWorkerCount()</u> Get number of threads used to generate idle events
IdleWorkersUtil	<u>getIdleWorkers()</u> Get the idle worker utility
<u>IOPerformanceCounter</u>	<u>getIoPerformanceCounter()</u> Get vHost IO performance counter.
<u>IOPerformanceCounter</u>	<u>getIoPerformanceCounter(int counterIndex)</u> Get vHost IO performance counter for a specific technology (see IVHost.COUNTER_*)
IOScheduler	<u>getIOScheduler()</u> Get IO scheduler for vHost.

int	getKeepAliveTimeout() Get the RTMPT connection keep alive timeout
LiveStreamPacketizerList	getLiveStreamPacketizerList() Get the LiveStreamPacketizerList
ILiveStreamRecordManager	getLiveStreamRecordManager() Returns the LiveStreamRecordManager associated with this vhost This object contains the active LiveStreamRecorders
LiveStreamTranscoderList	getLiveStreamTranscoderList() Get the LiveStreamTranscoderList
long	getLiveStreamTranscoderSessionCount() Get the current number of live stream transcoder sessions.
WMSProperties	getManagerProperties() Get VHost's manager's properties collection
int	getMaximumPendingReadBytes() Set maximum number of bytes a client connection can have waiting to be written before the connection is terminated.
int	getMaximumPendingWriteBytes() Get maximum number a bytes a client connection can have waiting to be sent before the connection is terminated.
int	getMaximumSetBufferTime() Get maximum number of milliseconds allowed for the NetStream.setBufferTime(secs) call.
MediaCasterList	getMediaCasterList() Get the list of media caster definitions (MediaCaster.xml)
MediaCasterSettings	getMediaCasterSettings() Get the media caster settings
int	getMediaReaderContentType(String mediaType) Get the content type of a media stream name prefix (see IMediaReader.CONTENTTYPE_*)
MediaReaderList	getMediaReaders() Get the media readers attached to vHost (MediaReaders.xml).
MediaWriterList	getMediaWriters() Get the media writers attached to vHost (MediaWriters.xml).
java.util.Map	getMp3TagMap() Get MP3 tag map attached to vHost (MP3Tags.xml).
String	getName() Get vHost name
HostPortConfig	getNetConnectionHostPortConfig() Get the socket configuration for server to server connections
int	getNetConnectionIdleFrequency() Get server to server idle frequency (milliseconds)
int	getNetConnectionProcessorCount() Get net connection processor count.

int	<u>getNextNetConnectionId()</u> Get next connection id.
int	<u>getPingTimeout()</u> Get ping timeout (milliseconds)
<u>WMSProperties</u>	<u>getProperties()</u> Get properties attached to this vHost.
String	<u>getProperty(String key)</u> Get virtual host property.
void	<u>getProtocolUsage(boolean[] protocolsInUse)</u> Get the protocols in use by this application instance (see IApplicationInstance.PROTOCOLUSAGE_*)
<u>PushPublishContext</u>	<u>getPushPublishContext()</u> Get the push publish context
<u>RTPContext</u>	<u>getRTPContext()</u> Get the RTP (real time protocol) virtual host context
com.wowza.wms.rtp.transport.RTPUDPDatagramConfig	<u>getRTPDatagramConfigIncoming()</u> Get the RTP Datagram Socket configuration
com.wowza.wms.rtp.transport.RTPUDPDatagramConfig	<u>getRTPDatagramConfigOutgoing()</u> Get the RTP Datagram Socket configuration
int	<u>getRTPIidleFrequency()</u> Get default RTP idle frequency (milliseconds)
int	<u>getRTSPMaxPathLen()</u> Get the max RTSP stream path length
<u>IVHostRTSPRequestValidator</u>	<u>getRTSPRequestValidator()</u> Get the RTSP request validator
java.util.List	<u>getStartupStreams()</u> Get the list of streams to start at virtual host startup
int	<u>getStartupStreamsDelayTime()</u> Get startup streams delay time (milliseconds).
<u>StreamList</u>	<u>getStreamTypes()</u> Get default stream type.
<u>ThreadPool</u>	<u>getThreadPool()</u> Get the VHost handler thread pool.
TimedTextProviderConfiguration	<u>getTimedTextProviderConfig()</u> Get the config of TimedTextProviders
String	<u>getTimeRunning()</u> Get the time vHost has been running.
double	<u>getTimeRunningSeconds()</u> Get time running in seconds

com.wowza.wms.transcoder.vod.TranscoderVODContext	getTranscoderVODContext() Get the VOD transcoder context
ThreadPool	getTransportThreadPool() Get the VHost transport thread pool.
com.wowza.wms.rtp.transport.UDPTransportManager	getUDPTransportManager() Get the UDP transport manager.
int	getUnidentifiedSessionTimeout() Get the unidentified session timeout (TCP session that we are not able to identify)
int	getValidationFrequency() Get time between validation pings (milliseconds)
void	init(String basePath) Initialize vHost.
boolean	isApplicationLoaded(String applicationName) Return true is the application is loaded
boolean	isShuttingDown() Is the VHost shutting down
boolean	isStartStarupStreams() Returns true if the startup streams are to start and vhost startup
boolean	isSuspended() Returns true is all HostPorts connected to this VHost are suspended
void	killClient(int clientId) Remove client from vHost and send disconnect message.
void	killRTSPSession(String rtspSessionId) Kill an RTSP connection by the RTSP session id
void	putHTTPStreamerAdapter(String ID, IHTTPStreamerAdapter adapter) Add an HTTPStreamerAdapter
void	removeAcceptorListener(IAcceptorNotify acceptorListener) Remove acceptor listener.
void	removeApplicationListener(IApplicationNotify applicationListener) Remove applation listener.
void	removeClient(int clientId) Remove client from vHost.
void	removeIdleWorkerListener(IIdleWorkerNotify idleWorkerListener) Remove idleWorker listener
void	reparentClient(IClient client) Move a client object to a new vhost.
void	setAdminInterfaceHostPort(HostPort adminInterfaceHostPort) Set admin interface host port (not used)
void	setApplicationTimeout(int applicationTimeout) Set application time out (milliseconds).

void	<u>setClientIdleFrequency</u> (int clientIdleFrequency) Set default client idle frequency (milliseconds)
void	<u>setClientTimeout</u> (int clientTimeout) Set client timeout.
void	<u>setCoreHandlerPoolSize</u> (int corePoolSize) Set the handler core thread pool size.
void	<u>setCoreTransportPoolSize</u> (int corePoolSize) Set the transport core thread pool size.
void	<u>setDynamicLogProperties</u> (java.util.Properties dynamicLogProperties) Set the dynamic log properties set at the vhost level
void	<u>setFileIOPoolSize</u> (int fileIOPoolSize) Set default file io thread pool size.
void	<u>setHTTPStreamerMaxPathLen</u> (int httpStreamerMaxPathLen) Set the max HTTP streamer path length
void	<u>setHTTPStreamerRequestValidator</u> (<u>IVHostHTTPStreamerRequestValidator</u> httpRequestValidator) Set the HTTP streamer request validator
void	<u>setIdleCheckFrequency</u> (int idleCheckFrequency) Set idle check frequency (milliseconds)
void	<u>setIdleMinimumWaitTime</u> (int idleMinimumWaitTime) Set the minimum time (milliseconds) the idle worker thread will sleep before generating idle events
void	<u>setIdleWorkerCount</u> (int idleWorkerCount) Set number of threads used to generate idle events
void	<u>setKeepAliveTimeout</u> (int keepAliveTimeout) Set the RTMPT connection keep alive timeout
void	<u>setMaximumPendingReadBytes</u> (int maximumPendingReaderBytes) Get maximum number of bytes a client connection can have waiting to be written before the connection is terminated.
void	<u>setMaximumPendingWriteBytes</u> (int maximumPendingWriteBytes) Set maximum number a bytes a client connection can have waiting to be sent before the connection is terminated.
void	<u>setMaximumSetBufferTime</u> (int maximumSetBufferTime) Set maximum number of milliseconds allowed for the NetStream.setBufferTime(secs) call.
void	<u>setNetConnectionIdleFrequency</u> (int netConnectionIdleFrequency) Set the server to server idle frequency (milliseconds)
void	<u>setNetConnectionProcessorCount</u> (int netConnectionProcessorCount) Set net connection processor count.
void	<u>setPingTimeout</u> (int pingTimeout) Set ping timeout (milliseconds)
void	<u>setRTPIdleFrequency</u> (int rtpIdleFrequency) Set default RTP idle frequency (milliseconds)

void	<u>setRTSPMaxPathLen</u> (int rtspStreamerMaxPathLen) Set the max RTSP stream path length
void	<u>setRTSPRequestValidator</u> (<u>IVHostRTSPRequestValidator</u> rtspRequestValidator) Set the RTSP request validator
void	<u>setShuttingDown</u> (boolean shuttingDown) Set VHost shutting down flag
void	<u>setStartStarupStreams</u> (boolean startStarupStreams) Set to true to startup startup stream as vhost startup
void	<u>setStartupStreamsDelayTime</u> (int startupStreamsDelayTime) Set startup streams delay time (milliseconds).
void	<u>setUnidentifiedSessionTimeout</u> (int unidentifiedSessionTimeout) Set the unidentified session timeout (TCP session that we are not able to identify)
void	<u>setValidationFrequency</u> (int validationFrequency) Set time between validation pings (milliseconds)
void	<u>shutdown</u> () Shutdown.
void	<u>shutdownApplication</u> (String appName) Shutdown an application by name.
void	<u>shutdownApplication</u> (String appName, boolean deleteFolder) Shutdown an application by name.
boolean	<u>startApplicationInstance</u> (String appName) Start an application instance.
boolean	<u>startApplicationInstance</u> (String appName, String appInstanceName) Start an application instance.
void	<u>startStartupStreams</u> () Method to start startup streams
void	<u>stopStartupStreams</u> () Method to stop startup streams
void	<u>suspendAllHostPorts</u> () Suspend all HostPorts from accepting new connections.
boolean	<u>touchApplicationInstance</u> (String appName) Start an application instance if it is not already started then touch it so it stays loaded for at least 3 seconds.
boolean	<u>touchApplicationInstance</u> (String appName, String appInstanceName) Start an application instance if it is not already started then touch it so it stays loaded for at least 3 seconds.
void	<u>unbindAllHostPorts</u> () Unbind all HostPorts and drop all connections
void	<u>updateLoggingDuration</u> () Internal: update the internal logging values.

Fields

VHOST_DEFAULT

```
public static final java.lang.String VHOST_DEFAULT
```

Constant value: `_defaultVHost_`

LICENSECOUNTER_PUBLISHER

```
public static final int LICENSECOUNTER_PUBLISHER
```

Constant value: `0`

LICENSECOUNTER_TRANSCODE_DECODE

```
public static final int LICENSECOUNTER_TRANSCODE_DECODE
```

Constant value: `1`

LICENSECOUNTER_TRANSCODE_ENCODE

```
public static final int LICENSECOUNTER_TRANSCODE_ENCODE
```

Constant value: `2`

LICENSECOUNTER_NDVR

```
public static final int LICENSECOUNTER_NDVR
```

Constant value: `3`

LICENSECOUNTER_DRM_EZDRM_LIVE

```
public static final int LICENSECOUNTER_DRM_EZDRM_LIVE
```

Constant value: `4`

LICENSECOUNTER_DRM_EZDRM_VOD

```
public static final int LICENSECOUNTER_DRM_EZDRM_VOD
```

Constant value: `5`

LICENSECOUNTER_DRM_VERIMATRIX_LIVE

```
public static final int LICENSECOUNTER_DRM_VERIMATRIX_LIVE
```

Constant value: `6`

LICENSECOUNTER_DRM_VERIMATRIX_VOD

```
public static final int LICENSECOUNTER_DRM_VERIMATRIX_VOD
```

Constant value: **7**

LICENSECOUNTER_PUBLISHERTRANSCODER

```
public static final int LICENSECOUNTER_PUBLISHERTRANSCODER
```

Constant value: **8**

LICENSECOUNTER_DRM_BUYDRM_LIVE

```
public static final int LICENSECOUNTER_DRM_BUYDRM_LIVE
```

Constant value: **9**

LICENSECOUNTER_DRM_BUYDRM_VOD

```
public static final int LICENSECOUNTER_DRM_BUYDRM_VOD
```

Constant value: **10**

LICENSECOUNTER_TRANSCODE_DECODEPOLLING

```
public static final int LICENSECOUNTER_TRANSCODE_DECODEPOLLING
```

Constant value: **11**

LICENSECOUNTER_TRANSCODE_ENCODEPOLLING

```
public static final int LICENSECOUNTER_TRANSCODE_ENCODEPOLLING
```

Constant value: **12**

LICENSECOUNTER_TRANSCODE_STREAMNAMES

```
public static final int LICENSECOUNTER_TRANSCODE_STREAMNAMES
```

Constant value: **13**

LICENSECOUNTER_TRANSCODE_DECODECOUNTAUDIO

```
public static final int LICENSECOUNTER_TRANSCODE_DECODECOUNTAUDIO
```

Constant value: **14**

(continued from last page)

LICENSECOUNTER_TRANSCODE_DECODECOUNTVIDEO

```
public static final int LICENSECOUNTER_TRANSCODE_DECODECOUNTVIDEO
```

Constant value: **15**

LICENSECOUNTER_TRANSCODE_DECODECOUNTAUDIOVIDEO

```
public static final int LICENSECOUNTER_TRANSCODE_DECODECOUNTAUDIOVIDEO
```

Constant value: **16**

LICENSECOUNTER_TRANSCODE_ENCODECOUNTAUDIO

```
public static final int LICENSECOUNTER_TRANSCODE_ENCODECOUNTAUDIO
```

Constant value: **17**

LICENSECOUNTER_TRANSCODE_ENCODECOUNTVIDEO

```
public static final int LICENSECOUNTER_TRANSCODE_ENCODECOUNTVIDEO
```

Constant value: **18**

LICENSECOUNTER_TRANSCODE_ENCODECOUNTAUDIOVIDEO

```
public static final int LICENSECOUNTER_TRANSCODE_ENCODECOUNTAUDIOVIDEO
```

Constant value: **19**

LICENSECOUNTER_TOTAL

```
public static final int LICENSECOUNTER_TOTAL
```

Constant value: **20**

LICENSECOUNTERS_DRM

```
public static final int LICENSECOUNTERS_DRM
```

COUNTER_RTMP

```
public static final int COUNTER_RTMP
```

Constant value: **0**

COUNTER_RTP

```
public static final int COUNTER_RTP
```


(continued from last page)

Constant value: **1**

COUNTER_HTTPCUPERTINO

```
public static final int COUNTER_HTTPCUPERTINO
```

Constant value: **2**

COUNTER_HTTPSMOOTH

```
public static final int COUNTER_HTTPSMOOTH
```

Constant value: **3**

COUNTER_HTTPSANJOSE

```
public static final int COUNTER_HTTPSANJOSE
```

Constant value: **4**

COUNTER_HTTPWEBM

```
public static final int COUNTER_HTTPWEBM
```

Constant value: **5**

COUNTER_HTTPMPEGDASH

```
public static final int COUNTER_HTTPMPEGDASH
```

Constant value: **6**

COUNTER_HTTPDVRCHUNKS

```
public static final int COUNTER_HTTPDVRCHUNKS
```

Constant value: **7**

COUNTER_TOTAL

```
public static final int COUNTER_TOTAL
```

Constant value: **8**

CODEC_VIDEO_UNKNOWN

```
public static final int CODEC_VIDEO_UNKNOWN
```

Constant value: **-1**

(continued from last page)

CODEC_VIDEO_SPARK

```
public static final int CODEC_VIDEO_SPARK
```

Constant value: **2**

CODEC_VIDEO_SCREEN

```
public static final int CODEC_VIDEO_SCREEN
```

Constant value: **3**

CODEC_VIDEO_VP6

```
public static final int CODEC_VIDEO_VP6
```

Constant value: **4**

CODEC_VIDEO_VP6A

```
public static final int CODEC_VIDEO_VP6A
```

Constant value: **5**

CODEC_VIDEO_SCREEN2

```
public static final int CODEC_VIDEO_SCREEN2
```

Constant value: **6**

CODEC_VIDEO_H264

```
public static final int CODEC_VIDEO_H264
```

Constant value: **7**

CODEC_VIDEO_VP8

```
public static final int CODEC_VIDEO_VP8
```

Constant value: **8**

CODEC_VIDEO_H263

```
public static final int CODEC_VIDEO_H263
```

Constant value: **9**

CODEC_VIDEO_MPEG4

```
public static final int CODEC_VIDEO_MPEG4
```

(continued from last page)

Constant value: **10**

CODEC_VIDEO_MPEG2

```
public static final int CODEC_VIDEO_MPEG2
```

Constant value: **11**

CODEC_VIDEO_H265

```
public static final int CODEC_VIDEO_H265
```

Constant value: **12**

CODEC_AUDIO_UNKNOWN

```
public static final int CODEC_AUDIO_UNKNOWN
```

Constant value: **-1**

CODEC_AUDIO_PCM_BE

```
public static final int CODEC_AUDIO_PCM_BE
```

Constant value: **0**

CODEC_AUDIO_PCM_SWF

```
public static final int CODEC_AUDIO_PCM_SWF
```

Constant value: **1**

CODEC_AUDIO_AC3

```
public static final int CODEC_AUDIO_AC3
```

Constant value: **1**

CODEC_AUDIO_MP3

```
public static final int CODEC_AUDIO_MP3
```

Constant value: **2**

CODEC_AUDIO_PCM_LE

```
public static final int CODEC_AUDIO_PCM_LE
```

Constant value: **3**

CODEC_AUDIO_NELLYMOSER_16MONO

```
public static final int CODEC_AUDIO_NELLYMOSER_16MONO
```

Constant value: **4**

CODEC_AUDIO_NELLYMOSER_8MONO

```
public static final int CODEC_AUDIO_NELLYMOSER_8MONO
```

Constant value: **5**

CODEC_AUDIO_NELLYMOSER

```
public static final int CODEC_AUDIO_NELLYMOSER
```

Constant value: **6**

CODEC_AUDIO_G711_ALAW

```
public static final int CODEC_AUDIO_G711_ALAW
```

Constant value: **7**

CODEC_AUDIO_G711_MULAW

```
public static final int CODEC_AUDIO_G711_MULAW
```

Constant value: **8**

CODEC_AUDIO_RESERVED

```
public static final int CODEC_AUDIO_RESERVED
```

Constant value: **9**

CODEC_AUDIO_VORBIS

```
public static final int CODEC_AUDIO_VORBIS
```

Constant value: **9**

CODEC_AUDIO_AAC

```
public static final int CODEC_AUDIO_AAC
```

Constant value: **10**

(continued from last page)

CODEC_AUDIO_SPEEX

```
public static final int CODEC_AUDIO_SPEEX
```

Constant value: **11**

CODEC_AUDIO_MP3_8

```
public static final int CODEC_AUDIO_MP3_8
```

Constant value: **15**

CODEC_STREAM_UNKNOWN

```
public static final int CODEC_STREAM_UNKNOWN
```

Constant value: **-1**

CODEC_STREAM_MP2T

```
public static final int CODEC_STREAM_MP2T
```

Constant value: **0**

FILEFORMAT_UNKNOWN

```
public static final int FILEFORMAT_UNKNOWN
```

Constant value: **-1**

FILEFORMAT_FLV

```
public static final int FILEFORMAT_FLV
```

Constant value: **1**

FILEFORMAT_MP4

```
public static final int FILEFORMAT_MP4
```

Constant value: **2**

CONTENTTYPE_UNKNOWN

```
public static final byte CONTENTTYPE_UNKNOWN
```

AMF Content type: unknown
Constant value: **0**

CONTENTTYPE_SETCHUNKSIZE

```
public static final byte CONTENTTYPE_SETCHUNKSIZE
```

(continued from last page)

AMF Content type: set packet chunk size
Constant value: **1**

CONTENTTYPE_WATCHDOG

public static final byte **CONTENTTYPE_WATCHDOG**

AMF Content type: watch dog
Constant value: **3**

CONTENTTYPE_BUFFERSIZE

public static final byte **CONTENTTYPE_BUFFERSIZE**

AMF Content type: set buffer size
Constant value: **4**

CONTENTTYPE_ACKBANDWIDTH

public static final byte **CONTENTTYPE_ACKBANDWIDTH**

AMF Content type: set acknowledge bandwidth size
Constant value: **5**

CONTENTTYPE_SETBANDWIDTH

public static final byte **CONTENTTYPE_SETBANDWIDTH**

AMF Content type: set bandwidth size
Constant value: **6**

CONTENTTYPE_AUDIO

public static final byte **CONTENTTYPE_AUDIO**

AMF Content type: audio packet
Constant value: **8**

CONTENTTYPE_VIDEO

public static final byte **CONTENTTYPE_VIDEO**

AMF Content type: video packet
Constant value: **9**

CONTENTTYPE_DATA

public static final byte **CONTENTTYPE_DATA**

AMF Content type: data packet
Constant value: **18**

CONTENTTYPE_DATA0

public static final byte **CONTENTTYPE_DATA0**

AMF Content type: data packet (AMF0)
Constant value: **18**

CONTENTTYPE_DATA3

public static final byte **CONTENTTYPE_DATA3**

AMF Content type: data packet (AMF3)
Constant value: **15**

CONTENTTYPE_SHAREDOBJECTS

public static final byte **CONTENTTYPE_SHAREDOBJECTS**

AMF Content type: shared object packet (AMF0)
Constant value: **19**

CONTENTTYPE_SHAREDOBJECTS0

public static final byte **CONTENTTYPE_SHAREDOBJECTS0**

AMF Content type: shared object packet (AMF0)
Constant value: **19**

CONTENTTYPE_SHAREDOBJECTS3

public static final byte **CONTENTTYPE_SHAREDOBJECTS3**

AMF Content type: shared object packet (AMF3)
Constant value: **16**

CONTENTTYPE_FUNCTION

public static final byte **CONTENTTYPE_FUNCTION**

AMF Content type: function data (AMF0)
Constant value: **20**

CONTENTTYPE_FUNCTION0

public static final byte **CONTENTTYPE_FUNCTION0**

AMF Content type: function data (AMF0)
Constant value: **20**

CONTENTTYPE_MEDIACHUNK

public static final byte **CONTENTTYPE_MEDIACHUNK**

AMF Content type: media chunk
Constant value: **22**

CONTENTTYPE_FUNCTION3

public static final byte **CONTENTTYPE_FUNCTION3**

AMF Content type: function data (AMF3)
Constant value: **17**

(continued from last page)

CONTENTTYPE_PLAYCALLBACK

```
public static final byte CONTENTTYPE_PLAYCALLBACK
```

AMF Content type: play callback
Constant value: **127**

ACCEPTORS_ACCEPTOR

```
public static final java.lang.String ACCEPTORS_ACCEPTOR
```

Acceptor types: acceptor
Constant value: **acceptor**

ACCEPTORS_HANDLERADAPTER

```
public static final java.lang.String ACCEPTORS_HANDLERADAPTER
```

Acceptor types: handler
Constant value: **handlerAdapter**

Methods

init

```
public void init(String basePath)
```

Initialize vHost.

Parameters:

basePath - base path to configuration files

shutdown

```
public void shutdown()
```

Shutdown.

getStreamTypes

```
public StreamList getStreamTypes()
```

Get default stream type.

Returns:

default stream type

getProperty

```
public String getProperty(String key)
```

Get virtual host property.

Parameters:

key - key

Returns:

property value or null if does not exist

getClientTimeout

```
public int getClientTimeout()
```

Get client timeout.

Returns:

client timeout

getNextNetConnectionId

```
public int getNextNetConnectionId()
```

Get next connection id.

Returns:

next connection id

getClient

```
public IClient getClient(int clientId)
```

Get client by client id.

Parameters:

clientId - client id

Returns:

client

getClient

```
public IClient getClient(int clientId,  
                        boolean create)
```

Get client by client id and create if does not exist.

Parameters:

clientId - client id

create - create if does not exist

Returns:

client

removeClient

```
public void removeClient(int clientId)
```

Remove client from vHost.

Parameters:

clientId - client id

killClient

```
public void killClient(int clientId)
```

Remove client from vHost and send disconnect message.

(continued from last page)

Parameters:

clientId - client id

killRTSPSession

```
public void killRTSPSession(String rtspSessionId)
```

Kill an RTSP connection by the RTSP session id

Parameters:

rtspSessionId - RTSP session id

getApplication

```
public IApplication getApplication(String applicationName)
```

Get application by name.

Parameters:

applicationName - application name

Returns:

application

isApplicationLoaded

```
public boolean isApplicationLoaded(String applicationName)
```

Return true is the application is loaded

Parameters:

applicationName - application name

Returns:

true if application is loaded

getHomePath

```
public String getHomePath()
```

Get vHost configuration path.

Returns:

configuration path

getProperties

```
public WMSProperties getProperties()
```

Get properties attached to this vHost.

Returns:

properties attached to this vHost

getManagerProperties

```
public WMSProperties getManagerProperties()
```

(continued from last page)

Get VHost's manager's properties collection

Returns:

manager's property collection

getCoreTransportPoolSize

```
public int getCoreTransportPoolSize()
```

Get the transport core thread pool size.

Returns:

default core thread pool size

setCoreTransportPoolSize

```
public void setCoreTransportPoolSize(int corePoolSize)
```

Set the transport core thread pool size.

Parameters:

corePoolSize - core thread pool size

getCoreHandlerPoolSize

```
public int getCoreHandlerPoolSize()
```

Get the handler core thread pool size.

Returns:

default core thread pool size

setCoreHandlerPoolSize

```
public void setCoreHandlerPoolSize(int corePoolSize)
```

Set the handler core thread pool size.

Parameters:

corePoolSize - core thread pool size

getFileIOPoolSize

```
public int getFileIOPoolSize()
```

Get the default file io pool size.

Returns:

default file io pool size

setFileIOPoolSize

```
public void setFileIOPoolSize(int fileIOPoolSize)
```

Set default file io thread pool size.

Parameters:

fileIOPoolSize - default file io thread pool size

setClientTimeout

```
public void setClientTimeout(int clientTimeout)
```

Set client timeout. An inactive client connected by RTMPT protocol will be deleted after this timeout.

Parameters:

clientTimeout - client timeout

getHostPortsList

```
public HostPortList getHostPortsList()
```

Get list of host port definitions for vHost.

Returns:

list of host port definitions for vHost

getThreadPool

```
public ThreadPool getThreadPool()
```

Get the VHost handler thread pool. Same as getHandlerThreadPool.

Returns:

VHost handler thread pool

getTransportThreadPool

```
public ThreadPool getTransportThreadPool()
```

Get the VHost transport thread pool. This thread pool is used to read/write data from the transports sockets.

Returns:

VHost transport thread pool

getHandlerThreadPool

```
public ThreadPool getHandlerThreadPool()
```

Get the VHost handler thread pool. This thread pool is used to process the incoming events.

Returns:

VHost handler thread pool

getName

```
public String getName()
```

Get vHost name

Returns:

vHost name

setAdminInterfaceHostPort

```
public void setAdminInterfaceHostPort(HostPort adminInterfaceHostPort)
```

(continued from last page)

Set admin interface host port (not used)

Parameters:

adminInterfaceHostPort - admin interface host port

getClientCount

```
public int getClientCount()
```

Get number of clients connected to this vHost.

Returns:

number of clients connected to this vHost

getNetConnectionProcessorCount

```
public int getNetConnectionProcessorCount()
```

Get net connection processor count. Number of threads used for server to server communication (not finished).

Returns:

net connection processor count

setNetConnectionProcessorCount

```
public void setNetConnectionProcessorCount(int netConnectionProcessorCount)
```

Set net connection processor count. Number of threads used for server to server communication (not finished).

Parameters:

netConnectionProcessorCount - net connection processor count

addApplicationListener

```
public void addApplicationListener(IApplicationNotify applicationListener)
```

Add application listener. An application listener will receive the following events: onApplicationCreate, onApplicationDestroy.

Parameters:

applicationListener - application listener

removeApplicationListener

```
public void removeApplicationListener(IApplicationNotify applicationListener)
```

Remove application listener.

Parameters:

applicationListener - application listener

addAcceptorListener

```
public void addAcceptorListener(IAcceptorNotify acceptorListener)
```

Add acceptor listener. Acceptor listeners will receive the following events: onAcceptorCreate, onAcceptorDestroy.

Parameters:

acceptorListener - acceptor listener

removeAcceptorListener

```
public void removeAcceptorListener(IAcceptorNotify acceptorListener)
```

Remove acceptor listener.

Parameters:

acceptorListener - acceptor listener

getIOScheduler

```
public IOScheduler getIOScheduler()
```

Get IO scheduler for vHost. IO scheduler is used to schedule reads from the disk to increase server throughput for static flv serving.

Returns:

IO scheduler for vHost

getIoPerformanceCounter

```
public IoPerformanceCounter getIoPerformanceCounter()
```

Get vHost IO performance counter.

Returns:

io performance counter

getIoPerformanceCounter

```
public IoPerformanceCounter getIoPerformanceCounter(int counterIndex)
```

Get vHost IO performance counter for a specific technology (see IVHost.COUNTER_*)

Parameters:

counterIndex - counter index (see IVHost.COUNTER_*)

Returns:

io performance counter

getConnectionCounter

```
public ConnectionCounter getConnectionCounter()
```

Get vHost connection counter.

Returns:

connection counter

getConnectionCounter

```
public ConnectionCounterSimple getConnectionCounter(int counterIndex)
```

Get vHost connection counter for a specific technology (see IVHost.COUNTER_*)

Parameters:

counterIndex - counter index (see IVHost.COUNTER_*)

(continued from last page)

Returns:

connection counter

getDateStarted

```
public String getDateStarted()
```

Get date and time the server was started.

Returns:

date and time the server was started

getTimeRunning

```
public String getTimeRunning()
```

Get the time vHost has been running.

Returns:

formatted string with vHost uptime

getTimeRunningSeconds

```
public double getTimeRunningSeconds()
```

Get time running in seconds

Returns:

time running in seconds

getConnectionLimit

```
public int getConnectionLimit()
```

Get vHost connection limit.

Returns:

vHost connection limit

getMediaReaders

```
public MediaReaderList getMediaReaders()
```

Get the media readers attached to vHost (MediaReaders.xml).

Returns:

media readers attached to vHost

getMediaWriters

```
public MediaWriterList getMediaWriters()
```

Get the media writers attached to vHost (MediaWriters.xml).

Returns:

media writers attached to vHost

(continued from last page)

getMp3TagMap

```
public java.util.Map getMp3TagMap()
```

Get MP3 tag map attached to vHost (MP3Tags.xml).

Returns:

MP3 tag map attached to vHost

getCEACaptionConversionContext

```
public CEACaptionConversionContext getCEACaptionConversionContext()
```

Get CEACaptionConversionContextMP3 tag map attached to vHost (CEACaptionConversion.xml).

Returns:

MP3 tag map attached to vHost

updateLoggingDuration

```
public void updateLoggingDuration()
```

Internal: update the internal logging values.

getApplicationTimeout

```
public int getApplicationTimeout()
```

Get application time out (milliseconds). Time from last client disconnect to application destruction.

Returns:

application time out (milliseconds)

setApplicationTimeout

```
public void setApplicationTimeout(int applicationTimeout)
```

Set application time out (milliseconds).

Parameters:

applicationTimeout - application time out (milliseconds)

getPingTimeout

```
public int getPingTimeout()
```

Get ping timeout (milliseconds)

Returns:

ping timeout (milliseconds)

setPingTimeout

```
public void setPingTimeout(int pingTimeout)
```

Set ping timeout (milliseconds)

Parameters:

(continued from last page)

pingTimeout - ping timeout (milliseconds)

getValidationFrequency

```
public int getValidationFrequency()
```

Get time between validation pings (milliseconds)

Returns:

time between validation pings (milliseconds)

setValidationFrequency

```
public void setValidationFrequency(int validationFrequency)
```

Set time between validation pings (milliseconds)

Parameters:

validationFrequency - time between validation pings (milliseconds)

getMaximumPendingWriteBytes

```
public int getMaximumPendingWriteBytes()
```

Get maximum number a bytes a client connection can have waiting to be sent before the connection is terminated. If set to zero this feature is turned off.

Returns:

maximum number a bytes a client connection can have waiting to be sent before the connection is terminated

setMaximumPendingWriteBytes

```
public void setMaximumPendingWriteBytes(int maximumPendingWriteBytes)
```

Set maximum number a bytes a client connection can have waiting to be sent before the connection is terminated. If set to zero this feature is turned off.

Parameters:

maximumPendingWriteBytes - maximum number a bytes a client connection can have waiting to be sent before the connection is terminated

getMaximumPendingReadBytes

```
public int getMaximumPendingReadBytes()
```

Set maximum number of bytes a client connection can have waiting to be written before the connection is terminated. If set to zero this feature is off.

Returns:

maximum number of bytes a client connection can have waiting to be written before the connection is terminated

setMaximumPendingReadBytes

```
public void setMaximumPendingReadBytes(int maximumPendingReaderBytes)
```

Get maximum number of bytes a client connection can have waiting to be written before the connection is terminated. If set to zero this feature is off.

Parameters:

(continued from last page)

`maximumPendingReaderBytes` - maximum number of bytes a client connection can have waiting to be written before the connection is terminated

getMaximumSetBufferTime

```
public int getMaximumSetBufferTime()
```

Get maximum number of milliseconds allowed for the `NetStream.setBufferTime(secs)` call. If set to zero this feature is turned off.

Returns:

maximum number of milliseconds allowed for the `NetStream.setBufferTime(secs)` call

setMaximumSetBufferTime

```
public void setMaximumSetBufferTime(int maximumSetBufferTime)
```

Set maximum number of milliseconds allowed for the `NetStream.setBufferTime(secs)` call. If set to zero this feature is turned off.

Parameters:

`maximumSetBufferTime` - maximum number of milliseconds allowed for the `NetStream.setBufferTime(secs)` call

getApplicationLock

```
public edu.emory.mathcs.backport.java.util.concurrent.locks.WMSReadWriteLock  
getApplicationLock()
```

Get the object used in synchronized statements to lock and application.

Returns:

object used in synchronized statements to lock and application

getClientIdleFrequency

```
public int getClientIdleFrequency()
```

Get default client idle frequency (milliseconds)

Returns:

default client idle frequency (milliseconds)

setClientIdleFrequency

```
public void setClientIdleFrequency(int clientIdleFrequency)
```

Set default client idle frequency (milliseconds)

Parameters:

`clientIdleFrequency` - default client idle frequency (milliseconds)

getRTPIdeFrequency

```
public int getRTPIdeFrequency()
```

Get default RTP idle frequency (milliseconds)

Returns:

default RTP idle frequency (milliseconds)

setRTPIdleFrequency

```
public void setRTPIdleFrequency(int rtpIdleFrequency)
```

Set default RTP idle frequency (milliseconds)

Parameters:

rtpIdleFrequency - RTP default client idle frequency (milliseconds)

getNetConnectionIdleFrequency

```
public int getNetConnectionIdleFrequency()
```

Get server to server idle frequency (milliseconds)

Returns:

server to server idle frequency (milliseconds)

setNetConnectionIdleFrequency

```
public void setNetConnectionIdleFrequency(int netConnectionIdleFrequency)
```

Set the server to server idle frequency (milliseconds)

Parameters:

netConnectionIdleFrequency - server to server idle frequency (milliseconds)

getIdleCheckFrequency

```
public int getIdleCheckFrequency()
```

Get idle check frequency (milliseconds)

Returns:

idle check frequency (milliseconds)

setIdleCheckFrequency

```
public void setIdleCheckFrequency(int idleCheckFrequency)
```

Set idle check frequency (milliseconds)

Parameters:

idleCheckFrequency - idle check frequency (milliseconds)

getIdleWorkerCount

```
public int getIdleWorkerCount()
```

Get number of threads used to generate idle events

Returns:

number of threads used to generate idle events

setIdleWorkerCount

```
public void setIdleWorkerCount(int idleWorkerCount)
```

(continued from last page)

Set number of threads used to generate idle events

Parameters:

idleWorkerCount - number of threads used to generate idle events

getKeepAliveTimeout

```
public int getKeepAliveTimeout()
```

Get the RTMPT connection keep alive timeout

Returns:

RTMPT connection keep alive timeout

setKeepAliveTimeout

```
public void setKeepAliveTimeout(int keepAliveTimeout)
```

Set the RTMPT connection keep alive timeout

Parameters:

keepAliveTimeout - RTMPT connection keep alive timeout

addIdleWorkerListener

```
public void addIdleWorkerListener(IIIdleWorkerNotify idleWorkerListener)
```

Add idleWorker listener. An idleWorker listener will receive the following events: onIdleWorkerCreate, onIdleWorkerDestroy.

Parameters:

idleWorkerListener - idleWorker listener

removeIdleWorkerListener

```
public void removeIdleWorkerListener(IIIdleWorkerNotify idleWorkerListener)
```

Remove idleWorker listener

Parameters:

idleWorkerListener - idleWorker listener

getNetConnectionHostPortConfig

```
public HostPortConfig getNetConnectionHostPortConfig()
```

Get the socket configuration for server to server connections

Returns:

socket configuration for server to server connections

getMediaCasterSettings

```
public MediaCasterSettings getMediaCasterSettings()
```

Get the media caster settings

Returns:

media caster settings

getMediaCasterList

```
public MediaCasterList getMediaCasterList()
```

Get the list of media caster definitions (MediaCaster.xml)

Returns:

list of media caster definitions

getApplicationNames

```
public java.util.List getApplicationNames()
```

Get a list of application names

Returns:

list of application names

getApplicationFolderNames

```
public java.util.List getApplicationFolderNames()
```

Get a list of application folder names

Returns:

list of application folder names

applicationExists

```
public boolean applicationExists(String name)
```

Return true if an application folder exists for this application name

Parameters:

name - application name

Returns:

true if an application folder exists for this application name

getRTPContext

```
public RTPContext getRTPContext()
```

Get the RTP (real time protocol) virtual host context

Returns:

RTP (real time protocol) virtual host context

getHTTPStreamerContext

```
public HTTPStreamerContext getHTTPStreamerContext()
```

Get the HTTPStreamer (Cupertino Streaming and Silverlight Smooth Streaming) host context

Returns:

HTTPStreamer (Cupertino Streaming and Silverlight Smooth Streaming) host context

getRTPDatagramConfigIncoming

```
public com.wowza.wms.rtp.transport.RTPUDPDatagramConfig getRTPDatagramConfigIncoming()
```

Get the RTP Datagram Socket configuration

Returns:

RTP Datagram Socket configuration

getRTPDatagramConfigOutgoing

```
public com.wowza.wms.rtp.transport.RTPUDPDatagramConfig getRTPDatagramConfigOutgoing()
```

Get the RTP Datagram Socket configuration

Returns:

RTP Datagram Socket configuration

getAuthenticationList

```
public AuthenticationList getAuthenticationList()
```

Get the list of available authentication methods

Returns:

list of available authentication methods

getIdleMinimumWaitTime

```
public int getIdleMinimumWaitTime()
```

Get the minimum time (milliseconds) the idle worker thread will sleep before generating idle events

Returns:

minimum time (milliseconds) the idle worker thread will sleep before generating idle events

setIdleMinimumWaitTime

```
public void setIdleMinimumWaitTime(int idleMinimumWaitTime)
```

Set the minimum time (milliseconds) the idle worker thread will sleep before generating idle events

Parameters:

idleMinimumWaitTime - minimum time (milliseconds) the idle worker thread will sleep before generating idle events

getIdleWorkers

```
public IdleWorkersUtil getIdleWorkers()
```

Get the idle worker utility

Returns:

idle worker utility

isShuttingDown

```
public boolean isShuttingDown()
```

(continued from last page)

Is the VHost shutting down

Returns:

true if the vhost is shutting down

setShuttingDown

```
public void setShuttingDown(boolean shuttingDown)
```

Set VHost shutting down flag

Parameters:

shuttingDown - true if the vhost is shutting down

reparentClient

```
public void reparentClient(IClient client)
```

Move a client object to a new vhost. This can only be done right after the handshake process has completed. See `IVHostNotify.onVHostClientConnect`.

Parameters:

client - client object to move

getDynamicLogProperties

```
public java.util.Properties getDynamicLogProperties()
```

Get the dynamic log properties defined at the vhost level in `conf/log4j.properties`

Returns:

dynamic log properties defined at the vhost level

setDynamicLogProperties

```
public void setDynamicLogProperties(java.util.Properties dynamicLogProperties)
```

Set the dynamic log properties set at the vhost level

Parameters:

dynamicLogProperties - dynamic log properties defined at the vhost level

shutdownApplication

```
public void shutdownApplication(String appName)
```

Shutdown an application by name. This will disconnect all clients connected to all child application instances.

Parameters:

appName - application name

shutdownApplication

```
public void shutdownApplication(String appName,  
    boolean deleteFolder)
```

Shutdown an application by name. This will disconnect all clients connected to all child application instances.

(continued from last page)

Parameters:

appName - application name
deleteFolder - delete the application folder

unbindAllHostPorts

```
public void unbindAllHostPorts()
```

Unbind all HostPorts and drop all connections

suspendAllHostPorts

```
public void suspendAllHostPorts()
```

Suspend all HostPorts from accepting new connections. Current connections will continue to be serviced

closeHostPort

```
public void closeHostPort(HostPort hostPort,  
    boolean isSuspend)
```

Close an individual HostPort

Parameters:

hostPort - host port to close
isSuspend - if true will just suspend the HostPort from accepting new connections, if false will unbind and drop all connections

isSuspended

```
public boolean isSuspended()
```

Returns true is all HostPorts connected to this VHost are suspended

touchApplicationInstance

```
public boolean touchApplicationInstance(String appName)
```

Start an application instance if it is not already started then touch it so it stays loaded for at least 3 seconds.

Parameters:

appName - application name

Returns:

true is successful

touchApplicationInstance

```
public boolean touchApplicationInstance(String appName,  
    String appInstanceName)
```

Start an application instance if it is not already started then touch it so it stays loaded for at least 3 seconds. The default appInstanceName _definst_ will be used.

Parameters:

appName - application name
appInstanceName - app instance name

Returns:

(continued from last page)

true is successful

startApplicationInstance

```
public boolean startApplicationInstance(String appName)
```

Start an application instance. The default appInstanceName _definst_ will be used.

Parameters:

appName - application name

Returns:

true is successful

startApplicationInstance

```
public boolean startApplicationInstance(String appName,  
String appInstanceName)
```

Start an application instance.

Parameters:

appName - application name

appInstanceName - app instance name

Returns:

true is successful

getUDPTransportManager

```
public com.wowza.wms.rtp.transport.UDPTransportManager getUDPTransportManager()
```

Get the UDP transport manager.

Returns:

UDP transport manager

getHTTPStreamerList

```
public HTTPStreamerList getHTTPStreamerList()
```

Get the list of HTTPStreamers

Returns:

list of HTTPStreamers

getHTTPStreamerAdapter

```
public IHTTPStreamerAdapter getHTTPStreamerAdapter(String ID)
```

Get an HTTPStreamerAdapter by ID

Parameters:

ID - HTTPStreamerAdapter ID

Returns:

HTTPStreamerAdapter

putHTTPStreamerAdapter

```
public void putHTTPStreamerAdapter(String ID,  
    IHTTPStreamerAdapter adapter)
```

Add an HTTPStreamerAdapter

Parameters:

ID - HTTPStreamerAdapter ID
adapter - HTTPStreamerAdapter

getHTTPStreamerAdapterIDs

```
public java.util.List getHTTPStreamerAdapterIDs()
```

Get a list of HTTPStreamerAdapter IDs

Returns:

list of HTTPStreamerAdapter IDs

getLiveStreamPacketizerList

```
public LiveStreamPacketizerList getLiveStreamPacketizerList()
```

Get the LiveStreamPacketizerList

Returns:

LiveStreamPacketizerList

getLiveStreamTranscoderList

```
public LiveStreamTranscoderList getLiveStreamTranscoderList()
```

Get the LiveStreamTranscoderList

Returns:

LiveStreamTranscoderList

getTimedTextProviderConfig

```
public TimedTextProviderConfiguration getTimedTextProviderConfig()
```

Get the config of TimedTextProviders

Returns:

TimedTextProviders config

getStartupStreams

```
public java.util.List getStartupStreams()
```

Get the list of streams to start at virtual host startup

Returns:

list of streams to start at virtual host startup

(continued from last page)

addStartupStream

```
public void addStartupStream(StartupStream startupStream)
```

Add a stream to the list of streams to start and virtual host startup

Parameters:

startupStream - startup stream

isStartStarupStreams

```
public boolean isStartStarupStreams()
```

Returns true if the startup streams are to start and vhost startup

Returns:

true if the startup streams are to start and vhost startup

setStartStarupStreams

```
public void setStartStarupStreams(boolean startStarupStreams)
```

Set to true to startup startup stream as vhost startup

Parameters:

startStarupStreams - true if the startup streams are to start and vhost startup

startStartupStreams

```
public void startStartupStreams()
```

Method to start startup streams

stopStartupStreams

```
public void stopStartupStreams()
```

Method to stop startup streams

getDvrRecorderList

```
public DvrRecorderList getDvrRecorderList()
```

Get the DvrRecorderList

Returns:

list of DVR Recorders

getDvrStoreList

```
public DvrStoreList getDvrStoreList()
```

Get the list of DVR Stores

Returns:

list of DVR Stores

(continued from last page)

getLiveStreamRecordManager

```
public ILiveStreamRecordManager getLiveStreamRecordManager()
```

Returns the LiveStreamRecordManager associated with this vhost This object contains the active LiveStreamRecorders

Returns:

ILiveStreamRecordManager associated with this vhost

getMediaReaderContentType

```
public int getMediaReaderContentType(String mediaType)
```

Get the content type of a media stream name prefix (see IMediaReader.CONTENTTYPE_*)

Parameters:

mediaType - mediaType (such as flv or smil)

Returns:

content type (see IMediaReader.CONTENTTYPE_*)

getProtocolUsage

```
public void getProtocolUsage(boolean[] protocolsInUse)
```

Get the protocols in use by this application instance (see IApplicationInstance.PROTOCOLUSAGE_*)

getTranscoderVODContext

```
public com.wowza.wms.transcoder.vod.TranscoderVODContext getTranscoderVODContext()
```

Get the VOD transcoder context

Returns:

VOD transcoder context

getHTTPStreamerRequestValidator

```
public IVHostHTTPStreamerRequestValidator getHTTPStreamerRequestValidator()
```

Get the HTTP streamer request validator

Returns:

HTTP streamer request validator

setHTTPStreamerRequestValidator

```
public void setHTTPStreamerRequestValidator(IVHostHTTPStreamerRequestValidator  
httpRequestValidator)
```

Set the HTTP streamer request validator

Parameters:

httpRequestValidator - HTTP streamer request validator

getHTTPStreamerMaxPathLen

```
public int getHTTPStreamerMaxPathLen()
```

(continued from last page)

Get the max HTTP streamer path length

Returns:

max HTTP streamer path length

setHTTPStreamerMaxPathLen

```
public void setHTTPStreamerMaxPathLen(int httpStreamerMaxPathLen)
```

Set the max HTTP streamer path length

Parameters:

httpStreamerMaxPathLen - max HTTP streamer path length

getRTSPRequestValidator

```
public IVHostRTSPRequestValidator getRTSPRequestValidator()
```

Get the RTSP request validator

Returns:

RTSP request validator

setRTSPRequestValidator

```
public void setRTSPRequestValidator(IVHostRTSPRequestValidator rtspRequestValidator)
```

Set the RTSP request validator

Parameters:

rtspRequestValidator - RTSP request validator

getRTSPMaxPathLen

```
public int getRTSPMaxPathLen()
```

Get the max RTSP stream path length

Returns:

max RTSP stream path length

setRTSPMaxPathLen

```
public void setRTSPMaxPathLen(int rtspStreamerMaxPathLen)
```

Set the max RTSP stream path length

Parameters:

rtspStreamerMaxPathLen - max RTSP stream path length

getLiveStreamTranscoderSessionCount

```
public long getLiveStreamTranscoderSessionCount()
```

Get the current number of live stream transcoder sessions.

Returns:

current number of live stream transcoder sessions

getPushPublishContext

```
public PushPublishContext getPushPublishContext()
```

Get the push publish context

Returns:

push publish context

getUnidentifiedSessionTimeout

```
public int getUnidentifiedSessionTimeout()
```

Get the unidentified session timeout (TCP session that we are not able to identify)

Returns:

timeout in milliseconds

setUnidentifiedSessionTimeout

```
public void setUnidentifiedSessionTimeout(int unidentifiedSessionTimeout)
```

Set the unidentified session timeout (TCP session that we are not able to identify)

Parameters:

unidentifiedSessionTimeout - timeout in milliseconds

getStartupStreamsDelayTime

```
public int getStartupStreamsDelayTime()
```

Get startup streams delay time (milliseconds). Time between items started in StartupStreams.xml.

Returns:

delay time

setStartupStreamsDelayTime

```
public void setStartupStreamsDelayTime(int startupStreamsDelayTime)
```

Set startup streams delay time (milliseconds). Time between items started in StartupStreams.xml.

Parameters:

startupStreamsDelayTime - delay time

com.wowza.wms.vhost

Interface IVHostHTTPStreamerRequestValidator

public interface IVHostHTTPStreamerRequestValidator
extends

Method Summary	
boolean	validateHTTPStreamerRequest (RtmpRequestMessage request, HostPort hostPort, String path)

Methods

validateHTTPStreamerRequest

```
public boolean validateHTTPStreamerRequest(RtmpRequestMessage request,
HostPort hostPort,
String path)
```

com.wowza.wms.vhost Interface IVHostItemNotify

public interface **IVHostItemNotify**
extends

IVHostItemNotify: listener interface used by VHostList addVHostItemListener

VHostList is the list vHost definitions in VHosts.xml not the vHosts themselves. With this listener interface you can monitor the vHost definition changes.

Method Summary

void	onVHostItemCreate (VHostItem vhostItem) Triggered when vHostItem created
void	onVHostItemDestroy (VHostItem vhostItem) Triggered when vHostItem destroyed
void	onVHostItemUpdate (VHostItem vhostItem) Triggered when vHostItem updated

Methods

onVHostItemCreate

public void **onVHostItemCreate**([VHostItem](#) vhostItem)

Triggered when vHostItem created

Parameters:

vhostItem - vhostItem

onVHostItemUpdate

public void **onVHostItemUpdate**([VHostItem](#) vhostItem)

Triggered when vHostItem updated

Parameters:

vhostItem - vhostItem

onVHostItemDestroy

public void **onVHostItemDestroy**([VHostItem](#) vhostItem)

Triggered when vHostItem destroyed

Parameters:

vhostItem - vhostItem

com.wowza.wms.vhost Interface IVHostNotify

public interface **IVHostNotify**
extends

IVHostNotify: listener interface used by VHostSingleton addVHostListener

Method Summary

void	onVHostClientConnect (IVHost vhost, IClient inClient, com.wowza.wms.request.RequestFunction function, AMFDataList params) Triggered before a client connects to this virtual host.
void	onVHostCreate (IVHost vhost) Triggered when vHost created
void	onVHostInit (IVHost vhost) Triggered when vHost initialized
void	onVHostShutdownComplete (IVHost vhost) Triggered at the end of vhost shutdown
void	onVHostShutdownStart (IVHost vhost) Triggered at the beginning of vhost shutdown

Methods

onVHostCreate

public void **onVHostCreate**([IVHost](#) vhost)

Triggered when vHost created

Parameters:

vhost

onVHostInit

public void **onVHostInit**([IVHost](#) vhost)

Triggered when vHost initialized

Parameters:

vhost - vhost

onVHostShutdownStart

public void **onVHostShutdownStart**([IVHost](#) vhost)

Triggered at the beginning of vhost shutdown

(continued from last page)

Parameters:vhost - vhost

onVHostShutdownComplete

```
public void onVHostShutdownComplete(IVHost vhost)
```

Triggered at the end of vhost shutdown

Parameters:vhost - vhost

onVHostClientConnect

```
public void onVHostClientConnect(IVHost vhost,  
    IClient inClient,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Triggered before a client connects to this virtual host. Provides an opportunity to rewrite the information that is being used to connect.

Parameters:

vhost
inClient - client object of the connection
function - function
params - parameters

com.wowza.wms.vhost

Interface IVHostRTSPRequestValidator

public interface IVHostRTSPRequestValidator
extends

Method Summary	
boolean	validateRTSPRequest (com.wowza.wms.rtsp.RTSPRequestMessage req)

Methods

validateRTSPRequest

public boolean **validateRTSPRequest**(com.wowza.wms.rtsp.RTSPRequestMessage req)

com.wowza.wms.vhost Interface IWorkerThreadClear

public interface **IWorkerThreadClear**
extends

IWorkerThreadClear: Internal use.

Method Summary

void	clear()
void	reset()

Methods

reset

public void **reset**()

clear

public void **clear**()

com.wowza.wms.vhost Class StreamItem

java.lang.Object

└─com.wowza.wms.vhost.StreamItem

public class **StreamItem**
extends Object

StreamItem: data object that defines a streamType.

Constructor Summary

public	StreamItem (String name, String baseClass, String playClass) Create a new streamItem
--------	---

Method Summary

void	clearProperty (String name) Clear property.
String	getBaseClass () Get base class path.
String	getDescription () Get streamType description.
String	getName () Get streamType name.
String	getPlayClass () Get play class path.
WMSProperties	getProperties () Get properties.
String	getProperty (String name) Get property by name.
void	setBaseClass (String baseClass) Set base class path.
void	setDescription (String description) Set streamType description.
void	setName (String name) Set streamType name.
void	setPlayClass (String playClass) Set play class path.
void	setProperty (String name, String value) Set property value.

String	toString()
--------	----------------------------

Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Constructors

StreamItem

```
public StreamItem(String name,  
                  String baseClass,  
                  String playClass)
```

Create a new streamItem

Parameters:

`name` - streamType name
`baseClass` - base class path
`playClass` - play class path

Methods

getBaseClass

```
public String getBaseClass()
```

Get base class path.

Returns:

base clas path

setBaseClass

```
public void setBaseClass(String baseClass)
```

Set base class path.

Parameters:

`baseClass` - base class path

getName

```
public String getName()
```

Get streamType name.

Returns:

streamType name

setName

```
public void setName(String name)
```

(continued from last page)

Set streamType name.

Parameters:

name - streamType name

getPlayClass

```
public String getPlayClass()
```

Get play class path.

Returns:

play class path

setPlayClass

```
public void setPlayClass(String playClass)
```

Set play class path.

Parameters:

playClass - play class path

setProperty

```
public void setProperty(String name,  
                        String value)
```

Set property value.

Parameters:

name - property name

value - property value

clearProperty

```
public void clearProperty(String name)
```

Clear property.

Parameters:

name - property name

getProperty

```
public String getProperty(String name)
```

Get property by name.

Parameters:

name - property name

Returns:

property value

getProperties

```
public WMSProperties getProperties()
```

(continued from last page)

Get properties.

Returns:

properties

getDescription

```
public String getDescription()
```

Get streamType description.

Returns:

streamType description

setDescription

```
public void setDescription(String description)
```

Set streamType description.

Parameters:

description - streamType description

toString

```
public String toString()
```


com.wowza.wms.vhost Class StreamList

java.lang.Object
└─com.wowza.wms.vhost.StreamList

public class **StreamList**
extends Object

StreamList: collection of StreamItems

Constructor Summary

public	StreamList() Create empty StreamList
--------	---

Method Summary

StreamItem	getStreamDef(String name) Get streamItem by streamType name.
java.util.Map	getStreamDefs() Get Map of streamItems (by streamType names).
java.util.List	getStreamTypeNames() Get list of streamType names.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

StreamList

public **StreamList()**

Create empty StreamList

Methods

getStreamDefs

public java.util.Map **getStreamDefs()**

Get Map of streamItems (by streamType names).

Returns:
Map of streamItems

getStreamTypeNames

```
public java.util.List getStreamTypeNames()
```

Get list of streamType names. Returns shallow copy of list.

Returns:

list of streamType names

getStreamDef

```
public StreamItem getStreamDef(String name)
```

Get streamItem by streamType name.

Parameters:

name - streamType name

Returns:

streamItem

com.wowza.wms.vhost Class ThreadPool

java.lang.Object

└─com.wowza.wms.vhost.ThreadPool

public class **ThreadPool**
extends Object

ThreadPool: class for managing a pool of threads.

Constructor Summary

public	ThreadPool (IVHost vhost, String name) Create a new thread pool attached to a vHost.
--------	--

Method Summary

void	execute (Runnable command) Execute a runnable object.
int	getActiveCount () Get number of active threads.
java.util.concurrent. Executor	getExecutor () Get the underlying Executor pool.
int	getQueueSize () Get the number of command objects in the LinkedBlockingQueue.
void	init (int corePoolSize) Initialize threadPool.
void	terminate () Terminate all threads and cleanup threadPool.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

ThreadPool

public **ThreadPool**([IVHost](#) vhost,
[String](#) name)

Create a new thread pool attached to a vHost.

Parameters:

(continued from last page)

vhost - vHost

Methods

init

```
public void init(int corePoolSize)
```

Initialize threadPool.

Parameters:

corePoolSize - core pool size

terminate

```
public void terminate()
```

Terminate all threads and cleanup threadPool.

execute

```
public void execute(Runnable command)
```

Execute a runnable object. If the threadPool is at the corePool size, the object will be added to a LinkedBlockingQueue in the order it was received.

Parameters:

command

getExecutor

```
public java.util.concurrent.Executor getExecutor()
```

Get the underlying Executor pool.

Returns:

Executor pool

getActiveCount

```
public int getActiveCount()
```

Get number of active threads.

Returns:

number of active threads

getQueueSize

```
public int getQueueSize()
```

Get the number of command objects in the LinkedBlockingQueue.

Returns:

number of command objects in the LinkedBlockingQueue

com.wowza.wms.vhost Class VHostItem

java.lang.Object

└─com.wowza.wms.vhost.VHostItem

public class **VHostItem**
extends Object

VHostItem: data class that hold definition of a virtual host.

Constructor Summary

public	VHostItem() Create empty vHostItem
--------	---

Method Summary

String	getConfigDir() Get configuration path for vHost.
int	getConnectionLimit() Get connection limit of this vHost item.
String	getName() Get vHost name
WMSProperties	getProperties() Get properties
boolean	isVisited() Has this vHostItem been visited during load of VHosts.xml file.
void	reset() Reset vHostItem to empty state
void	setConfigDir(String configDir) Set configuration path for vHost.
void	setConnectionLimit(int connectionLimit) Set connection limit of this vHost item.
void	setName(String name) Set vHost name
void	setProperties(WMSProperties properties) Set properties
void	setVisited(boolean visited) Set is visited.
String	toString() Format vHost item

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait,`
`wait`

Constructors

VHostItem

```
public VHostItem()
```

Create empty vHostItem

Methods

reset

```
public void reset()
```

Reset vHostItem to empty state

getConfigDir

```
public String getConfigDir()
```

Get configuration path for vHost.

Returns:

configuration path for vHost

setConfigDir

```
public void setConfigDir(String configDir)
```

Set configuration path for vHost.

Parameters:

configDir - configuration path for vHost

getName

```
public String getName()
```

Get vHost name

Returns:

vHost name

setName

```
public void setName(String name)
```

Set vHost name

Parameters:

(continued from last page)

name - vHost name

getProperties

```
public WMSProperties getProperties()
```

Get properties

Returns:

properties

setProperties

```
public void setProperties(WMSProperties properties)
```

Set properties

Parameters:

properties - properties

isVisited

```
public boolean isVisited()
```

Has this vHostItem been visited during load of VHosts.xml file. Internally used to track and delete vHost definitions on reload of VHosts.xml.

Returns:

Has this vHostItem been visited during load

setVisited

```
public void setVisited(boolean visited)
```

Set is visited.

Parameters:

visited - is visited

getConnectionLimit

```
public int getConnectionLimit()
```

Get connection limit of this vHost item.

Returns:

connection limit of this vHost item

setConnectionLimit

```
public void setConnectionLimit(int connectionLimit)
```

Set connection limit of this vHost item.

Parameters:

connectionLimit - connection limit of this vHost item

(continued from last page)

toString

```
public String toString()
```

Format vHost item

com.wowza.wms.vhost Class VHostList

java.lang.Object

└─com.wowza.wms.vhost.VHostList

public class **VHostList**
extends Object

VHostList: list of VHost items. Result of parsing VHosts.xml at server startup or VHosts.xml reload. This interface can keep track of reloads and carefully mark items for deletion.

Constructor Summary

public	<u>VHostList()</u> Create empty vHostList
--------	--

Method Summary

void	<u>addVHostItemListener()</u> (<u>IVHostItemNotify</u> vHostItemListener) Add vHostItem listener.
java.util.List	<u>getVHostItems()</u> Get a list of vHostItems.
java.util.Map	<u>getVHostMap()</u> Get the Map of vHostItem definitions.
java.util.List	<u>getVHostNames()</u> Get a list of vHost names.
void	<u>loadConfig()</u>
void	<u>notifyVHostItemCreate()</u> (<u>VHostItem</u> vhostItem) Notify vHostItem listener of item create.
void	<u>notifyVHostItemDestroy()</u> (<u>VHostItem</u> vhostItem) Notify vHostItem listener of item destroy.
void	<u>notifyVHostItemUpdate()</u> (<u>VHostItem</u> vhostItem) Notify vHostItem listener of item update.
void	<u>reloadConfig()</u> Reload VHosts.xml file.
void	<u>removeVHostItemListener()</u> (<u>IVHostItemNotify</u> vHostItemListener) Remove vHostItem listener.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

VHostList

```
public VHostList()
```

Create empty vHostList

Methods

loadConfig

```
public void loadConfig()
```

reloadConfig

```
public void reloadConfig()
```

Reload VHosts.xml file.

getVHostMap

```
public java.util.Map getVHostMap()
```

Get the Map of vHostItem defintions.

Returns:

Map of vHostItem defintions

addVHostItemListener

```
public void addVHostItemListener(IVHostItemNotify vHostItemListener)
```

Add vHostItem listener. vHostItem listeners will receive the following events: onVHostItemCreate, onVHostItemUpdate, onVHostItemDestroy.

Parameters:

vHostItemListener - vHostItem listener

removeVHostItemListener

```
public void removeVHostItemListener(IVHostItemNotify vHostItemListener)
```

Remove vHostItem listener.

Parameters:

vHostItemListener - vHostItem listener

notifyVHostItemCreate

```
public void notifyVHostItemCreate(VHostItem vhostItem)
```

Notify vHostItem listener of item create.

(continued from last page)

Parameters:

vhostItem - vHostItem

notifyVHostItemUpdate

```
public void notifyVHostItemUpdate(VHostItem vhostItem)
```

Notify vHostItem listener of item update.

Parameters:

vhostItem - vHostItem

notifyVHostItemDestroy

```
public void notifyVHostItemDestroy(VHostItem vhostItem)
```

Notify vHostItem listener of item destroy.

Parameters:

vhostItem - vHostItem

getVHostNames

```
public java.util.List getVHostNames()
```

Get a list of vHost names. Creates a copy of list.

Returns:

list of vHost names

getVHostItems

```
public java.util.List getVHostItems()
```

Get a list of vHostItems. Create a shallow copy of list.

Returns:

list of vHostItems

com.wowza.wms.vhost Class VHostSingleton

java.lang.Object

└─com.wowza.wms.vhost.VHostSingleton

public class **VHostSingleton**
extends Object

VHostSingleton: singleton that provide access to running vHosts.

Constructor Summary

public	VHostSingleton()
--------	----------------------------------

Method Summary

static void	addVHostListener (IVHostNotify vhostListener) Add a vHost listener.
static IVHost	getInstance (String vhostName) Get vHost by name.
static IVHost	getInstance (String vhostName, boolean doCreate) Get vHost by name.
static java.util.List	getVHostNames () Get a list of vHost names.
static void	init (String vhostName, String configHome) Initialize vhost by name.
static void	notifyVHostClientConnect (IVHost vhost, IClient inClient, com.wowza.wms.request.RequestFunction function, AMFDataList params) Notify vhost client connect
static void	notifyVHostCreate (IVHost vhost) Notify vHost listener of create.
static void	notifyVHostInit (IVHost vhost) Notify vHost listener of ini.
static void	notifyVHostShutdownComplete (IVHost vhost) Notify vHost listener of destruction.
static void	notifyVHostShutdownStart (IVHost vhost) Notify vHost listener of destruction.
static void	remove (String vhostName) Remove a vHost from list of running vHosts.
static void	removeApplicationListener (IVHostNotify vhostListener) Remove vHost listener

static void	setServer (IServer server) Set a reference to the current running server.
static void	shutdown (String vhostName) Shutdown a vHost by name.

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

VHostSingleton

```
public VHostSingleton()
```

Methods

getInstance

```
public static IVHost getInstance(String vhostName)
```

Get vHost by name. Do NOT create if does not exist.

Parameters:

vhostName - vHost name

Returns:

vHost

getInstance

```
public static IVHost getInstance(String vhostName,  
    boolean doCreate)
```

Get vHost by name. Create if does not exist.

Parameters:

vhostName - vhost name

doCreate - if true, create if does not exist

Returns:

vHost

init

```
public static void init(String vhostName,  
    String configHome)
```

Initialize vhost by name.

Parameters:

vhostName - vHost name

(continued from last page)

configHome - configuration path, path to VHost.xml

setServer

```
public static void setServer(IServer server)
```

Set a reference to the current running server.

Parameters:

server - server

shutdown

```
public static void shutdown(String vhostName)
```

Shutdown a vHost by name.

Parameters:

vhostName - vhost name

getVHostNames

```
public static java.util.List getVHostNames()
```

Get a list of vHost names. Return copy of list.

Returns:

vHost names

remove

```
public static void remove(String vhostName)
```

Remove a vHost from list of running vHosts.

Parameters:

vhostName - vhost name

addVHostListener

```
public static void addVHostListener(IVHostNotify vhostListener)
```

Add a vHost listener. A vHost listener receives the following events: onVHostCreate, onVHostInit, onVHostDestroy

Parameters:

vhostListener - vHost listener

removeApplicationListener

```
public static void removeApplicationListener(IVHostNotify vhostListener)
```

Remove vHost listener

Parameters:

vhostListener - vHost listener

(continued from last page)

notifyVHostClientConnect

```
public static void notifyVHostClientConnect(IVHost vhost,  
      IClient inClient,  
      com.wowza.wms.request.RequestFunction function,  
      AMFDataList params)
```

Notify vhost client connect

Parameters:

vhost - vHost
inClient - client
function - function
params - parameters

notifyVHostCreate

```
public static void notifyVHostCreate(IVHost vhost)
```

Notify vHost listener of create.

Parameters:

vhost - vHost

notifyVHostInit

```
public static void notifyVHostInit(IVHost vhost)
```

Notify vHost listener of ini.

Parameters:

vhost - vHost

notifyVHostShutdownStart

```
public static void notifyVHostShutdownStart(IVHost vhost)
```

Notify vHost listener of destruction.

Parameters:

vhost - vHost

notifyVHostShutdownComplete

```
public static void notifyVHostShutdownComplete(IVHost vhost)
```

Notify vHost listener of destruction.

Parameters:

vhost - vHost

Index

A

- ABSOLUTE_FORMAT 729
- ABSOLUTE_TIME_PATTERN 729
- acceptConnection 343, 360, 361
- ACCEPTORS_ACCEPTOR 1820
- ACCEPTORS_HANDLERADAPTER 1820
- acceptSession 635, 1302
- acquire 931, 938, 939, 940, 1420, 1444
- acquireAndRelease 931, 939
- acquireMediaCaster 945
- acquireSocketAddress 1270
- acquireUDPPortPair 1270, 1271
- actionListeners 1300
- activateOptions 738
- add 82, 143, 144, 145, 178, 179, 180, 1256, 1794
- addAcceptConnectionAttribute 375
- addAcceptorListener 1825
- addActionListener 1311
- addAdaptiveGroupItem 1059, 1101
- addApplicationInstanceListener 244
- addApplicationListener 1825
- addAudioCodecConfigPacket 1521
- addAudioData 1506, 1648, 1649
- addAudioDataInc 1648
- addAudioExtraData 1491
- addAuthorizedRESTAPI 1386, 1409
- addAuthQueryString 1123, 1191, 1198
- addBody 1253
- addBouncyCastleSecurityProvider 105
- addBytes 1253
- addCallbackListener 1529
- addChunk 211
- addClient 1420, 1444
- addClientListener 275, 1512, 1513
- addConnectionListener 342
- addConnectMetaData 1179
- addContentDescriptor 841
- addData 223
- addDataData 1506, 1649
- addDataDataInc 1650
- addDataEx 226
- addDataExtraData 1491
- addDifference 82
- addDRM 399
- addDvrChunkListener 551
- addDvrRecorderListener 313
- addDvrRecordingListener 540
- addDvrStoreListener 539
- addDvrStreamManagerListener 313
- addFirstABSTRequest 667
- addHTTPListener 316
- addHttpProvider 1791
- addHTTPSessionToTCPSession 647
- addHTTPStreamerAdapterID 1792
- addHTTPStreamerSession 301
- addIdleWorkerListener 1832
- addIOPerformance 630, 1308
- addIOPerformance2 630, 1308
- addItem 1635
- addLicense 1595
- addListener 684, 698, 1164, 1668
- addLivePacketListener 1530
- addLiveStreamPacketizerListener 309
- addLiveStreamTranscoderListener 310
- addManifestEntries 539, 550
- addMediaCasterListener 279
- addMediaReaderListener 306
- addMediaStreamListener 277, 1592
- addMediaWriterListener 304
- addMember 205
- addMetaDataListener 953
- addModuleListener 283
- addNameGroup 1593
- addObject 160
- addOutput 1113
- addOverlay 1756
- addOverlayImage 1755
- addPendingWrites 879
- addPlayReadyDecryptorKey 1573
- addPlayStreamByName 281
- addProperties 867
- addPublisher 298
- addPushPublishSession 318
- addPushPublishSessionListener 318
- addQueryStringRemoveList 610
- addRange 429
- addRendition 976

addRepeaterHeartBeatItem 542
addRTPIIncomingDatagramPortAll 301
addRTPIIncomingDatagramPortRange 301
addRTPLListener 316
addRTPSession 293
addRTSPStream 1303
addSegment 959
addServerListener 1379, 1403
addSession 1158, 1322
addSessionListener 1322
addSharedObjectListener 277, 1432
addSlotListener 1424, 1448
addSorterPacket 1763
addStartupStream 1838
addStreamAttribute 1342
addStreamDomainStr 638
addStreamDomainStrs 639
addStreamInfo 1342
addStreamSrcToMediaCaster 941
addString 159
addToCache 442
addToChunk 507, 652
addToEnvMap 868
addToManifest 494
addToPlaylist 1665, 1666
addTrack 1338
addTrackId 1340
addTrackInternal 1339
addTrait 160
addUserHTTPHeaders 642
addVHostItemListener 1862
addVHostListener 1866
addVideoCodecConfigPacket 1521
addVideoData 1505, 1646, 1647
addVideoDataInc 1648
addVideoExtraData 1491
addVideoH264SEILListener 1529
adjustFirstPacketTCs 68
AKAMAI_DEFAULT_HOSTID 1088
akamaiConfigName 1090
akamaiEventName 1089
akamaiHostId 1089
akamaiIngressURLPrefix 1089
akamaiOriginIp 1151
akamaiOriginPort 1151
akamaiPassword 1151
akamaiPlaybackURLPrefix 1089
akamaiProtocolName 1090
akamaiStreamId 1089
akamaiUserName 1151
ALGORITHMID_CBC 395
ALGORITHMID_CTR 395
ALGORITHMID_NONE 394
alphas 32
AMF3Utils 120
AMF_LEVEL0 131
AMF_LEVEL3 131
AMF_TEXTTYPE_ON_TEXT_DATA 1724
AMFData 132
AMFDataArray 141, 142
AMFDataByteArray 152, 153
AMFDataContextDeserialize 158
AMFDataContextSerialize 163
AMFDataItem 169, 170, 171
AMFDataList 177, 178
AMFDataMixedArray 189, 190
AMFDataObj 197, 198
AMFDataTrait 205
AMFDEBUGHEADERSIZE 210
AMFObj 210
AMFObjChunk 217
AMFPacket 221
AMFUtils 13
announce 1346
append 739
APPEND_FILE 708
appendParamsToUrl 110
appInstance 1043, 1073, 1298, 1585
appInstanceShutdown 1298
APPLICATION_KEY 728
applicationExists 1833
ARCHIVE_STRATEGY_APPEND 460
ARCHIVE_STRATEGY_DELETE 460
ARCHIVE_STRATEGY_VERSION 460
artifact 410, 432
assembleQueryStr 75
attachToWMSSession 1346
audioCodecId 966
audioCodecStringToId 60
audioCodecToMetaDataString 61

audioCodecToString 61
audioCodecTypeToString 95
audioHost 1233
audioPacketizers 1301
audioPort 1233
AUDIOSAMPLE_ACCESS_ALL 356
AUDIOSAMPLE_ACCESS_NONE 356
AUDIOSAMPLEACCESS 1489
authenticateHandler 563
authenticateHTTPProvider 333
authenticateHTTPProviderHandler 563
authenticatePlayHandler 1298
authenticatePublishHandler 1298
authenticateRTSP 334
authenticateSIP 335
AuthenticateUsernamePasswordProviderBase 329
authenticationMethod 563
AUTHMETHOD_PLAY 1297
AUTHMETHOD_PUBLISH 1297
AUTHMETHOD_UNKNOWN 1297
authMode 1190, 1197
AVSYNCMETHODS_RTPTIMECODE 1336
AVSYNCMETHODS_SENDERREPORT 1336
AVSYNCMETHODS_SYSTEMCLOCK 1336
AVSYNCMETHODS_UNKNOWN 1336
avSyncNameToId 1352

B

badStreams 1044
bind 1283
bindAddress 1154
bitrate 1108
bitrateAudio 966
bitrateTotal 966
bitrateVideo 966
bitrateVideoPeak 966
blockSize 878
booleanValue 172
broadcastPlayMessage 1591
broadcastGetObjectEncoding 1591
broadcastMsg 280
buffer 217
BufferUtils 32
bumpReconnectWaitTime 1169

byteArrayToFile 45
byteArrayToInt 35
byteArrayToIntSafeSync 856
byteArrayToLong 34
byteArrayToLongSafeSync 856
byteArrayToShort 36
byteArrayToString 33
bytesAvg 1155
byteStringLen 817
byteValue 172

C

cacheRTPStream 1271
calcBitrate 509
calcParity 1712
calcTotalPacketSize 226
calculateChunkGroupTime 441
call 364
CALLBACK_PARAM1 1007
CALLBACK_PARAM10 1008
CALLBACK_PARAM2 1007
CALLBACK_PARAM3 1007
CALLBACK_PARAM4 1007
CALLBACK_PARAM5 1007
CALLBACK_PARAM6 1007
CALLBACK_PARAM7 1007
CALLBACK_PARAM8 1008
CALLBACK_PARAM9 1008
callConnect 1160, 1226
canHandle 564, 570, 603, 886, 1369
canPlay 547
canRecord 537, 547
canRecordAudio 507, 1603
canRecordData 507, 1604
canRecordVideo 507, 1604
CAT_ALL 782
CAT_application 771
CAT_cupertino 771
CAT_dvrchunk 772
CAT_mpegdash 772
CAT_rtsp 771
CAT_sanjose 771
CAT_server 770
CAT_session 771

- CAT_smoothstreaming 771
- CAT_stream 771
- CAT_transcoder 772
- CAT_vhost 771
- CAT_webm 772
- CEA608 1743
- CEA608_EXTENDED_FR_SP_MAP 1711
- CEA608_EXTENDED_NA_MAP 1711
- CEA608_EXTENDED_PORT_GERMAN_DANISH_MAP 1712
- CEA608_MUSICAL_NOTE 1711
- CEA608_SOLID_BLOCK 1711
- cea608CharToCharUTF8 1712
- cea608ExtendedCharacterToUTF8 1713
- CencInfo 395
- challenge 1190
- CHANNEL1 1694
- CHANNEL2 1694
- charUTF8ToCAE608 1712
- checkAndSetPlayLogged 629
- checkParity 1712, 1713
- checkSendMetadata 1347
- chunk 1104
- CHUNK_FILENAME_FORMAT_AUDIO 518
- CHUNK_FILENAME_FORMAT_DATA 518
- CHUNK_FILENAME_FORMAT_METADATA 518
- CHUNK_FILENAME_FORMAT_VIDEO 518
- CHUNK_FOLDER_FORMAT 519
- CHUNKIDDATEOFFSET 1107
- chunkLists 1102
- chunks 1103
- CLASS 1072, 1082, 1088, 1097, 1112, 1148, 1186, 1190, 1197, 1215, 1225, 1233, 1242
- CLASSNAME 1072, 1083, 1088, 1097, 1112, 1148, 1186, 1190, 1197, 1215, 1225, 1233, 1242
- cleanUp 738
- clear 82, 792, 1257, 1425, 1443, 1519, 1848
- clearByteContainer 214
- clearFastPlaySettings 366, 1509
- clearIntData 159
- clearLoggingValues 585, 632, 659, 666, 673, 1311, 1518
- clearOutput 1113
- clearOverlay 1756
- clearProperty 919, 1851
- clearRTSPSessionExtraLines 1364
- clearSegments 960, 976
- clearStreamName 1588, 1589
- clearStreamSrcToMediaCaster 941
- clearTracks 1339
- client 329
- clone 82, 205, 222, 239, 489, 1457
- cloneProperties 323
- close 9, 29, 738, 862, 878, 884, 1420, 1449, 1461, 1468, 1508, 1555, 1622, 1630, 1650, 1655, 1659, 1667
- closeAndWait 1667
- ClosedCaptionCEA608Utils 390, 1712
- closeHostPort 1836
- closeQuietly 52
- closeStream 1027
- CODEC_AUDIO_AAC 1816
- CODEC_AUDIO_AC3 1815
- CODEC_AUDIO_G711_ALAW 1816
- CODEC_AUDIO_G711_MULAW 1816
- CODEC_AUDIO_MP3 1815
- CODEC_AUDIO_MP3_8 1817
- CODEC_AUDIO_NELLYMOSER 1816
- CODEC_AUDIO_NELLYMOSER_16MONO 1816
- CODEC_AUDIO_NELLYMOSER_8MONO 1816
- CODEC_AUDIO_PCM_BE 1815
- CODEC_AUDIO_PCM_LE 1815
- CODEC_AUDIO_PCM_SWF 1815
- CODEC_AUDIO_RESERVED 1816
- CODEC_AUDIO_SPEEX 1816
- CODEC_AUDIO_UNKNOWN 1815
- CODEC_AUDIO_VORBIS 1816
- CODEC_STREAM_MP2T 1817
- CODEC_STREAM_UNKNOWN 1817
- CODEC_TYPE 493
- CODEC_VIDEO_H263 1814
- CODEC_VIDEO_H264 1814
- CODEC_VIDEO_H265 1815
- CODEC_VIDEO_MPEG2 1815
- CODEC_VIDEO_MPEG4 1814
- CODEC_VIDEO_SCREEN 1814
- CODEC_VIDEO_SCREEN2 1814
- CODEC_VIDEO_SPARK 1813
- CODEC_VIDEO_UNKNOWN 1813
- CODEC_VIDEO_VP6 1814
- CODEC_VIDEO_VP6A 1814
- CODEC_VIDEO_VP8 1814

CODECSTR_FORMAT_VIDEO_POSTIOS4 95
CODECSTR_FORMAT_VIDEO_PREIOS4 95
CODECSTR_FORMAT_VIDEO_UNKNOWN 94
CODES_HREF 729
COL0_BLUE 1693
COL0_CYAN 1694
COL0_GREEN 1693
COL0_HIBYTE 1699
COL0_LOWBYTE 1700
COL0_MAGENTA 1694
COL0_RED 1694
COL0_WHITE 1693
COL0_YELLOW 1694
COL_0 1700
COL_12 1700
COL_16 1700
COL_20 1700
COL_24 1700
COL_28 1701
COL_4 1700
COL_5 1700
COLN_LOWBYTE 1701
compareTo 1782
compress 155
conce 1197
CONFIGURATOR_CLASS_KEY 728
configureSocketAcceptor 1791
connect 1053, 1160, 1234
connect_Out 1216
connected 1152
ConnectionCounter 342
connectionFlashVersion 1151
connectionHolder 1299
connectionPageURL 1151
connectionQueryStr 1150
connectionResetCount 1152
connectionSwfURL 1151
connectionTimeout 1098, 1152
connectLastAttempt 1152
connectLastSuccess 1152
connectMetaData 1155
containsDvrRecorder 297
containsExtraDataKey 228
containsHeader 579
containsHTTPStreamer 296
containsIndex 584, 658, 665, 672
containsKey 198, 230
containsLiveStreamPacketizer 297
containsLiveStreamTranscoder 311
containsProperty 1425, 1445
containsSlot 1426, 1445
containsStreamDomainStr 638
containsStreamNameParts 636
containsTime 559
CONTENTTYPE_ACKBANDWIDTH 1818
CONTENTTYPE_AUDIO 1818
CONTENTTYPE_BUFFERSIZE 1818
CONTENTTYPE_CHORD 839
CONTENTTYPE_DATA 1818
CONTENTTYPE_DATA0 1818
CONTENTTYPE_DATA3 1819
CONTENTTYPE_EVENTS 838
CONTENTTYPE_FUNCTION 1819
CONTENTTYPE_FUNCTION0 1819
CONTENTTYPE_FUNCTION3 1819
CONTENTTYPE_LYRICS 838
CONTENTTYPE_MEDIA 1465
CONTENTTYPE_MEDIACHUNK 1819
CONTENTTYPE_MEDIALIST 1465
CONTENTTYPE_MOVEMENT 838
CONTENTTYPE_OTHER 838
CONTENTTYPE_PLAYCALLBACK 1819
CONTENTTYPE_SETBANDWIDTH 1818
CONTENTTYPE_SETCRUNKSIZE 1817
CONTENTTYPE_SHAREDOBJECTS 1819
CONTENTTYPE_SHAREDOBJECTS0 1819
CONTENTTYPE_SHAREDOBJECTS3 1819
CONTENTTYPE_TRANSCRIPTION 838
CONTENTTYPE_TRIVIA 839
CONTENTTYPE_UNKNOWN 1817
CONTENTTYPE_URLIMAGES 839
CONTENTTYPE_URLWEBPAGES 839
CONTENTTYPE_VIDEO 1818
CONTENTTYPE_WATCHDOG 1818
CONTROLCODES_AOF 1695
CONTROLCODES_AON 1695
CONTROLCODES_BAO 1698
CONTROLCODES_BAS 1698
CONTROLCODES_BBO 1697
CONTROLCODES_BBS 1697

CONTROLCODES_BCO 1697
CONTROLCODES_BCS 1697
CONTROLCODES_BGO 1697
CONTROLCODES_BGS 1697
CONTROLCODES_BMO 1698
CONTROLCODES_BMS 1698
CONTROLCODES_BRO 1697
CONTROLCODES_BRS 1697
CONTROLCODES_BS 1695
CONTROLCODES_BT 1698
CONTROLCODES_BWO 1696
CONTROLCODES_BWS 1697
CONTROLCODES_BYO 1698
CONTROLCODES_BYS 1698
CONTROLCODES_CR 1696
CONTROLCODES_DER 1695
CONTROLCODES_EDM 1696
CONTROLCODES_ENM 1696
CONTROLCODES_EOC 1696
CONTROLCODES_FA 1698
CONTROLCODES_FAU 1698
CONTROLCODES_FON 1695
CONTROLCODES_RCL 1694
CONTROLCODES_RDC 1695
CONTROLCODES_RTD 1696
CONTROLCODES_RU2 1695
CONTROLCODES_RU3 1695
CONTROLCODES_RU4 1695
CONTROLCODES_TO1 1696
CONTROLCODES_TO2 1696
CONTROLCODES_TO3 1696
CONTROLCODES_TR 1696
convertDataEvenToAMFData 1175
convertOnMetadataToSetDataFrame 1175
convertParams 13
COOKIEDATEFORMAT 74
cookieStr 1298
copyFile 51
copyFile2 51
COUNT_COLS 1699
COUNT_ROWS 1699
COUNTER_HTTPCUPERTINO 1813
COUNTER_HTTPDVRCHUNKS 1813
COUNTER_HTTPMPEGDASH 1813
COUNTER_HTTPSANJOSE 1813
COUNTER_HTTPSMOOTH 1813
COUNTER_HTTPWEBM 1813
COUNTER_RTMP 1812
COUNTER_RTP 1812
COUNTER_TOTAL 1813
createAkamaiStreamName 1188
createBroadcastMessage 1252
createConnectMessage 1252
createContextDeserialize 135
createContextSerialize 135
createDefaultMessage 1252
createDvrAudioChunk 504
createDvrDataChunk 505
createDvrOnMetadataChunk 505
createDvrVideoChunk 504
createEnhancedSeekMessage 1253
createGroupPlaylist 1059
createHTTPSessionInstance 600
createIndexLive 659, 666
createInstance 865, 1645, 1664
createOutputItem 1063
createPlayStatusMessage 1252
createPushPublishSession 1055, 1081, 1179, 1239
createRecorder 690
createSeekMessage 1252
createSOMessage 1253
createStream 1025, 1343, 1646
CrontabEvent 1779
CTRL_playlist_node 776
CUPERTINO_CHUNKLIST 614
CUPERTINO_MANIFEST 614
CUPERTINO_MEDIA 614
CUPERTINO_SUBTITLE_CHUNK 614
CUPERTINO_SUBTITLE_LIST 614
CURRENTFLASHVERSION 1148
CURRENTFMLEVERSION 1148

D

data 432
DATA_TYPE_AMF3 129
DATA_TYPE_AMF3_ARRAY 131
DATA_TYPE_AMF3_BOOLEAN_FALSE 130
DATA_TYPE_AMF3_BOOLEAN_TRUE 130
DATA_TYPE_AMF3_BYTEARRAY 131

- DATA_TYPE_AMF3_DATE 131
- DATA_TYPE_AMF3_INTEGER 130
- DATA_TYPE_AMF3_NULL 130
- DATA_TYPE_AMF3_NUMBER 130
- DATA_TYPE_AMF3_OBJECT 131
- DATA_TYPE_AMF3_STRING 130
- DATA_TYPE_AMF3_UNDEFINED 130
- DATA_TYPE_AMF3_XML_LEGACY 131
- DATA_TYPE_AMF3_XML_TOP 131
- DATA_TYPE_ARRAY 128
- DATA_TYPE_AS_OBJECT 129
- DATA_TYPE_BOOLEAN 127
- DATA_TYPE_BYTEARRAY 129
- DATA_TYPE_CUSTOM_CLASS 129
- DATA_TYPE_DATE 128
- DATA_TYPE_INTEGER 129
- DATA_TYPE_LONG_STRING 129
- DATA_TYPE_MIXED_ARRAY 128
- DATA_TYPE_MOVIE_CLIP 128
- DATA_TYPE_NULL 128
- DATA_TYPE_NUMBER 127
- DATA_TYPE_OBJECT 128
- DATA_TYPE_OBJECT_END 128
- DATA_TYPE_RECORDSET 129
- DATA_TYPE_REFERENCE_OBJECT 128
- DATA_TYPE_STRING 127
- DATA_TYPE_UNDEFINED 128
- DATA_TYPE_UNKNOWN 127
- DATA_TYPE_XML 129
- DATA_TYPE_XML_TOP 130
- DATE_AND_TIME_FORMAT 729
- DATE_AND_TIME_PATTERN 729
- DATEFORMAT 169
- dateValue 172
- DAY 1779
- debug 750
- debugDVRShutdown 1586
- debugLogLastPacketCount 1148
- debugPackets 1148
- debugRTSPSession 1301
- DebugUtils 42
- debugWOWZProtocol 1398
- DECODE 18
- decode 21
- DECODE_OBJ_REF 196
- DECODE_TRAITS 197
- DECODE_TRAITS_EXT 196
- DECODE_TRAITS_REF 196
- DECODE_UNDEFINED 196
- decodeFileToFile 23
- decodeFromFile 22
- decodeHexString 33
- decodeRangeHeader 1766
- decodeS 1407
- decodeSS 1407
- decodeStorageDir 296, 1400
- decodeStreamInfo 1769, 1770
- decodeString 23
- decodeToFile 22
- decodeToObject 21
- decodeValue 110
- decompress 156
- decPendingWriteCount 879
- decrement 344
- DEFAULT_AMF_CONVERTER_TEXT_TYPE 1734
- DEFAULT_AMF_CONVERTER_TRACK_INDEX 1735
- DEFAULT_APPINSTANCE_NAME 271
- DEFAULT_APPLICATION_NAME 242
- DEFAULT_CAPTION_CHARACTER_SET 1732
- DEFAULT_CHUNK_MEMORY_CACHESIZE 468
- DEFAULT_CONFIGURATION_FILE 728
- DEFAULT_CONFIGURATION_KEY 728
- DEFAULT_CUPERTINO_PLAYLIST_GZIP_THRESHOLD 481
- DEFAULT_LANGUAGE 809
- DEFAULT_MAXIMUM_CAPTION_DURATION 1731
- DEFAULT_PLAYLIST_REQUEST_DELEGATE 477
- DEFAULT_PORT 738
- DEFAULT_PROPERTY_ALLOWABLE_AV_PACKET_DELTA 464
- DEFAULT_PROPERTY_APPEND_DISCONTINUITY_DELTA 458
- DEFAULT_PROPERTY_AUDIO_GROUP_COUNT 517
- DEFAULT_PROPERTY_AUDIO_ONLY_CHUNK_TARGET_DURATION 466
- DEFAULT_PROPERTY_BREAK_ONPTS 466
- DEFAULT_PROPERTY_CAPTION_LANGUAGES 1726
- DEFAULT_PROPERTY_CAPTION_FILE_NAMING_RULE 1730
- DEFAULT_PROPERTY_CAPTION_PATH_NAMING_RULE

- E 1731
- DEFAULT_PROPERTY_CHUNK_CACHE_CLASS 462
- DEFAULT_PROPERTY_CHUNK_DURATION_MINIMUM 467
- DEFAULT_PROPERTY_CHUNK_GROUPING_SECONDS 458
- DEFAULT_PROPERTY_CHUNK_READER_CLASS 460
- DEFAULT_PROPERTY_CHUNK_WRITER_CLASS 461
- DEFAULT_PROPERTY_DEBUG_MAX_INVALID_CHUNKS_LOGGED 471
- DEFAULT_PROPERTY_DEBUG_MAX_RAW_PACKETS 473
- DEFAULT_PROPERTY_DEBUG_MAX_VALID_CHUNKS_LOGGED 472
- DEFAULT_PROPERTY_DEBUG_RAW_PACKETS 473
- DEFAULT_PROPERTY_DVR_MAX_CHUNK_LOG 474
- DEFAULT_PROPERTY_FILE_SYSTEM_CLASS 461
- DEFAULT_PROPERTY_MANIFEST_PERSISTENT_CLASS 461
- DEFAULT_PROPERTY_MANIFEST_PURGE_SIZE 514
- DEFAULT_PROPERTY_MAX_ALLOWABLE_CHUNK_DURATION 515
- DEFAULT_PROPERTY_MAX_RECALC_DURATION_LOG 474
- DEFAULT_PROPERTY_MAX_RECORDING_LENGTH 470
- DEFAULT_PROPERTY_MAX_SAVE_HOLDER_SIZE 516
- DEFAULT_PROPERTY_MBR_ALTERNATIVE_MATCH_DELTA 468
- DEFAULT_PROPERTY_MBR_MINIMUM_PACKET_TIME_GAP_SIZE 469
- DEFAULT_PROPERTY_MBR_MINIMUM_UTCTIME_GAP_SIZE 469
- DEFAULT_PROPERTY_MEDIACACHE_READER_CLASS 470
- DEFAULT_PROPERTY_MINIMUM_AVAILABLE_CHUNKS 514
- DEFAULT_PROPERTY_PACKET_DELTA_TO_NOTIFY 465
- DEFAULT_PROPERTY_PACKET_DELTA_TO_RESET_TIME 464
- DEFAULT_PROPERTY_PACKET_DURATION_MAXIMUM 469
- DEFAULT_PROPERTY_PACKET_SORT_TIME 466
- DEFAULT_PROPERTY_PURGE_CONTROL_CLASS 515
- DEFAULT_PROPERTY_RECALC_DURATION_MAX_AVERAGE_TRIGGER 516
- DEFAULT_PROPERTY_RECALC_DURATION_TRIGGER_SIZE 516
- DEFAULT_PROPERTY_RECORDINGS_LOADER_CLASS 462
- DEFAULT_PROPERTY_REPEATER_HEARTBEAT_DURATION 467
- DEFAULT_PROPERTY_SANJOSE_ABST_DURATION_TOLERANCE 480
- DEFAULT_PROPERTY_SANJOSE_ABST_TIMESCALE 480
- DEFAULT_PROPERTY_SMOOTH_MANIFEST_LIVE_CANN_PAUSE 482
- DEFAULT_PROPERTY_SMOOTH_MANIFEST_LIVE_CANN_SEEK 482
- DEFAULT_PROPERTY_SMOOTH_MANIFEST_MAJOR_VERSION 483
- DEFAULT_PROPERTY_SMOOTH_MANIFEST_MINOR_VERSION 483
- DEFAULT_PROPERTY_SMOOTH_MANIFEST_RECORDED_CANN_PAUSE 482
- DEFAULT_PROPERTY_SMOOTH_MANIFEST_RECORDED_CANN_SEEK 482
- DEFAULT_PROPERTY_SMOOTH_MANIFEST_RECORDED_SPECIFY_DURATION 483
- DEFAULT_PROPERTY_STORAGE_DIRECTORY 459
- DEFAULT_PROPERTY_STREAM_IDLE_TIMEOUT 517
- DEFAULT_PROPERTY_STREAM_STARTUP_TIMEOUT 518
- DEFAULT_PROPERTY_TEXT_READER_CLASS 463
- DEFAULT_PROPERTY_TEXT_WRITER_CLASS 463
- DEFAULT_PROPERTY_UTC_ALIGN_TO_AUDIO 513
- DEFAULT_PROPERTY_UTC_USE_WOWZA_PACKET_ARRIVAL 513
- DEFAULT_PROPERTY_WAIT_FOR_CODEC_TIME 517
- DEFAULT_RANDOMACCESSREADER 1465
- DEFAULT_RANDOMACCESSREADEROPTIMIZER 1465
- DEFAULT_REPOSITORY_NAME 727
- DEFAULT_RTPIGNOREPROFILELEVELID 1338
- DEFAULT_RTPIGNORESPPARAMETERSETS 1338
- DEFAULT_RTSPSESSIONNAME 1338
- DEFAULT_XML_CONFIGURATION_FILE 728
- DefaultDvrStreamVersionHandler 407
- defaultLocale 104

defaultTimeZone 104
delete 1106
deleteArchivedStore 538
deleteDirectory 52
deleteFolder 52
deleteSlot 1421, 1445
deleteStream 1026
describe 1356
deserialize 137, 149, 155, 172, 184, 190, 202, 238, 400, 404, 424, 427, 429, 503, 794
deserializeBody 799, 823, 826, 829, 832, 834, 840, 845, 848, 851, 854, 859
deserializeDate 120
deserializeFrame 819
deserializeGetSize 794
deserializeInnerObject 135
deserializeInt 120
deserializePacket 227
DeserializeResult 820
deserializeString 121, 818
destroy 1762
destroyInstance 865
detachFromWMSSession 1346
determineChunkGroupIdentifier 441
determineExistingStoreForPlaying 408, 554
determineExistingStoreForRecording 408, 553
determineStreamVersion 509
determineUtcTime 444
didIt 1115
directOutput 1773
disconnect 344, 1053, 1112, 1175, 1235, 1242, 1423, 1432, 1448
doBreak 1250
doCRC32 39
doesStackContainMethod 46
doHTTPAuthentication 565
doIdle 948, 1319
DONOTHING_BYTE 1694
DONOTHING_SHORT 1694
DONT_BREAK_LINES 18
doSendOnMetaData 1152
doSessionRedirect 642
doSet 83
doubleValue 171
doWatchdog 905, 930, 937, 1084, 1268, 1401, 1595
dstAppInstanceName 1149
dstApplicationName 1149
dstStreamName 1073, 1149
DT_FORMAT 1779
dummy 88
dump 1782
DVR_DEFAULT_FILESTORE 457
DVR_DEFAULT_RECORDER_ID 457
DVR_REPEATER_PACKETIZER_ID 457
DVR_STREAMING_PACKETIZER_ID 457
DVR_TIME 485
DVR_WINDOW_DURATION_UNLIMITED 459
DvrManifestChunkEntry 410
DvrManifestCodecEntry 414
DvrManifestEntry 421
DvrManifestEntryFactory 424
DvrManifestEntryRange 425
DvrManifestEntryRangeGroup 428
DvrManifestOnMetadataEntry 432
DvrManifestTimeMapEntry 435
dvrRecorders 1585
dvrStart 420
dvrStop 420
dvrStreamStorageDeleted 530
dvrStreamStorageLoaded 530
dvrStreamStoreCreate 529
dvrStreamStoreDestroy 529
dvrStreamStoreInit 529
dvrToPt 559
dvrToUtc 559

E

elapsedTime 1300
ElapsedTimer 48
ENCODE 17
encodeBytes 19, 20, 423
encodeFileToFile 23
encodeFromFile 22
encodeHexString 32
encodeObject 18, 19
encodeString 23
encodeToFile 22
encodeValue 110
ENCRYPTION_TYPE_CENC 488
ENCRYPTION_TYPE_CUPERTINO 487

ENCRYPTION_TYPE_PLAYREADY 487
encryptions 410
end 107
endChunk 507, 651
entries 1107
equals 102
error 747, 748
EVT_ALL 783
EVT_announce 775
EVT_app_start 775
EVT_app_stop 775
EVT_comment 775
EVT_connect 772
EVT_connect_burst 772
EVT_connect_pending 772
EVT_create 773
EVT_decoderaudiostart 775
EVT_decoderaudiostop 775
EVT_decodervideostart 775
EVT_decodervideostop 776
EVT_describe 775
EVT_destroy 773
EVT_disconnect 772
EVT_encoderaudiostart 776
EVT_encoderaudiostop 776
EVT_encodervideostart 776
EVT_encodervideostop 776
EVT_pause 773
EVT_play 773
EVT_publish 773
EVT_record 774
EVT_recordstop 774
EVT_seek 774
EVT_server_start 774
EVT_server_stop 774
EVT_setbuffertime 773
EVT_setstreamtype 773
EVT_stop 774
EVT_unpause 773
EVT_unpublish 773
EVT_vhost_start 774
EVT_vhost_stop 774
execute 1856
exists 10, 881, 1431
existsRTSP TunnelingSession 1272

expandEndTime 440, 502
expandEnvironmentVariables 104, 105
expandHTTPHeadersStr 77
expandStartTime 440, 502
expandToPortPair 1271
extractCodecConfigFromTrackInfo 1347
extractHTTPRequestInfo 640
extraDataClone 228

F

fakePosts 1097
fastDateFormat 169
FastPlaySettings 1453
fatal 748, 749
FCPublish 1031
FCSubscribe 1032
fcSubscribe 381
FCUnPublish 1031
FCUnpublish 1032
FCUnSubscribe 1033
FCUnsubscribe 1033
fcUnSubscribe 381
fcUnSubscribeAll 381
FD_ALL 782
FD_c_client_id 768
FD_c_ip 767
FD_c_proto 767
FD_c_referrer 767
FD_c_user_agent 767
FD_cs_bytes 768
FD_cs_bytes_inc 734
FD_cs_stream_bytes 768
FD_cs_stream_bytes_inc 734
FD_cs_uri_query 769
FD_cs_uri_stem 769
FD_date 766
FD_s_ip 769
FD_s_port 770
FD_s_uri 767
FD_sc_bytes 768
FD_sc_bytes_inc 734
FD_sc_stream_bytes 769
FD_sc_stream_bytes_inc 734
FD_time 766

FD_tz 766
FD_x_app 767
FD_x_appinst 767
FD_x_category 766
FD_x_comment 770
FD_x_ctx 766
FD_x_ctx_override 767
FD_x_duration 769
FD_x_duration_inc 734
FD_x_event 766
FD_x_file_ext 770
FD_x_file_length 768
FD_x_file_name 769
FD_x_file_size 768
FD_x_severity 770
FD_x_sname 768
FD_x_sname_query 769
FD_x_spos 768
FD_x_status 769
FD_x_stream_id 770
FD_x_suri 770
FD_x_suri_query 770
FD_x_suri_stem 770
FD_x_vhost 767
FILEEXTENSION 1417
fileFormat 719
FILEFORMAT_FLV 1817
FILEFORMAT_MP4 1817
FILEFORMAT_UNKNOWN 1817
fileTemplate 722
fileToArray 51
FileUtils 51
fileVersionDelegate 722
filterMetadata 1659
filterQueryStr 610
filters 563
FLAGS_DEFAULT 816
floatValue 171
flush 1421, 1432, 1449, 1522, 1650
flushBase64 29
FLV_CHUNKHEADER_BUFFERSIZE 59
FLV_CHUNKHEADER_FIRSTBYTE 59
FLV_CHUNKHEADER_HEADERSIZE 59
FLV_CHUNKHEADER_ISIZE 58
FLV_CHUNKHEADER_ITIMECODE 58
FLV_CHUNKHEADER_ITYPE 58
FLV_CHUNKHEADER_SECONDBYTE 59
FLV_CHUNKHEADER_VALUESIZE 59
FLV_DFRAME 59
FLV_KFRAME 59
FLV_PFRAME 59
FLV_TCINDEXAUDIO 60
FLV_TCINDEXDATA 60
FLV_TCINDEXVIDEO 60
FLV_UFRAME 59
FLVUtils 60
forceNewLoggerInstance 754
forceReset 905
FORMAT_COLOR 1701
FORMAT_FLASH 1701
FORMAT_FLV 707
FORMAT_ITALICS 1701
FORMAT_ITALICS_UNDERLINE 1701
FORMAT_MP4 707
formatBytes 43
formatBytesInSingleRow 44
formatBytesShort 42
formatBytesStruct 44
formatDeleteCookie 76
formatH264CodecConfig 1770
formatH264CodecConfigPacket 1770
formatKID 400
formatMilliseconds 45
formatRTPIInfo 1358
formatSeconds 45
formatSetCookie 77
formatUtcTime 46
FORWARD 8
frame 785
frames 787
frameTypeToString 62
fS 1407

G

generateAuth 90, 91, 92
generateAuthHTTP 91
generateHash 93
generateHashBytes 92, 93
generateKey 404

generateRandomKID 396
get 146, 182, 199, 232, 233, 1430, 1795
getAbsTimecode 213, 224
getAccess 1450, 1520
getActiveCount 1856
getAdapterName 604
getAdaptiveGroupItems 1059, 1101
getAddress 1788
getAddressCount 1286
getAddressRawStr 1789
getAddressStr 1789
getAdminAgent 1398
getAdminGUID 1405
getAdminInterfaceObjectList 1379, 1403
getAkamaiConfigName 1092
getAkamaiEventName 1092
getAkamaiHostId 1092
getAkamaiIngressAdaptiveGroupPath 1091
getAkamaiIngressAdaptiveGroupURL 1092
getAkamaiIngressURL 1091
getAkamaiOriginIp 1163
getAkamaiOriginPort 1163
getAkamaiPassword 1162
getAkamaiPlaybackAdaptiveGroupURL 1092
getAkamaiPlaybackDomain 1091
getAkamaiPlaybackURL 1092
getAkamaiPlaybackURLPrefix 1093
getAkamaiSessionKey 1160
getAkamaiStreamId 1091
getAkamaiUserName 1162
getAlgorithm 398
getAlgorithmString 398
getAllAsStrings 325
getAllowDomains 286
getAndClearNotifyCreate 645
getAndClearPendingFrameGrabs 1753
getAndClearPendingOverlays 1757
getAndSetStartStream 1606, 1615
getAppInstance 243, 358, 546, 608, 631, 691, 903, 1014, 1062, 1077, 1171, 1235, 1306, 1344, 1591, 1602, 1623, 1646
getAppInstanceName 1348, 1590
getAppInstanceNames 245
getAppInstanceProperty 1048
getApplication 272, 358, 739, 1014, 1822
getApplicationFolderNames 1833
getApplicationInstance 937, 1615
getApplicationInstanceTouchTimeout 305
getApplicationLock 1830
getApplicationNames 1833
getApplicationPath 243
getApplicationProperty 1047
getApplicationTimeout 283, 1828
getAppName 1348, 1590
getArchiveStrategy 408, 536, 554
getArtifactsTextRepresentation 412, 433
getAudioCodec 62, 70, 414
getAudioCodecConfigPacket 1521
getAudioCodecId 970
getAudioHost 1237, 1277
getAudioMissing 1505
getAudioMP3Layer 70
getAudioPacketizerItem 1268, 1316
getAudioPort 1236, 1276
getAudioSize 1490
getAudioTC 1493
getAudioTrack 1341
getAuthenticatePlayHandler 1305
getAuthenticatePublishHandler 1305
getAuthenticationList 1834
getAuthenticationMethod 565, 570
getAuthMode 1199
getAutoAllocateInterleavePorts 1359
getAVSyncMethod 1351
getBandwidth 1080
getBaseClass 887, 920, 1850
getBaseFilePath 696
getBasePath 10, 868
getBindAddress 1180
getBitrate 649
getBitrateAudio 969
getBitrateTotal 968
getBitrateTotalPeak 968
getBitrateVideo 969
getBitrateVideoPeak 969
getBlockCount 880
getBlockMask 892
getBlockPath 892
getBlockPtr 878
getBlockSize 880
getBodySize 799, 823, 827, 829, 832, 834, 840, 845, 849, 851,

855, 859
getBoolean 148, 184, 200, 201, 235, 237
getBuffer 598
getBufferSize 835
getBufferTime 368, 1489
getBuilder 1110
getBurstStartStop 1517
getByte 147, 184, 200, 201, 235, 236
getByteAllocation 78
getByteContainerLevel 214
getCacheControlStatus4xx 610
getCacheName 1515
getCallbackParamCount 1010
getCanHandleFilter 868
getCEACaptionConversionContext 1828
getCencMp4ProtSchemeString 399
getCencMpegTsProtSchemeString 399
getChecksum 1457
getChunkCounter 216
getChunkDurationTarget 1062
getChunker 547
getChunkGroupDuration 441
getChunkIndex 435
getChunks 211
getClassName 203, 206, 238
getClient 275, 330, 337, 1496, 1821
getClientById 274
getClientCount 274, 1825
getClientCountTotal 274
getClientID 1028
getClientId 357, 1496
getClientIdGenerator 1381, 1404
getClientIdGeneratorRecycleDelaySize 1407
getClientIdGeneratorRecycleSize 1407
getClientIdGeneratorTimeout 1407
getClientIdleFrequency 286, 1830
getClientProperty 1048
getClients 275, 1419, 1444
getClientsLockObj 294
getClientTimeout 1821
getClientUpdates 1447
getClosestStartTime 439, 501, 552
getCodecEntryForTime 496
getCodecId 969
getCommandInterface 1402
getCommandInterfaceCommandHandler 1383, 1407
getCommandInterfaceHostPort 1378, 1402
getCommittedVirtualMemory 1406
getCommonInitialTextRepString 423
getConce 1199
getConfigDataOrDefault 1075
getConfigDir 1858
getConfigPath 243
getConfiguation 1790
getConnectionCounter 244, 279, 1377, 1378, 1401, 1826
getConnectionFlashVerion 1161
getConnectionFlashVersion 1161
getConnectionHolder 630, 1308
getConnectionLimit 1827, 1859
getConnectionPageURL 1162
getConnectionQueryStr 1160
getConnectionsMaximum 1386, 1410
getConnectionSwfURL 1161
getConnectionTimeout 866, 920, 1063, 1100, 1156
getConnectionValidator 1401
getConnectLastAttempt 907, 1169
getConnectLastForceReset 907
getConnectLastSuccess 907, 1169
getConnectTime 366
getContentDescriptors 841
getContentLength 573
getContentType 574, 840
getContextStr 303, 533, 546, 1055, 1066, 1081, 1112, 1118, 1159, 1239, 1526, 1625, 1762
getCookieStr 635, 1315
getCoreHandlerPoolSize 1380, 1403, 1823
getCoreTransportPoolSize 1379, 1403, 1823
getCount 1589
getCpuTime 105
getCrossSession 1083
getCrossSessionLock 1084
getCryptoPoolActiveCount 1405
getCryptoPoolMaxSize 1405
getCurrent 344
getCurrentDuration 696
getCurrentFile 697
getCurrentHeapSize 1406
getCurrentItem 1665
getCurrentSize 696
getCurrentTime 521

getData 225, 432, 827, 830, 1435
getDataBuffer 225
getDataMissing 1505
getDataSize 1490
getDataTC 1494
getDataType 1495
getDate 48, 148, 184, 201, 202, 235, 237
getDateStarted 245, 279, 367, 1378, 1402, 1827
getDateString 49
getDefaultBlockSize 887
getDefaultStreamingStore 535
getDefaultStreamPrefix 1386, 1411
getDePacketizerList 1271
getDePacketizerName 1272
getDescription 800, 827, 849, 855, 887, 920, 1852
getDirection 1454
getDirecton 9
getDouble 148, 183, 200, 201, 234, 236
getDRMs 399
getDstAppInstanceName 1173, 1221
getDstApplicationName 1172, 1220
getDstStreamName 1054, 1060, 1062, 1078, 1171
getDuration 422, 1344, 1470, 1570, 1656, 1659
getDvrApplicationContext 312
getDvrArtifact 411, 432
getDvrChunkAtTime 551
getDvrChunkByIndex 551
getDvrChunkNearTime 552
getDvrFileSystemClass 536
getDvrManager 546, 1603
getDvrPlaylistRequest 509
getDvrProperties 312
getDvrRecorder 1523, 1527, 1594
getDvrRecorderId 1603
getDvrRecorderList 312, 1523, 1839
getDvrRecorders 1595
getDvrRepeater 1527
getDvrSessionInfo 640
getDvrStorageDir 535
getDvrStorageWindowSeconds 535, 545
getDvrStoreList 1839
getDvrTime 500
getDynamicLogProperties 1381, 1404, 1835
getElapsedTime 380, 639, 1317, 1524
getEmbeddedFlag 835
getEncInfo 1474
getEncIVBytes 397
getEncKeyBytes 396
getEncKeyBytesList 397
getEncKeyString 397
getEncMethod 1573
getEncoding 739
getEncRepeaterBytes 399
getEncryptionDelegate 536
getEncryptionRepeaterSharedSecret 536
getEncryptions 411
getEncryptionsTextRepresentation 412
getEncryptionType 488
getEndIndex 426
getEndTime 697
getEntriesToPurge 496
getError 1781
getErrorField 1781
getErrorString 697
getEventHandler 1782
getEventStr 1781
getExecutor 1856
getExt 1518
getExtent 1456
getExtraData 227
getExtraDataKeys 228
getFastPlaySettings 365, 1509
getField 1780
getFileExtension 1646
getFileFormat 693
getFileInBytes 86
getFileInBytesRate 88
getFileInfo 637
getFileIOPoolSize 1823
getFilename 714, 716, 724
getFileOutBytes 87
getFileOutBytesRate 88
getFilePath 696
getFilePointer 9, 883
getFileSystem 547
getFileVersionDelegate 698
getFirstByte 226
getFirstEntry 440, 498
getFirstIndex 440
getFirstRendition 976

getFirstSegment 960
getFlags 818, 859
getFlashVer 357
getFloat 148, 183, 200, 201, 234, 237
getFps 1453
getFrame 786
getFrameCount 1761
getFrameIdentifier 827
getFrameMapIds 792
getFrames 788, 792
getFramesById 793
getFrameType 62, 63
getGlobalLogValue 756
getGroupId 1060
getGroupName 1060
getGUID 1405
getHandler 1283
getHandlerThreadPool 1380, 1404, 1824
getHeader 572, 579
getHeaderBytes 576
getHeaderFlags 72
getHeaderMap 572
getHeaderNames 573
getHeaders 579
getHeaderSize 580, 1520
getHeight 971
getHighestVersionedStore 534
getHomePath 1822
getHost 1276, 1342
getHostname 1054, 1078, 1172, 1236
getHostPortsList 1824
getHostType 1238, 1278
getHowToPublish 1156
getHTTPHeader 640
getHTTPHeaderMap 640
getHTTPHeaderNames 641
getHTTPHeaderServer 1387, 1411
getHTTPIdent2Response 1792
getHTTPIdentResponse 1792
getHTTPIntHeader 640
getHTTPOriginMode 609
getHttpProviders 1791
getHTTPStreamerAdapter 630, 1837
getHTTPStreamerAdapterIDs 1791, 1838
getHTTPStreamerApplicationContext 301
getHTTPStreamerContext 1833
getHTTPStreamerItem 603
getHTTPStreamerList 296, 1837
getHTTPStreamerMaxPathLen 1840
getHTTPStreamerProperties 295
getHTTPStreamerRequestValidator 1840
getHTTPStreamerSession 1462, 1524
getHTTPStreamerSessionCount 299, 300
getHTTPStreamerSessionCountsByName 299
getHTTPStreamerSessions 299, 300
getID 605
getId 211
getIdleCheckFrequency 1831
getIdleFrequency 371, 604, 1313
getIdleHandler 1309
getIdleMinimumWaitTime 1834
getIdleTimeout 904
getIdleWorkerCount 1831
getIdleWorkers 1834
getIdStr 818, 858
getIdString 921
getImplementation 1074
getIndex 422, 585, 659, 666, 673, 1638
getIndexLive 659, 666
getIndexMap 438
getInfo 878
getIngressDomain 1063, 1090
getIngressPath 1063, 1091
getIngressPort 1063, 1091
getInIpAddress 1282
getInitialRepeaterItems 541
getInnerObj 206
getInPort 1283
getInputStream 575
getInstance 755, 1398, 1865
getInt 147, 183, 200, 201, 233, 236
getIntData 158
getIntHeader 573, 580
getIOPerformanceCounter 281, 637, 1315
getIoPerformanceCounter 245, 1377, 1401, 1826
getIOScheduler 1826
getIoSession 1318
getIp 365, 1306
getIpAddress 632
getIsRotatingKeys 396

getItems 1634
getJmxRemoteConfig 1403
getKeepAliveTime 921
getKeepAliveTimeout 1832
getKey 199, 232
getKeyRotationType 396
getKeys 199, 232
getKID 398
getKIDBytes 398
getKIDBytesSwapped 398
getKIDsBytesList 398
getKIDsBytesListSwapped 398
getKIDsList 398
getLanguage 824, 840, 972
getLastAccessed 1457
getLastAudioTC 1177
getLastAudioTimecode 1651
getLastAuthenticateMethod 1305
getLastChunkId 1062
getLastClientId 1437
getLastConnectAcceptedStamp 345
getLastConnectAcceptedStampString 345
getLastConnectAcceptedTime 345
getLastConnectAcceptedTimeString 345
getLastConnectRejectedByReasonStamp 346
getLastConnectRejectedByReasonStampString 346
getLastConnectRejectedByReasonTime 346
getLastConnectRejectedByReasonTimeString 346
getLastConnectRejectedStamp 345
getLastConnectRejectedStampString 345
getLastConnectRejectedTime 345
getLastConnectRejectedTimeString 346
getLastDataTC 1177
getLastDataTimecode 1651
getLastDisconnectStamp 347
getLastDisconnectStampString 347
getLastDisconnectTime 347
getLastDisconnectTimeString 347
getLastKeyFrame 1506
getLastLiveEntry 440, 498
getLastOperation 1437
getLastPacket 1507
getLastPacketizer 1112
getLastPacketsByType 1355
getLastPurgeTime 521
getLastRecordedEntry 440, 495
getLastRecordedIndex 437, 500
getLastRequest 645
getLastRun 1781
getLastSession 1158
getLastStreamId 1026
getLastTC 69
getLastTouchTime 305
getLastValidateTime 376
getLastVideoTC 1177
getLastVideoTimecode 1651
getLen 786, 788
getLength 1471, 1638
getLicenseCounter 1407
getLicenseHolder 1606, 1624
getLicenseTracker 1407
getLicenseValidator 1402
getLiveDuration 439, 500
getLiveEntries 439, 498
getLiveEntriesWithLimit 439, 499
getLivePlaylist 673
getLiveRangeEndingBeforeTime 440
getLiveRepeaterCapabilities 380
getLiveStreamDvrRecorderControl 312
getLiveStreamingPacketizer 631
getLiveStreamPacketizer 931, 1029, 1523, 1524, 1593
getLiveStreamPacketizerControl 302
getLiveStreamPacketizerId 1613
getLiveStreamPacketizerList 296, 379, 1522, 1838
getLiveStreamPacketizerLock 1593
getLiveStreamPacketizerName 1062
getLiveStreamPacketizerProperties 294
getLiveStreamRecordManager 1839
getLiveStreamRepeater 932, 1525
getLiveStreamTranscoder 1528
getLiveStreamTranscoderControl 311
getLiveStreamTranscoderItem 1624
getLiveStreamTranscoderList 311, 380, 1523, 1838
getLiveStreamTranscoders 1528
getLiveStreamTranscoderSessionCount 1383, 1409, 1841
getLiveTailEntries 439
getLiveThreads 1406
getLocalBindAddress 1221
getLocale 576
getLocalListeners 1164

getLocalPath 881
getLock 227, 628, 792, 908, 928, 937, 960, 968, 977, 1156, 1270, 1664
getLockCount 877, 930, 944
getLogger 744, 756, 1016
getLoggerObj 754, 756
getLong 147, 183, 200, 201, 234, 236
getMACAddress 105
getManagerProperties 273, 1381, 1404, 1822
getManifest 546
getManifestChannel 501
getManifestRepresentation 411, 414, 422, 433, 435
getMapFileIndex 1053, 1080
getMapIdStr 799, 819, 858
getMaxHeapSize 1406
getMaxHTTPSessionsPerTCPSession 611
getMaximumPendingReadBytes 285, 1829
getMaximumPendingWriteBytes 284, 379, 1829
getMaximumSetBufferTime 285, 379, 1830
getMaxRetries 1063, 1100
getMaxRTCPWaitTime 1351
getMaxStorageDirDepth 295
getMaxTimecode 1506, 1650
getMaxTimeToLive 880, 889
getMediaCache 888
getMediaCacheFilters 317
getMediaCacheItem 867, 884
getMediaCacheSource 877
getMediaCaster 929, 938, 1541
getMediaCasterCount 937
getMediaCasterDef 904, 924
getMediaCasterDefs 923
getMediaCasterHostPortConfig 925
getMediaCasterId 903, 929
getMediaCasterList 1833
getMediaCasterNames 923, 938
getMediaCasterProcessorCount 926
getMediaCasterProperties 294
getMediaCasterRTSPRTSPRTPTransportMode 315
getMediaCasterSettings 1832
getMediaCasterStreamItem 905
getMediaCasterStreamItems 937
getMediaCasterStreams 282
getMediaCasterType 908, 1343
getMediaCasterValidator 304
getMediaExtension 10, 1462, 1471
getMediaFile 878
getMediaIOPerformance 1510
getMediaIOPerformanceCounter 358
getMediaList 977, 1462
getMediaListProvider 315
getMediaListSegment 968
getMediaName 10, 879
getMediaNameExpression 869
getMediaNameFromContext 873
getMediaPath 891
getMediaReader 1360
getMediaReaderContentType 314, 1840
getMediaReaderProperties 294
getMediaReaders 1827
getMediaWriterProperties 294
getMediaWriters 1827
getMember 206
getMemberCount 205
getMembers 205
getMessagesAttemptedCount 1056, 1180, 1240
getMessagesFailedCount 1056, 1180, 1240
getMessagesInBytes 86
getMessagesInBytesRate 87
getMessagesInCount 86
getMessagesInCountRate 87
getMessagesLossBytes 87
getMessagesLossBytesRate 88
getMessagesLossCount 86
getMessagesLossCountRate 87
getMessagesOutBytes 86
getMessagesOutBytesRate 88
getMessagesOutCount 86
getMessagesOutCountRate 87
getMessagesRetriedCount 1056, 1180, 1240
getMetadata 1468
getMetadataEntryForTime 496
getMetadataPacket 1260, 1347
getMetaDataProvider 1520
getMethod 573
getMethodValue 1199
getMimeType 800
getMinimumAvailableChunks 535
getMinTimeToLive 880, 888
getMissing 222

getMode 1356
getModFunctions 282
getModuleInstance 283
getModuleList 283
getMp3TagMap 1827
getMPEGDashCPSubElements 403
getMPEGTSAudioBitrate 1366
getMPEGTSAudioLanguage 1352
getMPEGTSAudioPID 1353
getMPEGTSProgramID 1352
getMPEGTSVideoBitrate 1366
getMPEGTSVideoPID 1353
getmRtpDestination 1239
getMsgBytes 577
getMultiplier 1453
getName 243, 272, 870, 887, 919, 960, 970, 1279, 1423, 1436, 1448, 1492, 1636, 1638, 1667, 1793, 1824, 1850, 1858
getNameGroupByGroupName 1594
getNameGroupRemoveDelayTime 1595
getNameGroups 1594
getNameGroupStreamNames 1594
getNameSpaces 402
getNameSpaceString 399
getNCCounter 1200
getNetConnection 910, 1156, 1202, 1496
getNetConnectionHostPortConfig 1832
getNetConnectionIdleFrequency 1831
getNetConnectionProcessorCount 1825
getNextChunkIndex 500
getNextCodecIndex 501
getNextMapIndex 1083
getNextMetadataIndex 501
getNextNameGroupId 1593
getNextNetConnectionId 1821
getNextRun 1781
getNextSessionId 1321
getNextStreamIndex 1589
getNextTimeMapIndex 501
getNodeByTagName 113
getNodeValue 114
getNodeValueByTagName 113
getNonce 1199
getNormalizedNTPTimecode 1357
getNumberFromString 1080
getNumberLiveEntries 439, 497
getNumberRecordedEntries 439, 497
getObject 148, 160, 184, 201, 202, 235, 237
getObjectEncoding 159, 163, 215, 378
getObjectNames 1430
getObjectReference 164
getOffset 598
getOffsetToNextTag 835
getOnMetadataData 72
getOnMetadataFilter 1660
getOpaqueObject 884
getOrCreate 1430
getOriginalTimecodeThreshold 1076, 1170
getOutHost 1355
getOutIpAddress 1283
getOutPort 1283
getOutputStream 578
getOverlayImages 1755
getOwnerIdentifier 829
getPacketizerLock 1062
getPacketizerName 534
getPacketsSentAudio 1179
getPacketsSentData 1180
getPacketsSentVideo 1180
getPacketStartTime 421
getPacketType 599
getPageUrl 370, 1032
getParam 1010
getParamBoolean 1013, 1014
getParamCount 1010
getParamDate 1012
getParamDouble 1012, 1013
getParameter 574
getParameterMap 575
getParameterNames 574
getParameterValues 575
getParamInt 1012
getParamLong 1013
getParamMixedArray 1011
getParamObj 1011, 1194
getParamString 1011
getParamType 1010
getParamValue 111
getParent 1426, 1450, 1556
getPassword 336, 1077, 1124, 1194
getPath 11, 564, 576, 1461, 1471

getPathDelimiter 869
getPeakThreads 1406
getPendingWriteCount 879
getPendingWrites 879
getPerformanceCount 888
getPictureData 800
getPictureType 800
getPingRoundTripTime 367
getPingTimeout 284, 376, 1828
getPlayClass 1851
getPlayDuration 641, 971
getPlayer 1505
getPlayerCount 930, 944
getPlaylist 1665
getPlaylistAlignment 509
getPlaylistCount 1099
getPlaylistRenditionDescription 1059
getPlaylistTimeout 1099
getPlayPackets 1506
getPlayReadyDecryptorKey 1573
getPlaySeek 641, 971
getPlayStart 641, 971
getPlayStreamCount 282, 359
getPlayStreamCountsByName 282
getPlayStreams 358
getPlayStreamsByName 282
getPollingInterval 1668
getPort 740, 866, 1054, 1079, 1172, 1789
getProcessorCount 1790
getProfile 1065, 1118
getProfileData 1083
getProperties 244, 273, 360, 545, 603, 608, 639, 888, 892, 920, 960, 968, 977, 1053, 1075, 1307, 1380, 1404, 1506, 1603, 1612, 1624, 1822, 1851, 1859
getPropertiesPath 891
getProperty 324, 919, 1422, 1445, 1820, 1851
getPropertyBoolean 324
getPropertyDouble 325
getPropertyInt 325
getPropertyLong 325
getPropertyStr 324
getProtocol 371, 575
getProtocolUsage 247, 315, 1408, 1840
getProtocolUsageTracker 1408
getPsshData 404
getPublishAudioCodecId 1525
getPublishData 674
getPublisher 1124, 1194, 1198, 1668
getPublisherCount 298
getPublishers 298, 1625
getPublishersMaximum 1377, 1410
getPublishStream 1203
getPublishStreamCount 359
getPublishStreamNames 303, 1586
getPublishStreams 359
getPublishVideoCodecId 1525
getPurgeController 547
getPushPublishContext 1842
getPushPublishImpl 1065, 1118
getPushPublishSessionCount 319
getPushPublishSessions 317, 318
getPushPublishType 1055, 1080
getPushPublishVersionStr 1079
getQOPValue 1199
getQueryStr 369, 634, 1310, 1518
getQueryString 573
getQueueSize 1856
getRange 426
getReadAheadThreshold 889
getReaderClass 888
getReadRetries 866
getReadTimeout 865
getReadWriteTimeout 1063, 1100
getRealm 1198
getReceiveBufferSize 866
getReceiveVideoFPS 1510
getReconnectDelay 1169
getReconnectWaitTime 907, 1055, 1080
getRecordedDuration 438, 500
getRecordedEntries 437, 438, 494, 499
getRecordedEntriesInRange 438, 495
getRecordedEntriesMap 495
getRecordedEntriesWithLimit 438, 499, 551
getRecordedEntryByIndex 438, 495, 551
getRecordedEntryByTime 551
getRecordedEntryByTimeKey 438, 495
getRecordedEntryStartingBeforeTime 438
getRecorder 682
getRecorderParams 691
getRecordersList 682

getRecordersMap 682
getRecorderState 692
getRecorderStateString 692
getRecordingName 1606
getRecordingStreamStore 535
getRedirectSessionBody 644
getRedirectSessionCode 643, 1318
getRedirectSessionContentType 644
getRedirectSessionMessage 1318
getRedirectSessionURL 643, 1318
getRedundancyPairId 1060
getRefCount 1420, 1444
getReference 134
getReferrer 370, 634, 1032, 1310
getRemoteAddr 576
getRemoteHost 576, 739
getRendition 976
getRenditions 976
getRepeat 1635, 1667
getRepeaterItemsDvr 541
getRepeaterLastSequence 1613
getRepeaterMediaCasterDef 608
getRepeaterOriginUrl 285, 376, 1031
getRepeaterQueryString 286
getRequest 616
getRequestFilters 564, 570
getRequestType 615
getRequestURI 573
getRequestURL 574
getResolvedAkamaiStreamId 1090
getRespAMFAudioObj 369, 1517
getRespAMFDataObj 369, 1517
getRespAMFVideoObj 369, 1517
getRespFunctions 366
getResponseAMFObj 369
getResponseOnIncompleteChunk 611
getRsoStorageDir 288
getRsoStoragePath 288
getRTMPAuthProvider 1173
getRTMPHeaderServer 1386, 1411
getRTPAVSyncMethod 291
getRTPContext 1338, 1833
getRTPDatagramConfigIncoming 1834
getRTPDatagramConfigOutgoing 1834
getRTPDestination 1364
getRTPIIdleFrequency 287, 1830
getRTPInfo 1357
getRTPMaxRTCPWaitTime 292
getRTPPlayAuthenticationMethod 291
getRTPProperties 294
getRTPPublishAuthenticationMethod 291
getRTPPushPublishSession 1235
getRTPSession 329, 337, 1288
getRTPSessionCount 293
getRTPSessionCountsByName 293
getRTPSessions 292
getRTPStream 378, 1521
getRTPStreamContext 1367
getRtpStreamWaitTimeout 1239
getRTPWriteListener 1308
getRTSPBindIpAddress 307, 1361
getRTSPConnectionAddressType 308, 1361
getRTSPConnectionIpAddress 307, 1361
getRTSPMaximumPendingWriteBytes 305, 1360
getRTSPMaxPathLen 1841
getRTSPOriginAddressType 308, 1362
getRTSPOriginIpAddress 308, 1362
getRTSPPlayRangeStart 1314
getRTSPPlayRangeStop 1314
getRTSPRequestValidator 1841
getRTSPSessionDescription 1358
getRTSPSessionExtraLines 1364
getRTSPSessionName 1358
getRTSPSessionTimeout 305, 1360
getRTSPStream 1303
getRTSPTunnelingLock 1272
getRTSPTunnelingSession 1272
getRTSPTunnelingSessionId 1314
getSample 599
getScheme 575
getSDPData 1287
getSDPLang 1365
getSecondByte 226
getSecureTokenSharedSecret 1162
getSegment 959, 960
getSegmentationType 694
getSegmentDuration 695
getSegmentNumber 697
getSegmentSchedule 699
getSegmentScheduleString 695

getSegmentSize 695
getSendBufferSize 866
getSeq 225
getSerializer 203
getSerializeSize 400, 488
getServerGUID 1405
getServerHostPort 371
getServerIp 632, 1309
getServerName 575
getServerPort 576, 633, 1309
getSession 1202, 1321, 1340
getSessionDescriptionData 947
getSessionGUID 1405
getSessionHolderByNetConnection 1158
getSessionHolderBySession 1158
getSessionId 627, 1302
getSessionIds 1321
getSessionListDebug 1157
getSessionProtocol 605, 632
getSessions 1269
getSessionTimeout 629
getSessionTracker 678
getSessionType 631
getSharedObjectReadAccess 290, 372
getSharedObjects 275
getSharedObjectWriteAccess 290, 372
getShort 147, 183, 200, 201, 234, 236
getShortDescription 824
getSize 211, 222, 599, 793, 818, 859, 971
getSlot 1419, 1443
getSlotNames 1419, 1443
getSlots 1425, 1443
getSlotVersion 1436
getSourcePath 887
getSoVersion 1436
getSpeed 1660
getSrc 212, 223, 1491
getSrcStream 1055, 1081
getSrcStreamBufferSize 1179
getSrcStreamName 1053, 1060, 1062, 1078, 1171, 1235
getSSLConfig 1792
getSslFactoryClass 1789
getStart 1638
getStartIndex 426
getStartLiveOnPreviousBufferTime 1670
getStartOnPreviousBufferTime 1656
getStartTC 1454
getStartTCOffset 1454
getStartTime 697
getStartTimecode 421
getStartupStreams 1838
getStartupStreamsDelayTime 1842
getStopTimecode 421
getStorageDir 1424, 1432, 1449
getStore 892
getStoreItem 879
getStream 628, 692, 903, 1016, 1065, 1118, 1341, 1587, 1588, 1602, 1645
getStreamAngle 1188
getStreamArray 933
getStreamAttributes 1352
getStreamAudioSampleAccess 289, 373
getStreamBandWidth 1188
getStreamBaseName 534
getStreamBitrate 1030, 1773
getStreamClientless 1587
getStreamCloseWaitTime 1175
getStreamCount 282
getStreamDef 1854
getStreamDefs 1853
getStreamExt 636, 931, 1348
getStreamFile 367, 368
getStreamFileForRead 1516
getStreamFileForWrite 1516
getStreamFileMapper 295
getStreamId 1341
getStreamInfo 1342, 1352
getStreamKeyDir 287
getStreamKeyPath 288
getStreamLastSeq 906
getStreamLength 1031, 1773
getStreamList 934
getStreamListLock 1586
getStreamLock 1338
getStreamManager 937
getStreamMissingTime 906
getStreamName 546, 635, 691, 1348, 1623
getStreamNameAliasProvider 298
getStreamNameLock 1586
getStreamNameLogging 1349

getStreamNameParts 636
getStreamNames 944
getStreamPacketizerItem 1269, 1316
getStreamPort 1277
getStreamPosition 637, 1471
getStreamProperties 294
getStreamProperty 1049
getStreamQueryStr 1354
getStreamReadAccess 289, 374
getStreamRecorderProperties 319
getStreamRunningIndex 908
getStreams 272, 1497, 1586
getStreamSessionId 1349
getStreamSessionIp 1350
getStreamSessionVersion 1349
getStreamsToPublish 1203
getStreamStorageDir 287
getStreamStoragePath 288
getStreamStore 534
getStreamStores 534
getStreamTimeout 905
getStreamTimeoutLastReset 906
getStreamTimeoutLastTime 906
getStreamTimeoutReason 906
getStreamTrack 1341
getStreamType 273, 360, 918, 1029, 1347, 1507, 1645
getStreamTypeNames 1854
getStreamTypes 1820
getStreamTypeStr 609
getStreamVersionHandler 536
getStreamVideoSampleAccess 288, 373
getStreamWriteAccess 290, 375
getString 147, 160, 182, 200, 201, 233, 235
getStringReference 164
getSyncLock 877
getSyncPacketType 1181
getSyncTimecode 1181
getSystemId 402
getSystemName 403
getSystemTime 106
getTarget 1780
getTargetEncoding 163
getText 824
getTextEncoding 799, 823, 840, 846, 849
getThreadPool 1380, 1404, 1824
getTime 49
getTimecode 212, 224, 842, 1253
getTimecodesOutOfOrderThreshold 1176
getTimedTextProperties 297
getTimedTextProviderConfig 1838
getTimeMap 503, 558
getTimeMapEntries 558
getTimeMapper 546
getTimeMapping 435
getTimeOffsetBetweenItems 1670
getTimeRunning 245, 280, 367, 639, 1317, 1378, 1402, 1827
getTimeRunningSeconds 245, 280, 367, 639, 1317, 1378, 1402, 1827
getTimeSeconds 49
getTimeStampFormat 840
getTimeString 49
getTimeToLive 1239
getTitle 960, 973
getTotal 347
getTotalAccepted 347
getTotalIOPerformanceCounter 357
getTotalPacketsSent 1176
getTotalRejected 348
getTrack 1282, 1340
getTrackId 1340
getTrackNames 1340
getTrait 160, 203
getTraitReference 164
getTranscoderApplicationContext 312
getTranscoderLicenseInUse 1384, 1409
getTranscoderLicenseTotal 1384, 1409
getTranscoderName 1623
getTranscoderProperties 295
getTranscoderType 1625
getTranscoderVODContext 1840
getTranscoderVODDestination 649
getTranscoderVODIndex 646, 647
getTranscoderVODIndexDestinationsMap 646
getTranscoderVODIndexDestinationsOrder 646
getTranscoderVODSession 645
getTransportMode 1357
getTransportThreadPool 1380, 1404, 1824
getTTL 1278, 1285
getTuningHandler 1383, 1409
getType 132, 182, 212, 223, 422, 426, 437, 970, 1065, 1117, 1254

getTypeAsString 968
getTypeStr 1792
getUDPManagedDeliveryCount 1365
getUDPManagedDeliveryDelay 1365
getUDPPortManager 1382, 1405
getUDPPortSharingManager 1382, 1405
getUDPTransport 1282, 1351
getUDPTransportManager 1270, 1837
getUnidentifiedSessionTimeout 1842
getUniqueId 929
getUniqueStreamIdStr 1524
getUnsignedShort 39
getUri 370, 633, 1310
getURL 402, 827, 851, 855
getUserAgent 633, 1304
getUserAgents 1379, 1402
getUserHTTPHeader 610, 642
getUserName 1077, 1124, 1194
getUserQueryStr 639
getUserTime 106
getUtcStartTime 421
getValidationFrequency 284, 1164, 1829
getValue 137, 149, 154, 172, 186, 203, 843, 845, 849
getVersion 117, 1028, 1376, 1400, 1421, 1445
getVersioningOption 694
getVHost 244, 273, 329, 337, 363, 604, 608, 627, 903, 1015, 1270, 1304, 1322, 1342, 1590
getVHostItems 1863
getVHostList 1381, 1405
getVHostMap 1862
getVHostNames 1863, 1866
getVideoCodec 62, 70, 414
getVideoCodecConfigPacket 1521
getVideoCodecId 969
getVideoFrameType 70
getVideoHost 1237, 1277
getVideoMissing 1505
getVideoPacketizerItem 1269, 1316
getVideoPort 1238, 1276
getVideoSize 1490
getVideoTC 1494
getVideoTimecodeOffset 70, 71
getVideoTrack 1341
getVODLastTimeTC 1363
getVODPlayLen 1363
getVODSpeed 1671
getVODStartTimeTC 1362
getVODTimedTextProviderList 297
getVODTimedTextProviderSet 297
getVODTranscodeNGRP 645, 972
getWaitOnMetadataTimeout 1178
getWidth 970
getWowzSession 382
getWriteListener 375
getXMLPropertyBool 116
getXMLPropertyDouble 116
getXMLPropertyExists 115
getXMLPropertyInt 115
getXMLPropertyIntSize 115
getXMLPropertyLong 115
getXMLPropertyLongSize 116
getXMLPropertyStr 114
gmtTimeZone 104
grabFrame 1753
groupPlaylistNeedsUpdating 1058
GZIP 18

H

handleArchivedStream 408, 554
handleCallback 1511
handleConnectResponse 1123, 1191, 1198
handleMessage 1259, 1285
handleOnMetadata 1622
handlePacket 1602, 1613, 1622
hasAudio 502, 548
hasCodecData 502
hasData 502, 548
hasEncryption 548
hasOnMetadata 502, 548
hasTimeMapData 503
hasVideo 502, 548
head 861
HEADERSIZE 808
height 967
hexadecimal 32
hostname 1073
HOSTNAME_KEY 728
HostPort 1788
HostPortList 1794

-
- hour 1779
- howToPublish 1149
- httpIngressDomainOverride 1089
- httpIngressDomainSuffix 1089
- HTTPORIGINMODE_AUTO 607
- HTTPORIGINMODE_OFF 607
- HTTPORIGINMODE_ON 607
- httpPlaybackDomainOverride 1089
- httpPlaybackDomainSuffix 1089
- HTTTPoller 1115
- HTTPRequestToByteArray 74, 75
- HTTPRequestToFile 74
- HTTPProvider2Base 563
- HTTPStreamerSessionCupertino 584
- HTTPStreamerSessionDirect 596
- HTTPStreamerSessionMPEGDash 658
- HTTPStreamerSessionSanJose 665
- HTTPStreamerSessionSmoothStreamer 672
- HTTPStreamerSessionWebM 678
- HTTTPUtils 74
- httpWriterDebug 1098
- I**
- ID3DeserializeFrameResult 785
- ID3DeserializeFramesResult 787
- ID3FOOTER_SIZE 791
- ID3Frames 792
- ID3HEADER_SIZE 791
- ID3HEADER_VERSION 791
- ID3HEADERFLAGS_DEFAULT 791
- ID3HEADERFLAGS_EXPERIMENTAL 791
- ID3HEADERFLAGS_EXTENDED 791
- ID3HEADERFLAGS_FOOTERPRESENT 791
- ID3HEADERFLAGS_UNSYNC 791
- ID3V2FrameAttachedPicture 799
- ID3V2FrameBase 816
- ID3V2FrameComment 823
- ID3V2FrameLinkedInformation 826
- ID3V2FramePrivate 829
- ID3V2FrameRawBytes 832
- ID3V2FrameRecommendedBufferSize 834
- ID3V2FrameSynchronizedText 839
- ID3V2FrameSynchronizedTextDescriptor 842
- ID3V2FrameTextInformation 845
- ID3V2FrameTextInformationUserDefined 848
- ID3V2FrameURLLink 851
- ID3V2FrameURLLinkUserDefined 854
- ID3V2Utils 856
- idle 1355, 1496
- idleFrequency 1301
- idleHandler 1300
- idStringToName 921
- importManifest 501
- incAbsTimecode 214
- incByteContainerLevel 214
- incClientCountTotal 274
- incPendingWriteCount 879
- incrementAccept 343
- incrementBytesIn 83
- incrementBytesLoss 85
- incrementBytesOut 85
- incrementFileIn 83
- incrementFileOut 84
- incrementMediaInBytes 1339, 1511
- incrementMediaLossBytes 1511
- incrementMediaOutBytes 1510
- incrementMessagesIn 83, 84
- incrementMessagesLoss 84, 85, 86
- incrementMessagesOut 84, 85
- incrementReject 343
- incSlotVersion 1436
- index 420
- indexOf 39
- indexOfDifferent 39
- indexTranscoderVOD 586, 646, 660, 667, 674
- inetAddr 1073
- info 746, 747
- init 8, 331, 332, 442, 444, 520, 525, 545, 556, 557, 563, 570, 604, 608, 681, 863, 867, 873, 877, 902, 929, 947, 1055, 1074, 1090, 1098, 1123, 1186, 1191, 1193, 1198, 1221, 1226, 1234, 1268, 1435, 1460, 1467, 1489, 1552, 1579, 1601, 1612, 1621, 1820, 1856, 1865
- initOnce 1198
- initContextLogging 755
- initialAttributes 541
- initialize 494
- initializeLogging 755
- initialManifest 541
- initialManifestEnd 541
- initLiveStreamRepeating 1029, 1525, 1556
-

initMedia 892
initProperties 1344
initStream 1026
InputStream 25
interleavePackets 65, 66
internalConnect 1160, 1226
interruptPlay 1555
intToByteArray 37
intToByteArraySafeSync 857
intToHexStr 100
intToStr 100
intValue 171
invoke 1369
invokePrevious 1016, 1017
IOPerformanceCounter 82
ioPerformanceCounter 1299
ioSession 1302
ioSessionToSessionHolder 1149
ip 1298
isAcceptConnection 274, 360
isAcceptConnectionSendClientId 363
isAcceptSession 635
isAcceptWOWZConnections 1383, 1408
isActive 628, 739, 1603, 1612
isAdaptiveStreaming 1076
isAddressMulticast 96
isAdmin 1793
isAMF0 159, 164
isAMF3 159, 164
isAMF3Start 133
isAnnounce 1299, 1307
isAnnounceOrDescribe 1308
isAppend 1504
isAppendFile 699
isAppInstanceLoaded 243
isApplicationLoaded 1822
isArrayStart 133
isAudio 225
isAudioCodecConfig 69
isAVSyncNonSR 1351
isBlockInCache 878
isBlockUDPOut 1286, 1366
isByteArrayStart 134
isByteContainerEmpty 214
isByteContainerFull 214
isCheckIpAddr 1350
isCheckSSRC 1350
isClient 1419, 1444
isClustered 1515
isComplete 405
isConnected 360, 1056, 1180, 1239, 1300, 1308
isCurrentSession 1157
isDebugAppTimeout 315
isDebugEnabled 745
isDebugLog 88, 1059, 1062, 1171, 1220
isDebugOn 1782
isDebugPackets 1176, 1219
isDebugRTSPSession 1313
isDescribe 1299, 1307
isDRMAvailable 1384, 1410
isDRMInUse 1385, 1410
isDVRAvailable 1384, 1410
isDVRInUse 1384, 1410
isDynamic 206
isDynamicLogContextLoaded 1381, 1404
isEmpty 100, 427, 429, 437, 792
isEnabledFor 745
isEncoderInUse 1761
isEncrypted 371
isErrorEnabled 745
isExpired 1782
isExtraData 228
isFakePosts 1099
isFileInfo 637
isFirstABSTRequest 667
isFlashMediaLiveEncoder 377
isFlashVerionFMLE 1175
isFlashVersion10 377
isFlashVersion90115 377
isFlashVersionFMLE 1175
isFlashVersionH264Capable 377
isForceAMF0 1254
isForceMPEGTSOut 1366
isForceRTSPInterleaved 1357
isGlobalLogValueSet 756
isHeadRequest 577
isHeadResponse 580
isHitEnd 1458
isHTTPOrigin 645
isHTTPOriginAvailable 1385, 1410

isHTTPOriginOn 610
isHttpWriterDebug 1062, 1100
isID3Footer 794
isID3Header 793
isInAdaptiveGroup 1059, 1101
isInfoEnabled 745
isInitiateWOWZConnections 1383, 1408
isInMulticast 1285
isInRange 427, 429
isInRedundancyPair 1059, 1091
isIntData 158
isLastSentAbsTimecode 215
isLive 319, 550, 1359
isLiveRepeater 376
isLiveStreamPacketize 1618
isLiveStreamTranscode 1626
isLoaded 547
isLocal 881
isLoggedConnect 1306
isLongTimecode 215
isMatch 1780
isMediaCacheSourceAllowed 317
isMediaCasterPlay 1526
isMember 205
isMergeOnMetadata 1526
isMixedArrayStart 134
isModePlay 1354
isModePublish 1354
isModeUnknown 1354
isMoveFirstVideoFrameToZero 694
isMoveToNextIfLiveStreamMissing 1670
isMPEGTSOut 1359
isMulticast 1275, 1285
isNew 213
ISO8601_FORMAT 730
ISO8601_PATTERN 730
isObjectEncodingAMF0 215, 378
isObjectEncodingAMF3 215, 377
isObjEnd 134
isObjStart 133
isOnMetadataPacket 72
isOnMetadataToSetDataFrame 1174
isOpen 9, 1461, 1468, 1508
isOutMulticast 1285
isOverlayAvailable 1756
isPacketizeAudio 652, 1614
isPacketizeData 652, 1615
isPacketizeVideo 652, 1614
isPassThru 889
isPaused 1344
isPending 1257
isPersistent 1420, 1431, 1444
isPlay 1495
isPlaying 1490
isPlaylistAcrossSessions 1099
isPlaylistReady 508, 509
isPlayLogged 630
isPublishDataEvents 1651
isPublisherStream 1530
isPublishStreamReady 1343, 1526
isPurgingEnabled 520
isPushPublishAvailable 1385, 1410
isReadAhead 889
isReadAheadTrigger 880
isReceiveAudio 1509
isReceiveVideo 1510
isReconnectorRunning 1150
isRecord 1495
isRecordData 693
isRecording 537, 550, 1605
isRecordingPaused 537, 550, 1605
isRedirectSession 643, 1317
isRedundant 1091
isRefuseIncomingUDPConnections 1367
isRefuseOutgoingUDPConnections 1367
isRemoveDefaultAppInstance 1174
isRepeaterEdge 1614
isRequestFullURL 865
isResetOnTimecodeOutOfOrder 1176
isResetPlayStream 1344
isRESTAPIAvailable 1385, 1410
isRestDocumentationServerEnabled 1409
isRestServerEnabled 1385, 1409
isResyncAudioVideoOnSR 1346
isReuseSockets 1100
isRTPIgnoreProfileLevelId 1365
isRTPIgnoreSPropParameterSets 1366
isRTPIncomingDatagramPortValid 301
isRTPWrapped 1234, 1238, 1278
isRTSP 1353

isRTSPAlwaysUseSDPPorts 1364
isRTSPPull 1354
isRunning 247, 1761
isSecure 371, 576, 647
isSendCloseOnRejectConnection 382
isSendFCAnnounce 1163
isSendFCPublish 1157
isSendOnMetadata 1173, 1632, 1653, 1669
isSendOriginalTimecodes 1076, 1170
isSendPlayStopLogEvent 1519
isSendPublishStopLogEvent 1520
isSendRecordStopLogEvent 1519
isSendReleaseStream 1157
isSendResult 1014
isSendSDESEvents 1358
isSendStreamCloseCommands 1174
isSendToBackupServer 1060, 1090
isSession 906
isSessionValid 1057, 1110, 1298, 1305
isShutdownClient 1315
isShutdownOnRelease 931
isShuttingDown 1834
isSingleStreamAV 1236
isSplitOnTcDiscontinuity 693
isSSL 371, 1155, 1180
isStartLiveOnPreviousKeyFrame 1669
isStartOnKeyFrame 693
isStartOnPreviousKeyFrame 1656
isStartStarupStreams 1839
isStopIfStreamMissing 1656
isStream 906, 1276
isStreamDomainProtectionActive 609
isStreaming 1793
isStreamIsRunning 907
isStreamStarted 1363
isSuspended 1382, 1400, 1791, 1836
isSwitchLog 1669
isTemplateLoaded 1624
isTimeout 628, 1361
isTimeoutSession 629
isTimesInMilliseconds 1669
isTraceEnabled 745
isTranscoderActive 1624
isTranscodeResult 1528
isTranscoderWatermark 1384, 1409

isUnpublishOnEnd 1671
isUseUstreamPingPong 1225
isValid 399, 929, 1475
isValidated 629
isValidFMLEConnections 309, 381
isValidAudioCodec 652
isValidStreamDomainStr 638
isValidVideoCodec 652
isVersionFile 699, 1569
isVHostRunning 1399
isVideo 225
isVideoCodecConfig 69
isVideoH264SEIListenerEmpty 1529
isVideoKeyFrame 69, 71
isVisited 1859
isWaitForVideoKeyFrame 1569
isWaitOnMetadataAudio 1178
isWaitOnMetadataAudioDatarate 1179
isWaitOnMetadataAvailable 1177
isWaitOnMetadataVideo 1177
isWaitOnMetadataVideoDatarate 1178
isWarnEnabled 744
isWowzaAudioOnly 970
isWowzaKeyFrameOnly 972
isWowzaVideoOnly 972

J

JNDI_CONTEXT_NAME 729

K

KEY_ROTATION_TYPE_EXTERNAL 395
KEY_ROTATION_TYPE_INTERNAL_RANDOM 395
KEY_ROTATION_TYPE_INTERNAL_STATIC 395
KEY_ROTATION_TYPE_NONE 395
KEYDATA_MODE_ENCRYPT 987
KEYDATA_MODE_INIT 987
KEYDATA_MODE_KEYFRAME_ENCRYPT 987
KEYDATA_MODE_KEYFRAME_INIT 987
KEYDATA_MODE_KEYFRAME_PLAYLIST 987
KEYDATA_MODE_PLAYLIST 987
killClient 1821
killRTSPSession 1822

L

LANG_ASCII 1698
LANG_ASCII_DBL 1699
LANG_GB231280 1699
LANG_KSC56011987 1699
LANG_PRIV1 1699
LANG_PRIV2 1699
LANG_REG1 1699
language 822
LANGUAGE_ID_ENGLISH 1725
LANGUAGE_LOCALE 1725
lastAudioTC 1153
lastAuthenticateMethod 1298
lastDataTC 1153
lastIdleReadTime 1153
lastIdleWriteTime 1153
lastModified 10, 881, 884
lastUpdate 1110
lastUpdateTime 1115
lastVideoTC 1153
len 217, 785, 787
length 11, 101, 881, 884, 893, 1518, 1555
LICENSECOUNTER_DRM_BUYDRM_LIVE 1811
LICENSECOUNTER_DRM_BUYDRM_VOD 1811
LICENSECOUNTER_DRM_EZDRM_LIVE 1810
LICENSECOUNTER_DRM_EZDRM_VOD 1810
LICENSECOUNTER_DRM_VERIMATRIX_LIVE 1810
LICENSECOUNTER_DRM_VERIMATRIX_VOD 1811
LICENSECOUNTER_NDVR 1810
LICENSECOUNTER_PUBLISHER 1810
LICENSECOUNTER_PUBLISHERTRANSCODER 1811
LICENSECOUNTER_TOTAL 1812
LICENSECOUNTER_TRANSCODE_DECODE 1810
LICENSECOUNTER_TRANSCODE_DECODECOUNTAU
DIO 1811
LICENSECOUNTER_TRANSCODE_DECODECOUNTAU
DIOVIDEO 1812
LICENSECOUNTER_TRANSCODE_DECODECOUNTVID
EO 1811
LICENSECOUNTER_TRANSCODE_DECODEPOLLING
1811
LICENSECOUNTER_TRANSCODE_ENCODE 1810
LICENSECOUNTER_TRANSCODE_ENCODECOUNTAU
DIO 1812
LICENSECOUNTER_TRANSCODE_ENCODECOUNTAU
DIOVIDEO 1812
LICENSECOUNTER_TRANSCODE_ENCODECOUNTVID
EO 1812
LICENSECOUNTER_TRANSCODE_ENCODEPOLLING
1811
LICENSECOUNTER_TRANSCODE_STREAMNAMES
1811
LICENSECOUNTERS_DRM 1812
listeners 1149
liveStreamPacketizer 1103
liveStreamPacketizers 1585
load 1080, 1093, 1101, 1187, 1219, 1226, 1239, 1433, 1449
loadArchivedRecordings 525
loadArchivedStore 538
loadCacheItem 886
loadConfig 1862
loadConfigFile 1769, 1775
loadConfigProperties 114
loadFile 801
loadFromKeyFile 400
lock 877, 958, 965, 975, 1149, 1193, 1234, 1302, 1426, 1449
lockRepeaterStreams 634, 1343
log 745, 746
LOG4J_ID_KEY 728
LOG4J_PACKAGE_NAME 727
logDebug 1020
logError 1020
loggedConnect 1299
LOGGERNAME_SERVER 754
logInfo 1020
logLiveChunk 585, 659, 666
logLiveFragment 673
logNotifier 1398
LogNotifyCalculateIncremental 734
logVODChunk 585, 659, 666
logVODFragment 673
logWarn 1020
longToByteArray 38
longToHexStr 100
longToStr 100
longValue 171

M

main 18, 1399
makeNewLoggerInstance 755
MANIFEST_TAGNAME_CAN_PLAY 493
MANIFEST_TAGNAME_CAN_RECORD 493
MANIFEST_TAGNAME_CHUNK_GROUPING 494
MANIFEST_TAGNAME_CURRENT_TIME 494
MANIFEST_TAGNAME_HAS_ENCRYPTION 493
MANIFEST_TAGNAME_PURGE_TIME 494
MANIFESTFILE_KEY_ARTIFACT 419
MANIFESTFILE_KEY_AUDIO_CODEC 419
MANIFESTFILE_KEY_CHUNKINDEX 419
MANIFESTFILE_KEY_DVRTIME 419
MANIFESTFILE_KEY_ENCRYPTIONS 420
MANIFESTFILE_KEY_INDEX 418
MANIFESTFILE_KEY_METADATA 420
MANIFESTFILE_KEY_NAME 419
MANIFESTFILE_KEY_PACKETTIME 419
MANIFESTFILE_KEY_SIZE 419
MANIFESTFILE_KEY_START 418
MANIFESTFILE_KEY_STOP 418
MANIFESTFILE_KEY_TYPE 418
MANIFESTFILE_KEY_UTCTIME 419
MANIFESTFILE_KEY_VIDEO_CODEC 419
map 1109
mapFileIndex 1072
markAllAsDiscontinuity 1110
markEntryAsDiscontinuity 1108
MAX_CCCOUNT 1693
maxRetries 1098
MAXSTREAMINDEX 1584
md5Digest 90, 1191
MD5DigestUtils 90
md5Lock 90
MEDIACACHE_PREFIX 458
MediaCasterItem 918
MediaCasterList 923
MediaCasterSettings 925
MediaCasterStreamItem 928
MediaCasterStreamManager 933
MediaCasterStreamMap 936
MEDIACASTERTYPE_LIVEREPEATER 902
MEDIACASTERTYPE_RTPLIVE 902
MEDIACASTERTYPE_SHOUTCAST 902
MEDIACASTERTYPE_UNKNOWN 902
MediaList 959
mediaList 975
MediaListRendition 967
MediaListSegment 975
mediaListSegment 965
MediaReaderEncInfo 1573
MediaStreamBase 1579
mediaStreamListeners 1585
MediaStreamMap 1586
MediaUtils 95
members 197
METHOD_NONE 1572
METHOD_SAMPLE_PLAYREADY 1573
methodValue 1197
MILLS_PER_HOUR 131
MIMETYPE_VIDEO_MP4 479
MIMETYPES_JPEG 798
MIMETYPES_PNG 798
MIMETYPES_URL 798
mImplementation 1074
MIN_PLAYLIST_COUNT 1097
MINUTE 1779
MISSING 1456
MODE_PLAY 1335
MODE_PUBLISH 1335
MODE_UNKNOWN 1335
ModuleBase 1010
ModuleClientLogging 1020
ModuleCore 1025
ModuleFastPlay 1034
ModuleMediaCaster 943
ModuleMediaCasterStreamMonitorAdvanced 1044
ModuleProperties 1047
monitors 1043
MONTH 1779
moveFirstVideoFrameToZero 719
MPEGDASH_MANIFEST 615
MPEGDASH_MEDIA 615
mRtpDestination 1234
msb0baseTime 104
msb1baseTime 104
MSG_STARTSTREAM 612
MSG_SWITCHSTREAM 612
mType 1074

N

name 958, 965
nameGroupDelayRemoveMap 1585
nameGroupId 1585
nameGroupRemoveDelayTime 1585
nameGroups 1584
ncCounter 1197
netConnection 1202
netconnectToSessionHolder 1148
NetworkUtils 96
newCacheItem 886
newValue 107
newXPathFactory 116
next 1667
nextChunkId 1107
nextStreamId 1585
NO_OPTIONS 17
nonce 1197
NONE 614
notifyActionOnCodecInfoAudio 1515
notifyActionOnCodecInfoVideo 1515
notifyActionOnMetaData 1514
notifyActionPause 1514
notifyActionPauseRaw 1514
notifyActionPlay 1513
notifyActionPublish 1514
notifyActionSeek 1514
notifyActionStop 1515
notifyActionUnPublish 1514
notifyAkamaiClientLogin 1166
notifyAkamaiSetChallenge 1165
notifyAkamaiSetOriginConnectionInfo 1165
notifyConnect 1166
notifyConnectFailure 1165
notifyConnectStart 1165
notifyConnectSuccess 1165
notifyDvrStreamManagerCreate 314
notifyDvrStreamManagerDestroy 314
notifyDvrStreamManagerInit 314
notifyDvrStreamStorageDeleted 540
notifyDvrStreamStoreCreate 540
notifyDvrStreamStoreDestroy 540
notifyDvrStreamStoreInit 540
notifyDvrStreamStoreLoaded 540
notifyFCAnnounce 1166
notifyFCPublish 1166
notifyHandshakeResult 1165
notifyHTTPSessionCreate 317, 585, 601, 660, 667, 673
notifyHTTPSessionDestroy 317, 585, 601, 660, 667, 674
notifyListener 721
notifyLivePacket 1530
notifyLiveStreamDvrRecorderCreate 313
notifyLiveStreamDvrRecorderDestroy 313
notifyLiveStreamDvrRecorderInit 313
notifyLiveStreamPacketizerCreate 309
notifyLiveStreamPacketizerDestroy 309
notifyLiveStreamPacketizerInit 309
notifyLiveStreamTranscoderCreate 310
notifyLiveStreamTranscoderDestroy 310
notifyLiveStreamTranscoderInit 310
notifyMediaReaderClose 307
notifyMediaReaderCreate 306
notifyMediaReaderExtractMetaData 307
notifyMediaReaderInit 306
notifyMediaReaderOpen 306
notifyMediaStreamCreate 1592
notifyMediaStreamDestroy 1592
notifyMediaWriterOnFLVAddMetadata 304
notifyMediaWriterOnWriteComplete 304
notifyPlayPublish 1592
notifyPlayUnpublish 1591
notifyPublishHandlerPlay 1167
notifyPushPublisherSessionCreate 1168
notifyPushPublisherSessionDestroy 1169
notifyPushPublishSessionCreate 319
notifyPushPublishSessionDestroy 319
notifyRTPSessionCreate 316
notifyRTPSessionDestroy 316
notifySessionClosed 1168
notifySessionCreate 1323
notifySessionDestroy 1323
notifySessionIdle 1168
notifySessionOpened 1168
notifySlotDelete 1449
notifySlotSetValue 1449
notifyStreamCreate 1167
notifyStreamOnPlayStatus 1167
notifyStreamOnStatus 1167
notifyTimeReset 540
notifyValidateSession 1168

notifyValidateSessionResult 1168
notifyVHostClientConnect 1866
notifyVHostCreate 1867
notifyVHostInit 1867
notifyVHostItemCreate 1862
notifyVHostItemDestroy 1863
notifyVHostItemUpdate 1863
notifyVHostShutdownComplete 1867
notifyVHostShutdownStart 1867
notifyVideoH264Packet 1529

O

ODDPARITY 1693
offlineTime 1155
offset 217
ON_CAPTION 1743
ON_CAPTION_INFO 1743
ON_METADATA_TYPE 493
ON_TEXT_DATA 1743
onAACEncodeInfo 954
onAcceptConnection 385
onAcceptorCreate 1796
onAcceptorDestroy 1796
onAfterDecodeFrame 1758
onAfterEncodeFrame 1760
onAfterScaleFrame 1758
onAkamaiClientLogin 1127, 1206
onAkamaiSetChallenge 1127, 1206
onAkamaiSetOriginConnectionInfo 1127, 1206
onAnnounce 1244, 1263, 1311
onAppCreate 982
onAppDestroy 982
onApplicationCreate 321
onApplicationDestroy 321
onApplicationInstanceCreate 320
onApplicationInstanceDestroy 320
onAppStart 943, 981, 1044
onAppStop 943, 981, 1044
onBeforeDecodeFrame 1758
onBeforeEncodeFrame 1760
onBeforeScaleFrame 1758
onBind 565, 566
onCalculateSourceAudioBitrate 1748
onCalculateSourceVideoBitrate 1748
onCall 983
onCallback 1536
onClientAccept 383
onClientConnect 383
onClientDisconnect 383
onClientReject 384
onCodecConfigAAC 951
onCodecInfoAudio 1535
onCodecInfoVideo 1535
onConnect 984, 1127, 1207
onConnectAccept 984
onConnectFailure 914, 1126, 1206
onConnectReject 985
onConnectStart 913, 1126, 1206
onConnectSuccess 913, 1126, 1206
onCreate 587
onCreateRecord 686
onCreateRecorder 700
onCronEvent 1784
onData 909
onDescribe 1244, 1263, 1311
onDestroy 589
onDisconnect 386, 949, 984
onDvrStreamManagerCreate 1597, 1598
onDvrStreamManagerDestroy 1597, 1598
onDvrStreamManagerInit 1597, 1598
onDvrStreamManagerInitStreamName 1599
onFCAnnounce 1127, 1207
onFCPublish 1127, 1207
onFile 79, 895
onFillChunkDataPacket 590, 592
onFillChunkEnd 589, 592
onFillChunkStart 589, 591
onFLVAddMetadata 1571
onFrameAAC 951
onFrameMP3 951
onGetParameter 1244, 1263, 1311
onGrabFrame 1754
onHandshakeResult 1126, 1206
onHeaderData 952, 954
onHTTPCupertinoEncryptionKeyCreateLive 988
onHTTPCupertinoEncryptionKeyCreateVOD 988
onHTTPCupertinoEncryptionKeyData 988
onHTTPCupertinoEncryptionKeyLiveChunk 989
onHTTPCupertinoEncryptionKeyRequest 987

- onHTTPCupertinoEncryptionKeyVODChunk 988
- onHTTPCupertinoStreamingSessionCreate 990
- onHTTPCupertinoStreamingSessionDestroy 990
- onHTTPMPEGDashEncryptionKeyLiveChunk 991
- onHTTPMPEGDashEncryptionKeyVODChunk 991
- onHTTPMPEGDashStreamingSessionCreate 992
- onHTTPMPEGDashStreamingSessionDestroy 992
- onHTTPRequest 566
- onHTTPSanJoseStreamingSessionCreate 993
- onHTTPSanJoseStreamingSessionDestroy 993
- onHTTPSessionCreate 994
- onHTTPSessionDestroy 994
- onHTTPSmoothStreamingPlayReadyCreateLive 995
- onHTTPSmoothStreamingPlayReadyCreateVOD 995
- onHTTPSmoothStreamingSessionCreate 996
- onHTTPSmoothStreamingSessionDestroy 996
- onHTTPStreamerRequest 617
- onHTTPStreamerSessionCreate 648
- onHTTPStreamerSessionDestroy 648
- onIndex 588
- onInit 588
- onInitAfterLoadTemplate 1748
- onInitBeforeLoadTemplate 1748
- onInitStart 1747
- onInitStop 1748
- onItem 896
- onlineTime 1155
- onLivePacket 1540
- onLiveStreamDvrRecorderCreate 1607
- onLiveStreamDvrRecorderDestroy 1607
- onLiveStreamDvrRecorderInit 1607
- onLiveStreamPacketizerCreate 1616
- onLiveStreamPacketizerDestroy 1616
- onLiveStreamPacketizerInit 1616
- onLiveStreamTranscoderCreate 1627
- onLiveStreamTranscoderDestroy 1627
- onLiveStreamTranscoderInit 1627
- onLog 732, 735
- onMediaCacheSourceCreate 890
- onMediaCacheSourceDestory 890
- onMediaCacheStoreCreate 894
- onMediaCacheStoreDestory 894
- onMediaCasterCreate 911
- onMediaCasterDestroy 911
- onMediaReaderClose 1473
- onMediaReaderCreate 1472
- onMediaReaderExtractMetaData 1473
- onMediaReaderInit 1472
- onMediaReaderOpen 1472
- onMediaStreamCreate 1547
- onMediaStreamDestroy 1547
- onMetaData 952, 954, 1534
- ONMETADATA_FILTER_DURATION 1658
- ONMETADATA_FILTER_NONE 1658
- onMetadataToSetDataFrame 1150
- onModuleLoad 980
- onModuleUnload 980
- onMP3EncodeInfo 954
- onNewVHost 1402
- onOpen 588
- onOptions 1245, 1264, 1312
- onPause 1245, 1264, 1312, 1532
- onPauseRaw 1534
- onPerSourceInfoAdd 871
- onPerSourceInfoRemove 871
- onPlay 1245, 1264, 1312, 1531
- onPlaylistItemStart 1633
- onPlaylistItemStop 1633
- onPublish 1532
- onPublishHandlerPlay 1128, 1208
- onPushPublisherSessionCreate 1129, 1208
- onPushPublisherSessionDestroy 1129, 1208
- onPushPublishSessionCreate 997, 1067, 1120
- onPushPublishSessionDestroy 997, 1067, 1121
- onRecord 1245, 1264, 1312
- onRedirect 1245, 1264, 1312
- onRegisterPlayer 911
- onRegisterStreamNameGroup 1751
- onRejectConnection 385
- onReleaseStream 1127, 1207
- onResetMediaCaster 916, 1045
- onResetStream 1752
- onResponseWriteStart 1371
- onResponseWriteStop 1371
- onResult 979, 1000
- onRTPSessionCreate 998, 1261
- onRTPSessionDestroy 998, 1261
- onSeek 1532
- onSegmentEnd 702
- onSegmentStart 701

onServerAfterVHostStartup 1391
onServerBeforeVHostStartup 1391
onServerConfigLoaded 1390
onServerCreate 1388
onServerInit 1388
onServerShutdownComplete 1389
onServerShutdownStart 1388
onSessionAudioDecodeCodecInfo 1751
onSessionAudioEncodeCodecInfo 1750
onSessionAudioEncodeCreate 1749
onSessionAudioEncodeInit 1749
onSessionAudioEncodeSetup 1750
onSessionClosed 1128, 1208
onSessionDataEncodeCreate 1749
onSessionDataEncodeInit 1750
onSessionDestinationCreate 1748
onSessionIdle 1128, 1208
onSessionOpened 1128, 1208
onSessionVideoDecodeCodecInfo 1751
onSessionVideoEncodeCodecInfo 1750
onSessionVideoEncodeCreate 1749
onSessionVideoEncodeInit 1749
onSessionVideoEncodeSetup 1750
onSetParameter 1245, 1263, 1312
onSetSourceStream 912
onSetup 1245, 1265, 1313
onSharedObjectConnect 1427
onSharedObjectCreate 1427
onSharedObjectDestroy 1427
onSharedObjectDisconnect 1428
onShutdownStart 1751
onShutdownStop 1752
onSlotDelete 1438
onSlotSetValue 1438
onSplitRecord 686
onSplitRecorder 701
onStartRecord 686
onStartRecorder 700
onStop 1532
onStopRecord 687
onStopRecorder 701
onStreamCreate 999, 1044, 1128, 1207
onStreamDestroy 999, 1044
onStreamOnPlayStatus 1128, 1207
onStreamOnStatus 1128, 1207
onStreamStart 914, 1542
onStreamStop 914
onSwitchRecord 687
onSwitchRecorder 701
onTeardown 1245, 1265, 1313
onTrim 952
onUnbind 565, 567
onUnPublish 1532
onUnRegisterPlayer 912
onUnregisterStreamNameGroup 1751
onValidateMediaCaster 915, 1044
onValidateMediaCastersStart 915, 1044
onValidateMediaCastersStop 915, 1045
onValidateSession 1129, 1208
onValidateSessionResult 1129, 1208
onVHostClientConnect 1846
onVHostCreate 1845
onVHostInit 1845
onVHostItemCreate 1844
onVHostItemDestroy 1844
onVHostItemUpdate 1844
onVHostShutdownComplete 1846
onVHostShutdownStart 1845
onVideoH264Packet 1539
onWriteComplete 1571
opaque 1190
open 8, 884, 1461, 1467, 1635
order 197
ORDERED 18
originalTimecodeThreshold 1074
outprint 1155
output 1257
OUTPUT_TYPE_AMF 1724
OUTPUT_TYPE_CEA608 1724
OUTPUT_TYPE_TTML 1724
outputFile 721
outputPath 721
OutputStream 28
OVERWRITE_FILE 708

P

p 1579
PACKET_TIME 486
packetComplete 1497

packetizerLicenses 1584
packetOutput 1774
packetsSentAudio 1153
packetsSentData 1153
packetsSentVideo 1153
packetTime 420
PARAM1 1005, 1193
PARAM10 1007
PARAM2 1006
PARAM3 1006
PARAM4 1006
PARAM5 1006
PARAM6 1006
PARAM7 1006
PARAM8 1006
PARAM9 1006
PARAMMETHODNAME 1005
parseAllowDomains 286
parseBodyForParams 574
parseCommaSeparatedLanguages 972
parseIdString 921
parseQueryStr 111
password 1073, 1193
PASSWORDFILEFORMAT_CLEAR 331
PASSWORDFILEFORMAT_UNKNOWN 331
pathToFileURL 109
pause 1028, 1346, 1553
PAUSE_PAUSE 1550
PAUSE_PLAY 1550
PAUSE_TOGGLE 1550
pauseRaw 1032, 1553
pauseRecording 538, 549, 1605
peekByte 132
PICTUREMAXFILESIZE 798
PICTURETYPE_ARTISTLOGO 798
PICTURETYPE_COVERBACK 798
PICTURETYPE_COVERFRONT 797
PICTURETYPE_FILEICON 797
PICTURETYPE_ILLUSTRATION 798
PICTURETYPE_MOVIESCREENCAPTURE 798
PICTURETYPE_OTHER 797
PICTURETYPE_OTHERFILEICON 797
PICTURETYPE_PUBLISHERLOGO 798
ping 364
play 1027, 1345, 1552, 1553, 1630, 1655, 1659, 1665, 1666, 1667
play2 1027
playDuration 967
PLAYEVENT_AFTERBUFFERFILL 1466
PLAYEVENT_AFTERMETADATA 1466
PLAYEVENT_BEFOREBUFFERFILL 1466
PLAYEVENT_BEFOREMETADATA 1466
PLAYEVENT_STARTPLAYBACK 1466
Playlist 1634
playlistAcrossSessions 1097
playlistCountFromMap 1097
playlistCountFromPacketizer 1097
playlistCrossName 1098
playlistEntry 1104
PlaylistItem 1637
playlistMap 1098
playlistName 1098
playlistStrA 1102
playlistStrB 1102
playlistTimeout 1097
playSeek 967
PLAYSIZES_AUDIO_BYTES 1551
PLAYSIZES_AUDIO_COUNT 1551
PLAYSIZES_DATA_BYTES 1551
PLAYSIZES_DATA_COUNT 1551
PLAYSIZES_LOSS_BYTES 1551
PLAYSIZES_LOSS_COUNT 1552
PLAYSIZES_SIZE 1552
PLAYSIZES_VIDEO_BYTES 1551
PLAYSIZES_VIDEO_COUNT 1551
playStart 967
PLAYSTATUSTYPE_COMPLETE 1551
PLAYSTATUSTYPE_STOP 1551
PLAYSTATUSTYPE_SWITCH 1550
PLAYTRANSITION_APPEND 1009
PLAYTRANSITION_APPEND_IMMEDIATE 1009
PLAYTRANSITION_RESET 1009
PLAYTRANSITION_RESET_IMMEDIATE 1009
PLAYTRANSITION_STOP 1009
PLAYTRANSITION_SWAP 1009
PLAYTRANSITION_SWITCH 1009
PLAYTRANSITION_UNKNOWN 1009
PLAYTRANSITIONSTR_APPEND 1008
PLAYTRANSITIONSTR_RESET 1008
PLAYTRANSITIONSTR_STOP 1008
PLAYTRANSITIONSTR_SWAP 1008

- PLAYTRANSITIONSTR_SWITCH 1008
- PLAYTRANSITIONSTR_UNKNOWN 1008
- pollInterval 1115
- port 1073
- postChunkAdded 527
- postChunksPurged 528
- posToBlock 880
- preChunkAdded 527
- preChunksPurged 527
- preload 880
- previous 1668
- profile 1117
- PROP_3GPP_CAPTION_CHARACTER_SET 1733
- PROP_AMF_CONVERTER_INSERT_ERASES_IN_GAPS 1735
- PROP_AMF_CONVERTER_TEXT_TYPE 1734
- PROP_AMF_CONVERTER_TRACK_INDEX 1734
- PROP_CAPTION_CHARACTER_SET 1732
- PROP_CAPTION_FILE_NAMING_RULE 1730
- PROP_CAPTION_FILENAME_QUERY_PARAMETER 1739
- PROP_CAPTION_LANGUAGE_QUERY_PARAMETER 1739
- PROP_CAPTION_LANGUAGES 1726
- PROP_CAPTION_PATH_NAMING_RULE 1731
- PROP_CEA608_CONVERTER_CHARACTER_SET 1733
- PROP_CEA608_CONVERTER_CHARS_PER_LINE 1733
- PROP_CEA608_CONVERTER_COLOR 1733
- PROP_CUPERTINO_CAPTION_LANGUAGES 1726
- PROP_CUPERTINO_LIVE_USE_WEBVTT 1734
- PROP_CUPERTINO_VOD_CAPTIONS_ENABLED 1725
- PROP_CUPERTINO_VOD_REMOVE_EXISTING_CEA608 1734
- PROP_CUPERTINO_VOD_USE_WEBVTT 1734
- PROP_DEBUG_3GPP_CAPTION_PARSER 1738
- PROP_DEBUG_3GPP_CAPTION_PARSER_TIME 1738
- PROP_DEBUG_CUPERTINO_VOD_CAPTION_LANGUAGE_SELECTION 1727
- PROP_DEBUG_CUPERTINO_VOD_CAPTION_PROVIDER_DETERMINATION 1728
- PROP_DEBUG_LIVE_CAPTION_INGEST 1740
- PROP_DEBUG_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608_LOG_AS_RAW 1743
- PROP_DEBUG_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608_LOG_AS_SCC 1742
- PROP_DEBUG_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608_VERBOSE 1742
- PROP_DEBUG_LIVE_CAPTION_ONCAPTION_INGEST 1740
- PROP_DEBUG_LIVE_CAPTION_ONTEXTDATA_INGEST 1740
- PROP_DEBUG_RTMP_VOD_CAPTION_LANGUAGE_SELECTION 1728
- PROP_DEBUG_RTMP_VOD_CAPTION_PROVIDER_DETERMINATION 1729
- PROP_DEBUG_SANJOSE_VOD_CAPTION_LANGUAGE_SELECTION 1728
- PROP_DEBUG_SANJOSE_VOD_CAPTION_PROVIDER_DETERMINATION 1729
- PROP_DEBUG_SCC_CAPTION_PARSER 1738
- PROP_DEBUG_SCC_CAPTION_PARSER_TIME 1738
- PROP_DEBUG_SCC_VOD_CAPTION_COMPANION_FILES 1738
- PROP_DEBUG_SRT_CAPTION_PARSER 1736
- PROP_DEBUG_SRT_CAPTION_PARSER_TIME 1737
- PROP_DEBUG_SRT_VOD_CAPTION_COMPANION_FILES 1736
- PROP_DEBUG_TTML_CAPTION_PARSER 1735
- PROP_DEBUG_TTML_CAPTION_PARSER_TIME 1736
- PROP_DEBUG_TTML_VOD_CAPTION_COMPANION_FILES 1735
- PROP_DEBUG_VOD_CAPTION_COMPANION_FILES 1729
- PROP_DEBUG_VOD_CAPTION_LANGUAGE_SELECTION 1727
- PROP_DEBUG_VOD_CAPTION_PROVIDER_DETERMINATION 1728
- PROP_DEBUG_WEBVTT_CAPTION_PARSER 1737
- PROP_DEBUG_WEBVTT_CAPTION_PARSER_TIME 1737
- PROP_DEBUG_WEBVTT_VOD_CAPTION_COMPANION_FILES 1737
- PROP_LIVE_CAPTION_DEFAULT_LANGUAGES 1740
- PROP_LIVE_CAPTION_INGEST_AMF_ID_STRING 1739
- PROP_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608_ENABLE_FIELD1 1741
- PROP_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608_ENABLE_FIELD2 1741
- PROP_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608_IDLE_TIME_TO_FLUSH_DECODER 1741
- PROP_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608

- _LANGUAGE_ID 1742
- PROP_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608
_MAX_WARN_MESSAGES 1742
- PROP_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608
_TRACK_NUMBER 1742
- PROP_LIVE_CAPTION_INGEST_ON_CAPTION_INFO_S
TRING 1741
- PROP_LIVE_CAPTION_INGEST_ON_CAPTION_STRING
1741
- PROP_LIVE_CAPTION_INGEST_TYPE 1740
- PROP_LIVE_CAPTION_USE_STREAM_NAME_GROUPS
1740
- PROP_MAXIMUM_CAPTION_DURATION 1731
- PROP_RTMP_CAPTION_LANGUAGES 1727
- PROP_RTMP_VOD_CAPTIONS_ENABLED 1725
- PROP_SANJOSE_CAPTION_LANGUAGES 1727
- PROP_SANJOSE_VOD_CAPTIONS_ENABLED 1726
- PROP_SRT_CAPTION_CHARACTER_SET 1732
- PROP_TTML_CAPTION_CHARACTER_SET 1732
- PROP_TTML_INLINE_TAGS_IN_TEXT 1736
- PROP_UNDEFINED_LANG_CODE 1730
- PROP_VOD_CAPTIONS_ENABLED 1725
- PROP_WEBVTT_CONVERTER_INSERT_ERASES_IN_GA
PS 1735
- properties 563, 958, 965, 975, 1072, 1299
- PROPERTY_ALLOWABLE_AV_PACKET_DELTA 463
- PROPERTY_APPEND_DISCONTINUITY_DELTA 458
- PROPERTY_ARCHIVE_STRATEGY 459
- PROPERTY_AUDIO_GROUP_COUNT 517
- PROPERTY_AUDIO_ONLY_CHUNK_TARGET_DURATION 465
- PROPERTY_BREAK_ON_PTS 466
- PROPERTY_CHUNK_CACHE_CLASS 462
- PROPERTY_CHUNK_DURATION_MINIMUM 467
- PROPERTY_CHUNK_GROUPING_SECONDS 458
- PROPERTY_CHUNK_MEMORY_CACHESIZE 467
- PROPERTY_CHUNK_READER_CLASS 460
- PROPERTY_CHUNK_WRITER_CLASS 460
- PROPERTY_CUPERTINO_ON_CHUNK_START_RESET_
COUNTER 481
- PROPERTY_CUPERTINO_PLAYLIST_ALLOW_CACHING
481
- PROPERTY_CUPERTINO_PLAYLIST_FORCE_LIVE 480
- PROPERTY_CUPERTINO_PLAYLIST_FORCE_NONLIVE
480
- PROPERTY_CUPERTINO_PLAYLIST_GZIP_THRESHOL
D 481
- PROPERTY_CUPERTINO_PLAYLIST_USE_GZIP 480
- PROPERTY_CUPERTINO_PUT_END_TAG_BEFORE_CH
UNKS 481
- PROPERTY_DEBUG_CHUNK_RETRIEVALS 473
- PROPERTY_DEBUG_CUPERTINO_PLAYER_ADAPTER
475
- PROPERTY_DEBUG_ENABLE 706
- PROPERTY_DEBUG_FAILED_CHUNK_RETRIEVALS
474
- PROPERTY_DEBUG_LOG_INVALID_CHUNK_DETAILS
470
- PROPERTY_DEBUG_LOG_INVALID_CHUNK_MATCHE
R 471
- PROPERTY_DEBUG_LOG_VALID_CHUNK_DETAILS
471
- PROPERTY_DEBUG_LOG_VALID_CHUNK_MATCHER
472
- PROPERTY_DEBUG_MAX_INVALID_CHUNKS_LOGGE
D 471
- PROPERTY_DEBUG_MAX_VALID_CHUNKS_LOGGED
472
- PROPERTY_DEBUG_MAXIMUM_RAW_PACKETS 473
- PROPERTY_DEBUG_MBR_ALIGNMENT 476
- PROPERTY_DEBUG_MBR_ALIGNMENT_RESOLUTION
476
- PROPERTY_DEBUG_MBR_DETAILS_WHEN_RECORD_
DNE 476
- PROPERTY_DEBUG_MBR_PLAYER_ADAPTER 476
- PROPERTY_DEBUG_MBR_PLAYER_ADAPTER_IF_SHIF
TED_EMPTY 476
- PROPERTY_DEBUG_MBR_RESOLVER 518
- PROPERTY_DEBUG_MBR_RESOLVER_GAP_DETAILS
518
- PROPERTY_DEBUG_MBR_RESOLVER_INTERSECTION_
DETAILS 518
- PROPERTY_DEBUG_METHODS 475
- PROPERTY_DEBUG_PLAYER_ADAPTER 475
- PROPERTY_DEBUG_PLAYLIST_REQUEST 477
- PROPERTY_DEBUG_RAW_PACKETS 472
- PROPERTY_DEBUG_RAW_PACKETS_MATCHER 473
- PROPERTY_DEBUG_REPEATERS 475
- PROPERTY_DEBUG_SANJOSE_PLAYER_ADAPTER 475
- PROPERTY_DEBUG_SMOOTH_PLAYER_ADAPTER 475

- PROPERTY_DEBUG_STATE_CHANGE 473
- PROPERTY_DEBUG_TOSSED_HOLDERS 472
- PROPERTY_ENCRYPTION_INFO_DELEGATE 477
- PROPERTY_FILE_FORMAT 707
- PROPERTY_FILE_SYSTEM_CLASS 461
- PROPERTY_FILE_VERSION_DELEGATE 711
- PROPERTY_FILE_VERSION_TEMPLATE 711
- PROPERTY_INTERCEPTOR_CHUNK_DELEGATE 512
- PROPERTY_MANIFEST_PERSISTER_CLASS 461
- PROPERTY_MANIFEST_PURGE_CACHE_SIZE 514
- PROPERTY_MAX_ALLOWABLE_CHUNK_DURATION 515
- PROPERTY_MAX_CHUNK_LOG 474
- PROPERTY_MAX_RECALC_DURATION_LOG 474
- PROPERTY_MAX_RECORDING_LENGTH 470
- PROPERTY_MAX_SAVE_HOLDER_SIZE 516
- PROPERTY_MBR_ALTERNATIVE_MATCH_DELTA 468
- PROPERTY_MBR_MINIMUM_PACKETTIME_GAP_SIZE 468
- PROPERTY_MBR_MINIMUM_UTCTIME_GAP_SIZE 469
- PROPERTY_MBR_USE_SIMPLEALIGNMENT 515
- PROPERTY_MBR_USE_UTC_FOR_ALIGNMENT 468
- PROPERTY_MEDIACACHE_ENABLED 470
- PROPERTY_MEDIACACHE_READER_CLASS 470
- PROPERTY_MINIMUM_AVAILABLE_CHUNKS 513
- PROPERTY_MOVE_FIRST_VIDEO_FRAME_TO_ZERO 709
- PROPERTY_NOTIFY_LISTENER 710
- PROPERTY_OUTPUT_FILE 710
- PROPERTY_OUTPUT_PATH 710
- PROPERTY_PACKET_DELTA_TO_NOTIFY 464
- PROPERTY_PACKET_DELTA_TO_RESET_TIME 464
- PROPERTY_PACKET_DURATION_MAXIMUM 469
- PROPERTY_PACKET_SORT_TIME 466
- PROPERTY_PLAYLIST_REQUEST_DELEGATE 477
- PROPERTY_PURGE_CONTROL_CLASS 514
- PROPERTY_RECALC_DURATION_MAX_AV_DIFFERENCE_TRIGGER 515
- PROPERTY_RECALC_DURATION_TRIGGER_SIZE 516
- PROPERTY_RECORD_AUDIO 465
- PROPERTY_RECORD_DATA 465, 709
- PROPERTY_RECORD_VIDEO 465
- PROPERTY_RECORDINGS_LOADER_CLASS 462
- PROPERTY_REPEATER_HEARTBEAT_DURATION 467
- PROPERTY_REPEATER_SHARED_SECRET 467
- PROPERTY_SANJOSE_ABST_DURATION_TOLERANCE 480
- PROPERTY_SANJOSE_ABST_TIMESCALE 479
- PROPERTY_SANJOSE_PLAYLIST_DELIVERYTYPE 477
- PROPERTY_SANJOSE_PLAYLIST_LIVE_STREAMTYPE 478
- PROPERTY_SANJOSE_PLAYLIST_MIMETYPE 477
- PROPERTY_SANJOSE_PLAYLIST_RECORDED_STREAMTYPE 478
- PROPERTY_SANJOSE_PLAYLIST_VERSION 478
- PROPERTY_SEGMENT_DURATION 709
- PROPERTY_SEGMENT_SCHEDULE 710
- PROPERTY_SEGMENT_SIZE 709
- PROPERTY_SEGMENTATION_TYPE 706
- PROPERTY_SMOOTH_MANIFEST_H264_CODEC 484
- PROPERTY_SMOOTH_MANIFEST_LIVE_CAN_PAUSE 482
- PROPERTY_SMOOTH_MANIFEST_LIVE_CAN_SEEK 481
- PROPERTY_SMOOTH_MANIFEST_MAJOR_VERSION 483
- PROPERTY_SMOOTH_MANIFEST_MINOR_VERSION 483
- PROPERTY_SMOOTH_MANIFEST_RECORDED_CAN_PAUSE 482
- PROPERTY_SMOOTH_MANIFEST_RECORDED_CAN_SEEK 482
- PROPERTY_SMOOTH_MANIFEST_RECORDED_SPECIFY_DURATION 483
- PROPERTY_SMOOTH_MANIFEST_VERBOSE_DURATION 484
- PROPERTY_SPLIT_ON_TC_DISCONTINUITY 709
- PROPERTY_START_ON_KEYFRAME 708
- PROPERTY_START_RECORDING_ON_STARTUP 465
- PROPERTY_STORAGE_DIRECTORY 459
- PROPERTY_STREAM_DATA 484
- PROPERTY_TEXT_READER_CLASS 463
- PROPERTY_TEXT_WRITER_CLASS 463
- PROPERTY_USE_SIMPLE_FILE_VERSION_NAMING 706
- PROPERTY_UTC_ALIGN_TO_AUDIO 513
- PROPERTY_UTC_DELEGATE_CLASS 513
- PROPERTY_UTC_USE_WOWZA_PACKET_ARRIVAL 513
- PROPERTY_VERIFY_CHUNK_EXISTENCE_ON_MANIFEST_LOAD 514
- PROPERTY_VERSIONING_OPTION 708

PROPERTY_WAIT_FOR_CODEC_TIME 517
PROPERTY_WINDOW_DURATION 459
PROPS_ITEM_LASTMODIFIED 876
PROPS_ITEM_LENGTH 876
PROPS_ITEM_MEDIANAME 876
PROPS_SOURCE_BLOCKSIZE 877
PROPS_SOURCE_NAME 877
PROTOCOLUSAGE_CUPERTINO 266
PROTOCOLUSAGE_MC_LIVEREPEATER 271
PROTOCOLUSAGE_MC_RTP 271
PROTOCOLUSAGE_MC_SHOUTCAST 271
PROTOCOLUSAGE_MEDIACACHE 268
PROTOCOLUSAGE_MPEGDASH 267
PROTOCOLUSAGE_ORIGINCUPERTINO 267
PROTOCOLUSAGE_ORIGINMPEGDASH 268
PROTOCOLUSAGE_ORIGINSANJOSE 267
PROTOCOLUSAGE_ORIGINSMOOTH 267
PROTOCOLUSAGE_PUSHPUBLISHING 268
PROTOCOLUSAGE_RTMP 265
PROTOCOLUSAGE_RTMP_IN 268
PROTOCOLUSAGE_RTMP_OUT 268
PROTOCOLUSAGE_RTMPE 266
PROTOCOLUSAGE_RTMPE_IN 269
PROTOCOLUSAGE_RTMPE_OUT 269
PROTOCOLUSAGE_RTMPS 266
PROTOCOLUSAGE_RTMPS_IN 268
PROTOCOLUSAGE_RTMPS_OUT 268
PROTOCOLUSAGE_RTMPT 265
PROTOCOLUSAGE_RTMPT_IN 268
PROTOCOLUSAGE_RTMPT_OUT 268
PROTOCOLUSAGE_RTMPTE 266
PROTOCOLUSAGE_RTMPTE_IN 269
PROTOCOLUSAGE_RTMPTE_OUT 269
PROTOCOLUSAGE_RTMPTS 266
PROTOCOLUSAGE_RTMPTS_IN 269
PROTOCOLUSAGE_RTMPTS_OUT 269
PROTOCOLUSAGE_RTP 266
PROTOCOLUSAGE_RTP_IN 269
PROTOCOLUSAGE_RTP_OUT 269
PROTOCOLUSAGE_RTSP_IN 270
PROTOCOLUSAGE_RTSP_OUT 270
PROTOCOLUSAGE_RTSP_PULL_IN 271
PROTOCOLUSAGE_RTSP_PULL_OUT 271
PROTOCOLUSAGE_SANJOSE 266
PROTOCOLUSAGE_SMOOTH 266
PROTOCOLUSAGE_TOTAL 271
PROTOCOLUSAGE_WEBM 267
PROTOCOLUSAGE_WOWZ 267
PROTOCOLUSAGE_WOWZ_IN 270
PROTOCOLUSAGE_WOWZ_OUT 270
PROTOCOLUSAGE_WOWZE 267
PROTOCOLUSAGE_WOWZE_IN 270
PROTOCOLUSAGE_WOWZE_OUT 270
PROTOCOLUSAGE_WOWZS 267
PROTOCOLUSAGE_WOWZS_IN 270
PROTOCOLUSAGE_WOWZS_OUT 270
PROTO_HTTPCUPERTINO 781
PROTO_HTTPDVRCHUNK 782
PROTO_HTTPMPEGDASH 782
PROTO_HTTPSANJOSE 781
PROTO_HTTPSCUPERTINO 782
PROTO_HTTPSDVRCHUNK 782
PROTO_HTTPSMOOTH 781
PROTO_HTTPSSANJOSE 782
PROTO_HTTPSSMOOTH 782
PROTO_HTTPSSTREAMER 782
PROTO_HTTPSTREAMER 781
PROTO_RTMP 780
PROTO_RTMPE 780
PROTO_RTMPS 780
PROTO_RTMPT 780
PROTO_RTMPTE 780
PROTO_RTMPTS 780
PROTO_RTSP 781
PROTO_WOWZ 781
PROTO_WOWZE 781
PROTO_WOWZS 781
ptToDvr 559
publish 1027, 1511, 1579, 1645, 1646
publisher 1193
PublishingProviderBase 1652
PublishingProviderLive 1655
PublishingProviderMediaReader 1658
publishStream 1202
purge 1425, 1443
purgeEntries 496, 550
purgeManifestEntries 539
PUSHPUBLISH_TYPE_RTMP 1052
PUSHPUBLISH_TYPE_RTP 1053
PUSHPUBLISH_TYPE_UNKNOWN 1052

PushPublishBase 1074
PushPublishContext 1083
PushPublishHTTPAkamaiBase 1090
PushPublishHTTPBase 1098
PushPublishHTTPOutputItem 1103
PushPublishHTTPOutputItemChunk 1104
pushPublishHTTPPlaylistDebug 1098
PushPublishHTTPPlaylistEntry 1106
PushPublishHTTPPlaylistList 1108
PushPublishHTTPPlaylistMap 1110
PushPublishHTTPPusher 1112
pushPublishImpl 1117
PushPublishRTMP 1156
PushPublishRTMPAkamaiHandler 1186
PushPublishRTMPAuthProviderAdobe 1191
PushPublishRTMPAuthProviderBase 1193
PushPublishRTMPAuthProviderLimelight 1198
PushPublishRTMPNetConnectionSession 1202
PushPublishRTMPNotifyBase 1206
PushPublishRTMPProfileHandler 1215
PushPublishRTMPUstream 1225
PushPublishRTP 1234
PushPublishRTPRunner 1242
PushPublishRTSPActionNotify 1244
PUSHPUBLISHSESSION_TYPE_RTMP 1065
PUSHPUBLISHSESSION_TYPE_RTP 1065
PUSHPUBLISHSESSION_TYPE_UNKNOWN 1064
PushPublishSessionBase 1117
PushPublishSessionHTTPBase 1119
PushPublishSessionNotifyBase 1120
PushPublishSessionRTMP 1227
PushPublishSessionRTP 1247
pushPublishStreamDebug 1072
pushPublishType 1072
put 198, 199, 231, 232, 1430
putAll 325
putAllExtraData 227
putAudioPacketizerItem 1268, 1316
putCrossSession 1083
putDvrRecorder 1527
putExtraData 227
putFileInfo 637
putFrame 792
putGlobalLogValue 756
putHTTPStreamerAdapter 1838

putLiveStreamTranscoder 1528
putMetaData 1570
putProfileData 1083
putRTSPSessionExtraLine 1364
putRTSPTunnelingSession 1273
putSlot 1419, 1444
putStreamNameParts 636
putStreamPacketizerItem 1269, 1317
putVideoPacketizerItem 1269, 1317

Q

qopValue 1197
QUERY_PARAM_DVR 508
queryStr 1300
QUERYSTR_CAPTION_FILENAME 1739
QUERYSTR_CAPTIONLANGUAGES 1739

R

ranges 428
read 9, 25, 861, 882, 883
READ_ACCESS_ALL 356
READ_ACCESS_NONE 356
READACCESS 1442, 1488
readAhead 878
readAppConfig 246
readAppInstConfig 302
readChunk 63
readChunkHeader 63
readConfig 1382, 1406
readHeader 64
readPrevChunkHeader 63
readWriteTimeout 1098
readXMLConfig 1406
realm 1197
receiveAudio 1030
RECEIVER_NAME_KEY 728
receiveVideo 1030
reconnectWaitTime 1072
record 1345
recordData 719
RECORDER_STATE_ERROR 711
RECORDER_STATE_ERROR_STRING 712
RECORDER_STATE_PENDING 711

RECORDER_STATE_PENDING_STRING 712
RECORDER_STATE_RECORDING 712
RECORDER_STATE_RECORDING_STRING 713
RECORDER_STATE_WAITING 712
RECORDER_STATE_WAITING_STRING 712
recordFactory 685
recordingPaused 523
recordingReset 524
recordingResumed 523
recordingStarted 523
recordingStopped 524
redirectConnection 362
redirectSession 644, 1301, 1318
redirectSessionCode 1301
redirectSessionMessage 1302
redirectSessionURL 1302
redundant 1088
referrer 1300
refreshManifest 494
registerCallback 1512
registerLiveStreamTranscoder 1408
registerOnPlayStatus 1512
registerOnStatus 1512
registerPlayer 904, 930, 941
registerPlayRTPSession 293
rejectConnection 343, 361
rejectSession 635, 1302
release 931, 940, 1420, 1444
releaseMediaCaster 945
releaseSessionId 1321
releaseSocketAddress 1270
releaseStream 1026
releaseUDPPortPair 1271
reloadConfig 1862
reloadVHostConfig 1376, 1400
remove 142, 178, 200, 233, 941, 1431, 1866
removeAcceptorListener 1826
removeActionListener 1311
removeAppInstance 246
removeApplicationInstanceListener 244
removeApplicationListener 1825, 1866
removeCallbackListener 1529
removeClient 1420, 1432, 1444, 1821
removeClientListener 276, 1513
removeConnectionListener 342
removeCrossSession 1083
removeDefaultAppInstance 1152
removeDvrChunkListener 551
removeDvrRecorder 1527, 1594
removeDvrRecorderListener 313
removeDvrRecordingListener 540
removeDvrStoreListener 539
removeDvrStreamManagerListener 314
removeFrame 792
removeFromPlaylist 1666
removeGlobalLogValue 756
removeGlobalLogValues 755
removeHeader 579
removeHTTPListener 317
removeHTTPStreamerSession 301
removeIdleWorkerListener 1832
removeItem 1635
removeListener 684, 698, 1164, 1668
removeLivePacketListener 1530
removeLiveStreamPacketizer 1592, 1593
removeLiveStreamPacketizerListener 309
removeLiveStreamTranscoder 1528
removeLiveStreamTranscoderListener 310
removeMediaCasterListener 279
removeMediaReaderListener 306
removeMediaStreamListener 277, 1592
removeMediaWriterListener 304
removeMetaDataListener 953
removeModuleListener 283
removeNameGroup 1593, 1594
removePendingWrites 879
removePlayStreamByName 281
removePublisher 298
removePushPublishSession 318
removePushPublishSessionListener 318
removePushPublishSessions 318
removeRendition 976
removeRTPListener 316
removeRTPSession 293
removeRTSPStream 1303
removeRTSPTunnelingSession 1272
removeSDPFiles 1234
removeSegment 959, 960
removeServerListener 1379, 1403
removeSession 1158, 1322

removeSessionListener 1323
removeSharedObjectListener 278, 1432
removeSlotListener 1424, 1449
removeStream 1590
removeStreamDomainStr 638
removeStreamStore 541
removeTrack 1339
removeTrackInternal 1339
removeVHostItemListener 1862
removeVideoH264SEILListener 1529
renditionIndex 1104
renditions 975
reparentClient 379, 1835
ReplaceItem 107
requestFilters 563
requiresLayout 739
reset 545, 930, 959, 1552, 1848, 1858
resetConnection 1053, 1092, 1159, 1235
resetMDC 755
resetMediaCasterStream 302
resetNewPacketizer 1063, 1112
resetNoLookup 1552
resetOnTimecodeOutOfOrder 1152
resetSentMetadataFlag 1347
resetStream 507, 537, 652, 944, 1602, 1613, 1622
resetTimecodes 1557
resolveHostName 1159
resolveMediaList 1459
resolvePlayAlias 1543, 1544, 1545
resolveStreamAlias 1543, 1545
ResponseFunction 1250, 1251
ResponseFunctions 1256
resumeEncoding 29
resumeRecording 538, 549, 1605
retrieveRawChunk 442, 522
reuseSockets 1097
REVERSE 8
rewind 1468
rtmpAuthProvider 1155
rtmpDebugLog 1148
RTMPENABLED 1370
RTPContext 1268
RTPDestination 1275
RTPPort 1282
RTPPushPublishSession 1287

rtpPushPublishSession 1233
RTPSession 1302
rtpSession 329
RTPSessions 1321
RTPStream 1338
RTPUtils 1766
rtpWriteListener 1299
rtspPlayRangeStart 1301
rtspPlayRangeStop 1301
rtspTunnelingSessionId 1297
run 1115, 1242, 1665

S

salt 1190
SANJOSE_F4M_STREAMINGTYPE_STREAMING 479
SANJOSE_F4M_STREAMTYPE_DVR 478
SANJOSE_F4M_STREAMTYPE_LIVE 479
SANJOSE_F4M_STREAMTYPE_LIVEORRECORDED 479
SANJOSE_F4M_STREAMTYPE_RECORDED 479
SANJOSE_F4M_VERSION_1_0 479
SANJOSE_F4M_VERSION_2_0 479
SANJOSE_MANIFEST 615
SANJOSE_MEDIA 615
saveBlock 878
SDPFILEEXTENSION 1233
SDPLOCATION_AUDIO 1335
SDPLOCATION_STREAM 1335
SDPLOCATION_VIDEO 1335
secureTokenSharedSecret 1151
seek 9, 883, 1028, 1468, 1553, 1631, 1655, 1659
SEEK_EXACT 1467
SEEK_KEYCLOSE 1467
SEEK_KEYDOWN 1467
SEEK_KEYUP 1466
SEEKTARGET_AUDIO 1466
SEEKTARGET_ENHANCED 1466
SEEKTARGET_VIDEOKEYFRAME 1466
SEGMENT_BY_DURATION 707
SEGMENT_BY_SCHEDULE 707
SEGMENT_BY_SIZE 707
SEGMENT_NONE 707
segmentationType 718
segmentDuration 720
segments 958

segmentSchedule 720
segmentSize 720
SEI_PAYLOADTYPE 1686
SEI_STARTCODE 1693
SEI_USERDATATYPECODE 1687
send 1424, 1448, 1508
sendAdaptiveGroupPlaylist 1059
sendAMF3 1508
sendClientOnStatusError 1017
sendControlBytes 1517
sendDirect 1507, 1508
sendDirectAMF3 1507
sendFCAnnounce 1150
sendFCPublish 1149
sendingPerformance 1072
sendInternal 1448
sendLivePlaySeek 1498
sendLivePlayStart 1497
sendLivePlaySwitch 1498
sendMessage 1284
sendOnMetadata 506, 1150, 1652
sendOriginalTimecodes 1074, 1150
sendOutput 1063
sendPacketizerWarning 1115
sendPauseNotify 1503
sendPlayReset 1499
sendPlaySeek 1500
sendPlayStart 1500, 1501, 1502
sendPlayStatus 1502, 1503
sendPlayStop 1499
sendPlaySwitch 1501
sendReleaseStream 1150
sendResponse 1284
sendResult 1015, 1016
sendStreamCloseCommands 1150
sendStreamCloseCommandsSendTime 1152
sendStreamNotFound 1497
sendStreamOnStatusError 1017
SendToBackupServer 1186
sendToBackupServer 1088
sendUnpauseNotify 1503, 1504
sendVODPlaySwitch 1498
sendZeroLengthPacket 1470
serialize 136, 149, 155, 172, 173, 185, 190, 202, 203, 238, 400, 404, 411, 414, 422, 427, 429, 433, 435, 488, 503, 793, 818, 859
SERIALIZE_CURRENT_VERSION 420
serializeBody 799, 823, 826, 829, 832, 834, 839, 845, 848, 851, 854, 859
serializeDate 120
serializeFooter 793
serializeHeader 793
serializeInt 120
serializePacket 227
serializer 197
serializeString 122, 817
serializeStringLen 817
serializeStringNoLength 122
serializeTags 793
serializeZeroLengthString 121
Server 1398
SERVER_DEFAULT 1376
serverIp 1300
serverPort 1300
service 603
session 1202
sessionClosed 905, 1159, 1285
sessionId 1297
sessionOpened 905, 1159, 1285
sessionOrder 1149
SESSIONPROTOCOL_COUNT 627
SESSIONPROTOCOL_CUPERTINOSTREAMING 626
SESSIONPROTOCOL_DIRECTSTREAMING 627
SESSIONPROTOCOL_DVRCHUNKSTREAMING 626
SESSIONPROTOCOL_MPEGDASHSTREAMING 626
SESSIONPROTOCOL_SANJOSESTREAMING 626
SESSIONPROTOCOL_SMOOTHSTREAMING 626
SESSIONPROTOCOL_UNKNOWN 626
SESSIONPROTOCOL_WEBMSTREAMING 626
sessionStart 947
sessionStop 947
SESSIONTYPE_LIVE 625
SESSIONTYPE_LIVEDVR 626
SESSIONTYPE_UNKNOWN 625
SESSIONTYPE_VOD 625
set 145, 146, 181, 182
setAbsTimecode 225
setAbsTimecodeLong 213
setAbsTimecodeShort 213
setAcceptConnection 274, 379
setAcceptConnectionDescription 378

setAcceptConnectionExObj 378
setAcceptConnectionObj 378
setAcceptConnectionSendClientId 363
setAcceptSession 635
setAcceptWOWZConnections 1383, 1408
setActive 628
setAdaptiveStreaming 1076
setAddressCount 1286
setAdminInterfaceHostPort 1824
setAkamaiConfigName 1092
setAkamaiEventName 1092
setAkamaiHostId 1092
setAkamaiOriginIp 1163, 1187
setAkamaiOriginPort 1163, 1187
setAkamaiPassword 1163, 1188
setAkamaiPlaybackURLPrefix 1093
setAkamaiStreamId 1091, 1187
setAkamaiUserName 1162, 1188
setAlgorithm 398
setAllowDomains 286
setAnnounce 1307
setAppend 1504
setAppInstance 400, 631, 903, 1078, 1171, 1235, 1306, 1623
setAppInstanceName 1348
setAppInstanceProperty 1048
setApplication 739, 1220
setApplicationInstanceTouchTimeout 305
setApplicationProperty 1047
setApplicationTimeout 283, 1828
setAppName 1348
setAudioCodecId 970
setAudioHost 1237, 1278
setAudioPort 1237, 1277
setAudioSize 1490
setAudioTC 1493, 1494
setAuthenticatePlayHandler 1305
setAuthenticatePublishHandler 1305
setAuthenticationMethod 565, 570
setAuthMode 1199
setAVSyncMethod 1351
setBandwidthLimit 1027
setBaseClass 887, 920, 1850
setBasePath 868
setBindAddress 1180
setBitrate 649
setBitrateAudio 969
setBitrateTotal 968
setBitrateVideo 969
setBitrateVideoPeak 969
setBlockMask 892
setBlockUDPOut 1286, 1366
setBody 1253
setBuffer 598
setBufferSize 835
setBufferTime 368, 1028, 1489, 1554
setBuilder 1110
setByteContainerLevel 214
setCacheControlStatus4xx 611
setCanHandleFilter 869
setCanPlay 548
setCanRecord 547
setCheckIpAddr 1351
setCheckSSRC 1350
setChecksum 1457
setChunkCounter 216
setChunkGroupDuration 441
setClassName 203, 206, 238
setClient 330, 337, 1497
setClientIdleFrequency 287, 1830
setClientProperty 1048
setClientTimeout 1824
setClustered 1515
setCodecInfoAudio 507, 652
setCodecInfoVideo 507, 652
setCommandInterface 1402
setCommandInterfaceHostPort 1378, 1402
setConce 1199
setConfigDir 1858
setConnected 1308
setConnectionFlashVerion 1161
setConnectionFlashVersion 1161, 1216
setConnectionInfoPageURL 1217
setConnectionInfoQueryString 1216
setConnectionInfoSwfURL 1216
setConnectionLimit 1859
setConnectionPageURL 1162
setConnectionQueryStr 1160
setConnectionStatus 1240
setConnectionSwfURL 1161
setConnectionTimeout 862, 866, 920, 1100, 1156, 1216

setConnectionValidator 342
setConnectLastAttempt 1169, 1217
setConnectLastSuccess 1170, 1217
setContentTypes 841
setCookieStr 635, 1315
setCoreHandlerPoolSize 1380, 1404, 1823
setCoreTransportPoolSize 1379, 1403, 1823
setCurrentDvrTime 520
setData 827, 830, 1435
setDataBuffer 223
setDataSize 1491
setDataTC 1494, 1495
setDataTypes 1495
setDebug 1782
setDebugAppTimeout 316
setDebugLog 88, 1054, 1079, 1171
setDebugPackets 1176, 1220
setDebugRTSPSession 1314
setDefaultBlockSize 888
setDefaultStreamingStore 535
setDefaultStreamPrefix 1386, 1411
setDescribe 1307
setDescription 800, 827, 849, 855, 887, 920, 1852
setDestinationAddress 862, 867
setDirection 1454
setDirecton 10
setDomainName 1788
setDstAppInstanceName 1173, 1221
setDstApplicationName 1173, 1220
setDstStreamName 1054, 1078, 1172, 1221
setDuration 1631, 1656, 1659
setDvrRecorder 1527
setDvrRecorderId 1603
setDvrRecorderList 313, 1523
setDvrSessionInfo 640
setDynamic 206
setDynamicLogProperties 1381, 1404, 1835
setEmbeddedFlag 835
setEncKeyBytes 395
setEncKeyBytesList 396
setEncMethod 1573
setEncoding 739
setEncRepeaterBytes 399
setEncryptions 411
setError 1781
setErrorField 1781
setEventHandler 1782
setEventStr 1781
setExt 1519
setFakePosts 1100
setFastPlay 1035
setFastPlaySettings 365, 1509
setField 1780
setFileExtension 1646
setFileIOPoolSize 1823
setFlags 818, 859
setFlashVer 357
setForceAMF0 1254
setForceMPEGTSOut 1366
setForceRTSPInterleaved 1357
setFps 1453
setFrame 786
setFrameCount 1761
setFrameIdentifier 827
setFrames 788
setHandler 1284
setHasEncryption 548
setHeader 579
setHeaderSize 1521
setHeadResponse 580
setHeight 971
setHost 1054, 1079, 1236, 1276, 1342
setHostname 1054, 1079, 1172
setHostType 1238, 1278
setHowToPublish 1157
setHTTPHeaderServer 1387, 1411
setHTTPIdent2Response 1792
setHTTPIdentResponse 1792
setHTTPOrigin 645
setHTTPOriginMode 609
setHTTPStreamerAdapter 631
setHTTPStreamerItem 604
setHTTPStreamerList 296
setHTTPStreamerMaxPathLen 1841
setHTTPStreamerRequestValidator 1840
setHTTPStreamerSession 1462, 1524
setHttpWriterDebug 1100
setID 605
setId 211
setIdleCheckFrequency 1831

setIdleFrequency 372, 1313
setIdleHandler 1309
setIdleMinimumWaitTime 1834
setIdleWorkerCount 1831
setIdStr 818, 858
setImplementation 1075
setIndex 1638
setInitiateWOWZConnections 1383, 1408
setInnerObj 207
setIntData 158
setInternalRecorder 699
setIntHeader 579
setIOPerformanceCounter 1315
setIoPerformanceCounter 1401
setIoSession 1319
setIp 1307
setIpAddress 632, 1789
setIsPlaying 1490
setIVBytes 397
setKeepAliveTime 921
setKeepAliveTimeout 1832
setKeyRotationInterval 396
setKeyRotationType 396
setKID 397
setKIDsList 397
setLanguage 824, 840, 972
setLastAccessed 1457
setLastAuthenticateMethod 1306
setLastClientId 1437
setLastModified 881
setLastOperation 1437
setLastRun 1781, 1782
setLastSentAbsTimecode 215
setLastValidateTime 376
setLen 786, 788
setLength 881, 893
setLive 1360
setLivePlaylist 673
setLiveRepeaterCapabilities 380
setLiveStreamDvrRecorderControl 312
setLiveStreamingPacketizer 631
setLiveStreamPacketizer 932, 1029, 1524
setLiveStreamPacketizerControl 302
setLiveStreamPacketizerId 1613
setLiveStreamPacketizerList 296, 380, 1522
setLiveStreamRepeater 932, 1525
setLiveStreamTranscoderControl 311
setLiveStreamTranscoderItem 1624
setLiveStreamTranscoderList 311, 380, 1523
setLocal 881
setLocalBindAddress 1221
setLocalPath 881
setLock 968, 977
setLoggedConnect 1306
setLongTimecode 215
setMapFileIndex 1053, 1080
setMaxHTTPSessionsPerTCPSession 611
setMaximumPendingReadBytes 285, 1829
setMaximumPendingWriteBytes 284, 1829
setMaximumSetBufferTime 285, 1830
setMaxRetries 862, 1101
setMaxRTCPWaitTime 1352
setMaxStorageDirDepth 295
setMaxTimeToLive 880, 889
setMediaCache 888
setMediaCacheItem 867, 884
setMediaCacheSource 877
setMediaCasterDef 904
setMediaCasterHostPortConfig 926
setMediaCasterId 903
setMediaCasterItem 1541
setMediaCasterPlay 1526
setMediaCasterProcessorCount 926
setMediacasterRTPRTSPRTPTransportMode 315
setMediaCasterType 908
setMediaCasterValidator 304
setMediaList 977
setMediaListProvider 315
setMediaListSegment 968
setMediaNameExpression 869
setMediaReaderItem 1461, 1467
setMediaWriterItem 1569
setMergeOnMetadata 1527
setMessageBytes 1252
setMetaDataProvider 1520
setMethodValue 1199
setMimeType 800
setMinTimeToLive 880, 889
setMode 1356
setMoveToNextIfLiveStreamMissing 1671

setMPEGTSAudioBitrate 1366
setMPEGTSAudioLanguage 1352
setMPEGTSAudioPID 1353
setMPEGTSOut 1359
setMPEGTSProgramID 1353
setMPEGTSVideoBitrate 1366
setMPEGTSVideoPID 1353
setMultiplier 1453
setName 243, 272, 870, 887, 919, 960, 970, 1279, 1423, 1436, 1448, 1492, 1554, 1793, 1850, 1858
setNameGroupRemoveDelayTime 1595
setNCCounter 1200
setNetConnection 1203, 1496
setNetConnectionIdleFrequency 1831
setNetConnectionProcessorCount 1825
setNew 213
setNonce 1199
setObjectEncoding 159, 163, 215, 378
setOffset 599
setOffsetToNextTag 835
setOnMetadataFilter 1660
setOnMetadataToSetDataFrame 1174, 1218
setOpaqueObject 884
setOpen 1509
setOriginalTimecodeThreshold 1076, 1077, 1170, 1218
setOutHost 1355
setOwnerIdentifier 830
setPacketizeAudio 1614
setPacketizeData 1615
setPacketizeVideo 1614
setPacketType 599
setParent 1450, 1569
setPassThru 889
setPassword 1077, 1124, 1187, 1194
setPathDelimiter 869
setPersistent 1421, 1431, 1445
setPictureData 800
setPictureDataAsURL 800
setPictureType 800
setPingTimeout 284, 1828
setPlay 1496
setPlayClass 1851
setPlayDuration 641, 971
setPlayer 1505
setPlaylistAcrossSessions 1099
setPlaylistCountFromMap 1099
setPlaylistCountFromPacketizer 1099
setPlaylistTimeout 1099
setPlayLogged 630
setPlaySeek 642, 972
setPlayStart 641, 971
setPollingInterval 1668
setPort 740, 866, 1054, 1055, 1079, 1172, 1236, 1789
setProcessorCount 1790
setProfile 1065, 1118
setProperties 564, 567, 603, 888, 892, 1461, 1471, 1859
setProperty 323, 919, 1422, 1423, 1445, 1446, 1851
setProxyAddress 862, 865
setPsshData 405
setPublishAudioCodecId 1525
setPublishData 674
setPublishDataEvents 1651
setPublisherStream 1530
setPublishStream 1203
setPublishVideoCodecId 1525
setPushPublishImpl 1065, 1118
setQOPValue 1199
setQueryStr 634, 1310, 1518
setRange 426
setReadAhead 889
setReadAheadThreshold 889
setReaderClass 888
setReadRetries 866
setReadTimeout 862, 865
setReadWriteTimeout 1100
setRealm 1198
setRealTimeStartTime 1631, 1656, 1659
setReceiveAudio 1509
setReceiveBufferSize 862, 866
setReceiveVideo 1510
setReceiveVideoFPS 1510
setReconnectWaitTime 907, 1055, 1080, 1217
setRecord 1495
setRecordAudio 1604
setRecordData 1604
setRecordingName 1605
setRecordingStreamStore 535
setRecordVideo 1604
setRedirectSession 643, 1317
setRedirectSessionBody 644

setRedirectSessionCode 643, 1318
setRedirectSessionContentType 644
setRedirectSessionMessage 1318
setRedirectSessionURL 643, 1318
setRedundant 1091
setReferrer 634, 1310
setRefuseIncomingUDPConnections 1367
setRefuseOutgoingUDPConnections 1367
setRemoteHost 739
setRemoveDefaultAppInstance 1174, 1218
setRepeat 1635, 1667
setRepeaterEdge 1614
setRepeaterMediaCasterDef 608
setRepeaterOriginUrl 285, 376, 1031
setRepeaterQueryString 286
setRequest 616
setRequestFilters 564, 570
setRequestFullURL 862, 865
setRequestType 616
setResetOnTimecodeOutOfOrder 1176, 1218
setResetPlayStream 1345
setResponseCode 580
setResponseOnIncompleteChunk 611
setRestDocumentationServerEnabled 1409
setRestServerEnabled 1385, 1408
setResyncAudioVideoOnSR 1347
setRetAMFNumber 1253
setReuseSockets 1100
setRsoStorageDir 288
setRTMPAuthProvider 1173
setRTMPHeaderServer 1386, 1411
setRTPAVSyncMethod 292
setRTPDestination 1364
setRTPIIdleFrequency 287, 1831
setRTPIgnoreProfileLevelId 1365
setRTPIgnoreSPropParameterSets 1366
setRTPMaxRTCPWaitTime 292
setRTPPlayAuthenticationMethod 291
setRTPPublishAuthenticationMethod 291
setRTPPushPublishSession 1235
setRTPSession 330, 337, 1288
setRTPStream 1522
setRTPStreamContext 1367
setRtpStreamWaitTimeout 1239
setRTPWrapped 1238, 1278
setRTSP 1353
setRTSPAlwaysUseSDPPorts 1364
setRTSPBindIpAddress 307, 1361
setRTSPConnectionAddressType 308, 1362
setRTSPConnectionIpAddress 307, 1361
setRTSPMaximumPendingWriteBytes 306, 1360
setRTSPMaxPathLen 1841
setRTSPOriginAddressType 308, 1362
setRTSPOriginIpAddress 308, 1362
setRTSPPlayRangeStart 1314
setRTSPPlayRangeStop 1314
setRTSPPull 1354
setRTSPRequestValidator 1841
setRTSPSessionDescription 1358
setRTSPSessionName 1358
setRTSPSessionTimeout 305, 1360
setRTSPTunnelingSessionId 1314
setSample 599
setSDPData 1288
setSDPLang 1365
setSecure 647
setSecureTokenSharedSecret 1162, 1217
setSendBufferSize 862, 866
setSendCloseOnRejectConnection 382
setSendFCAnnounce 1164, 1217
setSendFCPublish 1157, 1216
setSendOnMetadata 1173, 1218, 1631, 1653, 1669
setSendOriginalTimecodes 1076, 1170, 1217
setSendPlayStopLogEvent 1519
setSendPublishStopLogEvent 1520
setSendRecordStopLogEvent 1519
setSendReleaseStream 1157, 1216
setSendSDESEvents 1359
setSendStreamCloseCommands 1174, 1218
setSendToBackupServer 1090
setSeq 225
setSerializer 203
setServer 1866
setServerIp 633, 1309
setServerPort 633, 1309
setSession 1202, 1341
setSessionId 627, 1303
setSessionProtocol 605, 632
setSessionTimeout 629
setSessionTracker 678

setSessionType 631
setSessionValid 1305
setSharedObjectReadAccess 290, 372
setSharedObjectWriteAccess 291, 372
setShortDescription 824
setShutdownClient 379, 1315
setShutdownOnRelease 931
setShuttingDown 1835
setSingleStreamAV 1236
setSize 211, 222, 599, 971
setSlotValue 1446, 1447
setSlotVersion 1436
setSourcePath 887
setSoVersion 1436
setSpeed 1660
setSrc 212, 224, 1251, 1491
setSrcStream 1055, 1081
setSrcStreamBufferSize 1179
setSrcStreamName 1054, 1078, 1171, 1235
setSSL 1180, 1219
setSSLConfig 1792
setSslFactoryClass 1790
setStartLiveOnPreviousBufferTime 1670
setStartLiveOnPreviousKeyFrame 1670
setStartOnPreviousBufferTime 1656
setStartOnPreviousKeyFrame 1656
setStartRecordingOnStartup 1606
setStartStartupStreams 1839
setStartTC 1454
setStartTCOffset 1454
setStartupStreamsDelayTime 1842
setStopIfStreamMissing 1656
setStorageDir 1424, 1432, 1449
setStream 629, 692, 903, 1065, 1118
setStreamAngle 1188
setStreamAudioSampleAccess 289, 374
setStreamBandWidth 1188
setStreamCloseWaitTime 1175, 1218
setStreamDomainProtectionActive 609
setStreamExt 636, 931, 1348
setStreamFileMapper 295
setStreamId 1187
setStreamKeyDir 287
setStreamName 636, 1054, 1078, 1349, 1588, 1623
setStreamNameAliasProvider 298
setStreamNameLogging 1349
setStreamPort 1277
setStreamPosition 637, 1471
setStreamProperty 1048
setStreamQueryStr 1354
setStreamReadAccess 289, 374
setStreamRunning 908
setStreamSessionId 1349
setStreamSessionIp 1350
setStreamSessionVersion 1350
setStreamsToPublish 1203
setStreamStorageDir 287
setStreamTimeout 905
setStreamType 274, 360, 918, 1029, 1347, 1507, 1645
setStreamTypeStr 609
setStreamVersionHandler 536
setStreamVideoSampleAccess 289, 373
setStreamWriteAccess 290, 375
setSuspended 1791
setSwitchLog 1669
setSyncPacketType 1181
setSyncTimecode 1181
setSystemId 402
setTarget 1780
setTargetEncoding 163
setText 824
setTextEncoding 799, 824, 840, 846, 849
setThreadContext 381
setTimecode 212, 224, 843, 1254
setTimecodes 224
setTimecodesOutOfOrderThreshold 1176, 1218
setTimeOffsetBetweenItems 1670
setTimeoutSession 629
setTimesInMilliseconds 1669
setTimeStampFormat 840
setTimeToLive 1238
setTitle 961, 973
setTrack 1282
setTranscodeResult 1528
setTranscoderName 1623
setTranscoderType 1625
setTranscoderVODDestination 649
setTranscoderVODIndex 646
setTranscoderVODSession 645
setTransportMode 1357

- setTTL 1278, 1285
- setType 132, 212, 223, 970, 1254
- setTypeStr 1792
- setUDPManagedDeliveryCount 1365
- setUDPManagedDeliveryDelay 1365
- setUnidentifiedSessionTimeout 1842
- setUnpublishOnEnd 1671
- setUri 634, 1310
- setURL 402, 827, 852, 855
- setUserAgent 633, 1304
- setUserAgents 1379, 1403
- setUserHTTPHeader 610, 642
- setUserName 1077, 1124, 1187, 1194
- setUserQueryStr 639
- setUseUstreamPingPong 1226
- setValid 929
- setValidateFMLEConnections 310, 381
- setValidationFrequency 284, 1164, 1217, 1829
- setValue 843, 846, 849
- setVersion 1421, 1445
- setVersionFile 1569
- setVHost 329, 337, 604, 627, 1304
- setVideoCodecId 969
- setVideoHost 1237, 1277
- setVideoPort 1238, 1276
- setVideoSize 1490
- setVideoTC 1494
- setVisited 1859
- setVODLastTimeTC 1363
- setVODPlayLen 1363
- setVODSpeed 1671
- setVODStartTimeTC 1363
- setVODTimedTextProviderList 297
- setVODTranscodeNGRP 646, 972
- setWaitForVideoKeyFrame 1569
- setWaitOnMetadataAudio 1178, 1219
- setWaitOnMetadataAudioDatarate 1179, 1219
- setWaitOnMetadataAvailable 1177, 1219
- setWaitOnMetadataTimeout 1178, 1219
- setWaitOnMetadataVideo 1178, 1219
- setWaitOnMetadataVideoDatarate 1178, 1219
- setWidth 970
- setWowzaAudioOnly 970
- setWowzaKeyFrameOnly 972
- setWowzaVideoOnly 972
- setWowzSession 382
- SharedObject 1442, 1443
- SHARED_OBJECT_CMD_CONNECT 1417
- SHARED_OBJECT_CMD_CONNECTSUCCESS 1418
- SHARED_OBJECT_CMD_DELETE 1418
- SHARED_OBJECT_CMD_DISCONNECT 1417
- SHARED_OBJECT_CMD_ERROR 1418
- SHARED_OBJECT_CMD_SEND 1418
- SHARED_OBJECT_CMD_SETVALUE 1417
- SHARED_OBJECT_STATUS_CHANGE 1418
- SHARED_OBJECT_STATUS_CLEAR 1418
- SHARED_OBJECT_STATUS_DELETE 1418
- SHARED_OBJECT_STATUS_SUCCESS 1418
- shortDescription 823
- shortValue 171
- shouldDeleteArchivedStream 408, 555
- shouldDvrRecord 1609
- shouldLoadArchivedStream 408, 555
- shouldLoadStream 525
- shouldLoadStreamVersion 526
- shouldStartRecordingOnStartup 1606
- shutdown 242, 272, 545, 584, 628, 658, 665, 672, 678, 904, 930, 937, 1268, 1283, 1304, 1344, 1493, 1552, 1602, 1612, 1623, 1820, 1866
- shutdownAppInstance 246
- shutdownApplication 1835
- shutdownClient 271, 359, 1301
- shutdownRTPSession 1273
- shutdownSession 604
- shutdownStream 944
- sinfo 1578
- singleStreamAV 1233
- size 143, 153, 178, 198, 1420, 1430, 1443, 1518, 1555, 1791, 1795
- skipByte 133
- sloppyGetTrack 1339
- SMIL_PARAM_IS_WOWZA_CAPTION_STREAM 1743
- SMIL_PARAM_WOWZA_CAPTION_INGEST_TYPE 1743
- SMOOTHSTREAMING_AUDIO_FRAGMENT 615
- SMOOTHSTREAMING_DATA_FRAGMENT 615
- SMOOTHSTREAMING_MANIFEST 615
- SMOOTHSTREAMING_VIDEO_FRAGMENT 615
- socketAddressToUniqueStr 97
- SOURCE_TYPE_3GPP 1724
- SOURCE_TYPE_AMF 1723
- SOURCE_TYPE_SCC 1724

SOURCE_TYPE_SRT 1724
SOURCE_TYPE_TTML 1723
SOURCE_TYPE_WEBVT 1724
splitCookie 76
splitOnTcdDiscontinuity 719
splitPragmas 75
splitQueryStr 75, 1194
splitRecorder 691
splitRecording 683
srcStream 1073
srcStreamBufferSize 1154
srcStreamName 1073
stackTraceToSingleLineString 46
stackTraceToString 46
stampToString 101
stampToStringNoUnits 101
start 107, 1399, 1780
startApplicationInstance 1837
startAudioData 1648
startAudioPacket 1522
startChunk 506, 651
STARTCODE_3DI 791
STARTCODE_ID3 790
startCommandInterface 1376, 1399
startDataData 1650
startDataPacket 1522
startMediaCasterStream 303
startOnKeyFrame 719
startOutputRunner 1112
startPlay 1555
startPlayback 1470
startPublishing 1516
startRecorder 690
startRecording 537, 549, 682, 683, 1604
startRTPPull 1767, 1768
startServer 1399
startStartupStreams 1839
startStats 1410
startStream 934, 1602, 1613
startsWith 40
startTransmit 1033
startVHost 1377, 1400
startVHosts 1377, 1401
startVideoData 1647
startVideoPacket 1522
STAT_connect_application_not_available 776
STAT_connect_application_not_found 777
STAT_connect_bad_gateway 778
STAT_connect_internal_error 777
STAT_connect_license_limit 777
STAT_connect_pending_wating 776
STAT_connect_redirect 777
STAT_connect_rejected_by_application 777
STAT_connect_rejected_by_module 777
STAT_connect_resource_limit 777
STAT_connect_service_unavailable 778
STAT_connect_successful 776
STAT_connect_unknown_protocol 777
STAT_general_internal_error 780
STAT_general_successful 780
STAT_play_bad_request 778
STAT_play_internal_error 779
STAT_play_rejected_by_application 778
STAT_play_rejected_by_module 778
STAT_play_stream_not_found 778
STAT_play_successful 778
STAT_play_unsupported_media_type 778
STAT_publish_bad_request 779
STAT_publish_in_use 779
STAT_publish_internal_error 779
STAT_publish_rejected_by_application 779
STAT_publish_successful 779
STAT_publish_unsupported_media_type 779
STAT_stop_client_disconnect 779
STAT_stop_successful 779
stateChange 541
statusCodeToStr 77
stop 1780
stopAdminAgent 1382, 1399
stopCommandInterface 1376, 1399
stopMediaCasterStream 303
stopName 1493, 1554
stopPublishing 1516
stopRecorder 691
stopRecording 537, 549, 683, 1605
stopRTPPull 1767
stopServer 1399
stopStartupStreams 1839
stopStats 1411
stopStream 934

stopTransmit 1033
stopVHost 1376, 1400
stopVHosts 1377, 1400
storeChunks 539, 550
storeOnMetadata 539, 550
Stream 1663
stream 1117
streamCloseWaitTime 1152
streamCodecToString 60
streamExists 934, 1343
streamId 1215
STREAMINFO_SESSIONATTRIBUTES 1337
STREAMINFO_SESSIONBANDWIDTH 1337
STREAMINFO_SESSIONCONNECTIONDATA 1337
STREAMINFO_SESSIONEMAILADDRESS 1337
STREAMINFO_SESSIONINFORMATION 1336
STREAMINFO_SESSIONNAME 1336
STREAMINFO_SESSIONPHONENUMBER 1337
STREAMINFO_SESSIONPORIGIN 1336
STREAMINFO_SESSIONPROTOCOLVERSION 1336
STREAMINFO_SESSIONREPEATTIMES 1337
STREAMINFO_SESSIONTIMEZONES 1337
STREAMINFO_SESSIONTIMING 1337
STREAMINFO_SESSIONURI 1336
StreamItem 1850
streamLicenses 1584
StreamList 1853
streamLock 1585
streamMonitorAudioStartTimeout 1043
streamMonitorAudioTCNegTolerance 1042
streamMonitorAudioTCPosTolerance 1042
streamMonitorAudioTCToleranceEnable 1042
streamMonitorAudioTimeout 1043
streamMonitorAVSyncTolerance 1043
streamMonitorAVSyncToleranceEnable 1043
streamMonitorDataTCNegTolerance 1042
streamMonitorDataTCPosTolerance 1042
streamMonitorDataTCToleranceEnable 1042
streamMonitorDebug 1043
streamMonitorResetNameGroups 1043
streamMonitorStreamStartTimeout 1041
streamMonitorStreamTimeout 1041
streamMonitorVideoStartTimeout 1043
streamMonitorVideoTCNegTolerance 1042
streamMonitorVideoTCPosTolerance 1042
streamMonitorVideoTCToleranceEnable 1042
streamMonitorVideoTimeout 1043
streamNames 1584
streamNamesLock 1585
streamNameToGroup 1584
streamNameToValidFilename 51
streamPacketizers 1301
StreamRecorderFileVersionDelegate 716
StreamRecorderParameters 722
StreamRecorderSimpleFileVersionDelegate 724
streams 1297, 1584
streamsOrder 1297
streamSrcToMediaCaster 941
streamsToPublish 1202
STREAMTIMEOUTREASON_GOOD 901
STREAMTIMEOUTREASON_MISSING 902
STREAMTIMEOUTREASON_NORTSPSESSION 901
STREAMTIMEOUTREASON_NOSESSION 901
STREAMTIMEOUTREASON_NOSTREAM 901
STREAMTIMEOUTREASON_NOTIMEOUT 901
STREAMTIMEOUTREASON_NOURL 901
STREAMTIMEOUTREASON_RECONNECTRUNNING 901
STREAMTIMEOUTREASON_UNKNOWN 901
streamToFileForRead 1537
streamToFileForWrite 1538
streamToIndex 1588
StreamUtils 1772
stringToMilliseconds 101
StringUtils 100
supportsKeyGeneration 403
suspendAllHostPorts 1836
suspendAllVHosts 1381, 1400
suspendCommandInterface 1399
suspendEncoding 29
suspendServer 1382, 1400
swapKIDBytes 397
switchName 1492, 1554
switchPlay 1555
switchRecorder 690
switchSetupToMPEGTS 1356
syncPacketType 1155
syncTimecode 1155
SystemUtils 104

T

TAG_APIC 809
TAG_COMM 816
TAG_LINK 809
TAG_PRIV 809
TAG_RBUF 809
TAG_SYLT 809
TAG_TALB 809
TAG_TBPM 810
TAG_TCOM 810
TAG_TCON 810
TAG_TCOP 810
TAG_TDEN 810
TAG_TDLY 810
TAG_TDOR 810
TAG_TDRC 810
TAG_TDRL 811
TAG_TDTG 811
TAG_TENC 811
TAG_TEXT 811
TAG_TFLT 811
TAG_TIPL 811
TAG_TTT1 811
TAG_TTT2 811
TAG_TTT3 811
TAG_TKEY 812
TAG_TLAN 812
TAG_TLEN 812
TAG_TMCL 812
TAG_TMED 812
TAG_TMOO 812
TAG_TOAL 812
TAG_TOFN 812
TAG_TOLY 813
TAG_TOPE 813
TAG_TOWN 813
TAG_TPE1 813
TAG_TPE2 813
TAG_TPE3 813
TAG_TPE4 813
TAG_TPOS 813
TAG_TPRO 814
TAG_TPUB 814
TAG_TRCK 814
TAG_TRSN 814
TAG_TRSO 814
TAG_TSOA 814
TAG_TSOP 814
TAG_TSOT 814
TAG_TSRC 814
TAG_TSSE 815
TAG_TSST 815
TAG_TXXX 816
TAG_UNKN 809
TAG_WCOM 815
TAG_WCOP 815
TAG_WOAF 815
TAG_WOAR 815
TAG_WOAS 815
TAG_WORS 816
TAG_WPAY 816
TAG_WPUB 816
TAG_WXXX 816
TAGS_TEXTINFORMATION 815
TEMP_CONSOLE_APPENDER_NAME 729
TEMP_LIST_APPENDER_NAME 729
terminate 1856
testFlashVersion 377
testNextByte 132
text 823
textEncoding 822
TEXTENCODING_DEFAULT 808
TEXTENCODING_ISO_8859_1 808
TEXTENCODING_URL 808
TEXTENCODING_UTF16 808
TEXTENCODING_UTF16BE 808
TEXTENCODING_UTF8 808
textEncodingToCharsetName 816
ThreadPool 1855
tiggerReadAhead 880
TIME_MAP_TYPE 493
timecodesOutOfOrderThreshold 1150
timeCreated 1299
TIMED_TEXT_TYPE_CAPTION 1723
timeReset 524
TIMESTAMP_RULE_FORMAT 728
TIMESTAMPFORMAT_MILLISECONDS 839
TIMESTAMPFORMAT_MPEG 839
title 958

toArray 154
toByteBuffer 154
toDvr 559
toHex 44
toLong 45
toNTPTime 105
toSMILString 959, 967, 976
toString 150, 156, 172, 186, 190, 203, 212, 224, 325, 400, 412, 414, 427, 430, 433, 435, 892, 921, 959, 967, 975, 1275, 1638, 1790, 1852, 1859
toStringList 102
totalIOPerformance2Last 1299
totalIOPerformanceLast 1299
totalPacketsSent 1153
touch 305, 359, 627, 1304, 1361, 1458, 1603, 1614
touchApplicationInstance 1836
toValidFilename 51
trait 197
TRANSCODERTYPE_LIVE 1621
TRANSCODERTYPE_VOD 1621
transportFindBestMatch 1359
traverseDirectory 52
triggerAMF3Switch 137
trim 1511, 1579
trimTrailingZero 817
truncatePacket 222
tryConnect 1150
tryToConnect 1159
type 132, 420, 966, 1117
TYPE_ADMIN 1788
TYPE_STREAMING 1788

U

UDPAppender 738
unbind 1283
unbindAllHostPorts 1836
unbindAllVHosts 1382, 1400
uncacheRTPStream 1271
unlock 877, 1426, 1450
unlockRepeaterStreams 1343
unpublish 1646
unregisterCallback 1511
unregisterLiveStreamTranscoder 1408
unregisterOnPlayStatus 1512
unregisterOnStatus 1512
unregisterPlayer 904, 930, 941
updateBlockMask 892
updateIOPerformance 1401
updateKey 396
updateKeyInfo 404
updateLoggingDuration 1403, 1518, 1828
updateLoggingValues 585, 632, 659, 666, 673, 1310, 1518, 1555
updateOnCuePointTimecode 71
updateSDPDestination 1766
uri 1300
URL_SAFE 18
urlToId 110
URLUtils 109
userAgent 1298
userExists 336
userName 1073, 1193
UTC_TIME 486
utcTime 420
utcToDvr 559
UTF8_BRITISH_POUND 1710
UTF8_BULLET 1687
UTF8_CENTS 1710
UTF8_CLOSING_DOUBLE_QUOTE 1687
UTF8_CLOSING_GUILLEMETS 1687
UTF8_CONNECTING_VERTICAL_BAR 1687
UTF8_COPYRIGHT 1687
UTF8_CURRENCY 1687
UTF8_DASH 1702
UTF8_DEGREE 1687
UTF8_DOUBLEQUOTE 1702
UTF8_EM_DASH 1687
UTF8_EXCLAMATION 1702
UTF8_GREATERTHAN 1703
UTF8_INVERTED_EXCLAMATION 1688
UTF8_LEFT_SINGLE_QUOTATION 1688
UTF8_LEFTPAREN 1702
UTF8_LESSTHAN 1702
UTF8_LOWER_A 1703
UTF8_LOWER_A_WITH_CIRCUMFLEX 1709
UTF8_LOWER_A_WITH_DIAERESIS 1688
UTF8_LOWER_A_WITH_GRAVE 1710
UTF8_LOWER_A_WITH_RING 1689
UTF8_LOWER_A_WITH_TILDE 1689
UTF8_LOWER_B 1703

UTF8_LOWER_C 1703	UTF8_PERIOD 1701
UTF8_LOWER_D 1703	UTF8_PIPE 1703
UTF8_LOWER_E 1703	UTF8_QUESTION 1702
UTF8_LOWER_E_WITH_CIRCUMFLEX 1709	UTF8_REGISTERED_MARK 1688
UTF8_LOWER_E_WITH_DIAERESIS 1689	UTF8_REVERSE_QUERY 1711
UTF8_LOWER_E_WITH_GRAVE 1710	UTF8_RIGHTPAREN 1702
UTF8_LOWER_F 1704	UTF8_SERVICE_MARK 1688
UTF8_LOWER_G 1704	UTF8_SINGLEQUOTE 1701
UTF8_LOWER_H 1704	UTF8_SPACE 1702
UTF8_LOWER_I 1704	UTF8_TRADEMARK 1711
UTF8_LOWER_I_WITH_CIRCUMFLEX 1709	UTF8_UNDERSCORE 1702
UTF8_LOWER_I_WITH_DIAERESIS 1689	UTF8_UPPER_A 1706
UTF8_LOWER_I_WITH_GRAVE 1689	UTF8_UPPER_A_WITH_ACUTE 1690
UTF8_LOWER_J 1704	UTF8_UPPER_A_WITH_CURCUMFLEX 1690
UTF8_LOWER_K 1704	UTF8_UPPER_A_WITH_DIAERESIS 1690
UTF8_LOWER_L 1704	UTF8_UPPER_A_WITH_GRAVE 1690
UTF8_LOWER_M 1704	UTF8_UPPER_A_WITH_RING 1690
UTF8_LOWER_N 1705	UTF8_UPPER_A_WITH_TILDE 1690
UTF8_LOWER_O 1705	UTF8_UPPER_B 1706
UTF8_LOWER_O_WITH_CIRCUMFLEX 1710	UTF8_UPPER_C 1706
UTF8_LOWER_O_WITH_DIAERESIS 1689	UTF8_UPPER_C_WITH_CEDILLA 1691
UTF8_LOWER_O_WITH_GRAVE 1689	UTF8_UPPER_D 1706
UTF8_LOWER_O_WITH_STROKE 1689	UTF8_UPPER_E 1707
UTF8_LOWER_O_WITH_TILDE 1690	UTF8_UPPER_E_WITH_ACUTE 1691
UTF8_LOWER_P 1705	UTF8_UPPER_E_WITH_CIRCUMFLEX 1691
UTF8_LOWER_Q 1705	UTF8_UPPER_E_WITH_DIAERESIS 1691
UTF8_LOWER_R 1705	UTF8_UPPER_E_WITH_GRAVE 1691
UTF8_LOWER_S 1705	UTF8_UPPER_F 1707
UTF8_LOWER_T 1705	UTF8_UPPER_G 1707
UTF8_LOWER_U 1705	UTF8_UPPER_H 1707
UTF8_LOWER_U_WITH_CIRCUMFLEX 1710	UTF8_UPPER_I 1707
UTF8_LOWER_U_WITH_DIAERESIS 1690	UTF8_UPPER_I_WITH_CIRCUMFLEX 1691
UTF8_LOWER_U_WITH_GRAVE 1690	UTF8_UPPER_I_WITH_DIAERESIS 1691
UTF8_LOWER_V 1705	UTF8_UPPER_I_WITH_GRAVE 1691
UTF8_LOWER_W 1706	UTF8_UPPER_I_WITH_TILDE 1692
UTF8_LOWER_X 1706	UTF8_UPPER_J 1707
UTF8_LOWER_Y 1706	UTF8_UPPER_K 1707
UTF8_LOWER_Z 1706	UTF8_UPPER_L 1707
UTF8_MINUSHYPHEN 1703	UTF8_UPPER_M 1708
UTF8_MUSICAL_NOTE 1710	UTF8_UPPER_N 1708
UTF8_NBSP 1710	UTF8_UPPER_O 1708
UTF8_NOTMATCH 1701	UTF8_UPPER_O_WITH_ACUTE 1692
UTF8_ONE_HALF 1711	UTF8_UPPER_O_WITH_CIRCUMFLEX 1692
UTF8_OPENING_DOUBLE_QUOTE 1688	UTF8_UPPER_O_WITH_DIAERESIS 1692
UTF8_OPENING_GUILLEMET'S 1688	UTF8_UPPER_O_WITH_GRAVE 1692

UTF8_UPPER_O_WITH_STROKE 1692

UTF8_UPPER_O_WITH_TILDE 1692

UTF8_UPPER_P 1708

UTF8_UPPER_Q 1708

UTF8_UPPER_R 1708

UTF8_UPPER_S 1708

UTF8_UPPER_T 1708

UTF8_UPPER_U 1708

UTF8_UPPER_U_WITH_ACUTE 1692

UTF8_UPPER_U_WITH_CIRCUMFLEX 1693

UTF8_UPPER_U_WITH_DIAERESIS 1693

UTF8_UPPER_U_WITH_GRAVE 1693

UTF8_UPPER_V 1709

UTF8_UPPER_W 1709

UTF8_UPPER_X 1709

UTF8_UPPER_Y 1709

UTF8_UPPER_Z 1709

UTF8_YEN 1688

UTF8MAP 1711

UTF8MAP_INDEX_OFFSET 1711

V

validate 878

validateHTTPStreamerRequest 1843

validateNewConnection 342

validateRTSPRequest 1847

validationFrequency 1151

validStreamDomainToString 638

valueOf 101, 486

values 486

VERSION_FILE 708

versionFile 52

versioningOption 718

vhost 329, 1297

VHOST_DEFAULT 1810

VHostItem 1858

VHostList 1862

VHostSingleton 1865

videoCodecId 966

videoCodecStringToId 61

videoCodecToMetaDataString 61

videoCodecToString 61

videoCodecTypeToString 95

videoHost 1233

videoPacketizers 1301

videoPort 1233

VIDEOSAMPLE_ACCESS_ALL 356

VIDEOSAMPLE_ACCESS_NONE 356

VIDEOSAMPLEACCESS 1489

vodTranscodeNGRP 967

W

waitOnMetadataAudio 1154

waitOnMetadataAudioDatarate 1154

waitOnMetadataAvailable 1154

waitOnMetadataReady 1154

waitOnMetadataStarttime 1154

waitOnMetadataTimeout 1154

waitOnMetadataVideo 1154

waitOnMetadataVideoDatarate 1154

warn 751

WEEKDAY 1779

width 967

WMSLogger 744

WMSLoggerFactory 754

WMSLoggerIDs 783

WMSProperties 323

wowzaAudioOnly 966

wowzaKeyFrameOnly 966

wowzaVideoOnly 966

WOWZDEBUGHEADERSIZE 210

wrap 154

write 29, 1254, 1255

WRITE_ACCESS_ALL 357

WRITE_ACCESS_NONE 357

WRITEACCESS 1442, 1489

writeAppConfig 247

writeAppInstConfig 302

writeChunk 65

writeCodecConfig 1769

writeConfig 1382, 1406

writeDeleteError 1425, 1447

writeDuration 68

writeError 1447

writeGeneratedKeyFrame 1469

writeHeader 64

writeHeaderFlags 72

writePackets 67, 68, 1468, 1469, 1568

writeSDPFiles 1234

writeSetValueError 1424, 1447

writeShortHeader 64

writeString 164

writeXMLConfig 1406

X

XMLUtils 113

Y

YEAR 1779