



Wowza Media Server® 3

Server Side API

Wowza Media Server 3: Server Side API



Copyright © 2006 – 2012 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 any warranties of any kind, either express or implied, regarding the information contained herein.

Third Party Information

This document contains links to third party websites that are not under the control of Wowza Media Systems, LLC (“Wowza”) and Wowza is not responsible for the content on any linked site. If you access a third party website mentioned in this document, then you do so at your own risk. Wowza provides these links only as a convenience, and the inclusion of any link does not imply that Wowza endorses or accepts any responsibility for the content on third party sites.

This document also refers to other third party software that is not licensed, sold, distributed or otherwise endorsed by Wowza. Please ensure that any and all use of Wowza® software and third party software is properly licensed.

Trademarks

Wowza, Wowza Media Systems, Wowza Media Server and related logos are either registered trademarks or trademarks of Wowza Media System, LLC in the United States and/or other countries.

Adobe and Flash are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Microsoft and Silverlight are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

QuickTime, iPhone, iPad and iPod are either registered trademarks or trademarks of Apple, Inc. in the United States and/or other countries.

Other product names, logos, designs, titles, words or phrases mentioned may be third party registered trademarks or trademarks in the United States and/or other countries.

Third party trademarks are used solely to identify and describe third party products as being compatible with Wowza products. Wowza is in no way sponsored, endorsed by or otherwise affiliated with any such third party trademark owners.

Third Party Copyright Notices

Log4j and Mina: Copyright © 2006, The Apache Software Foundation

Java Service Wrapper: Copyright © 1999, 2006, Tanuki Software, Inc.

Silver Egg Technology: Copyright © 2001, Silver Egg Technology

Java ID3 Tag Library and JLayer 1.0 (classic): Copyright © 1991, 1999, Free Software Foundation, Inc.

Bouncy Castle Crypto API: Copyright © 2000 – 2008, The Legion Of The Bouncy Castle

Apache Commons Lang libraries and Modeler libraries: Copyright © 2001-2008, The Apache Software Foundation

WebM VP8 Codec libraries: Copyright © 2010, Google Inc. All rights reserved.

Vorbis/Ogg libraries: Copyright © 2011, Xiph.org Foundation

Libgcc s-4 library and Libstdc++ library: Copyright © 2011, Free Software Foundation, Inc.

Speex Codec: Copyright © 2002-2003, Jean-Marc Valin/Xiph.org Foundation

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	<u>open()</u> Open the media asset
int	<u>read</u> (byte[] buf, int off, int size) Read bytes from the media asset
void	<u>seek</u> (long pos) Seek to a position in the media asset
void	<u>setDirecton</u> (int directon) 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 boolean</code>	<u>decodeToFile</u> (String dataToDecode, String filename) Convenience method for decoding data to a file.
<code>static Object</code>	<u>decodeToObject</u> (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 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.

(continued from last page)

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/rfcc-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)
```


(continued from last page)

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)
```

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
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)`

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](#)

(continued from last page)

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)
```

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

(continued from last page)

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.

(continued from last page)

Parameters:

infile - Input file
outfile - Output file

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

```

java.lang.Object
  |
  +- java.io.OutputStream
        |
        +- java.io.FilterOutputStream
              |
              +- com.wowza.util.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 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

[startsWith](#)(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:

bytes - byte array

offset - offset

len - length

(continued from last page)

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

byteArrayToLong

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

Convert byte array to long

Parameters:
b - byte array (8 bytes)

Returns:
long value

(continued from last page)

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:

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:

(continued from last page)

int value

byteArrayToInt

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

Conver 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.

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)

(continued from last page)

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.

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

(continued from last page)

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

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:

(continued from last page)

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:

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)
```

(continued from last page)

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	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	formatSeconds (long t) Give a time in seconds, return a String representing the time in hh:mm:ss.

static String	formatUtcTime (long utcTime)
static String	stackTraceToString (Throwable e)
static String	toHex (byte value) Format a byte value to a 0xff format
static String	toHex (int value) Format a byte value to a 0xffffffff format
static String	toLong (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)

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.

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)
```

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

(continued from last page)

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)
```

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

java.lang.Object

└─com.wowza.util.FileUtils

public class **FileUtils**
extends Object

FileUtils: File utilities

Constructor Summary

public	FileUtils()
--------	-----------------------------

Method Summary

static void	copyFile (java.io.File fromFile, java.io.File toFile) Simple file copy routine
static void	copyFile2 (java.io.File in, java.io.File out)
static boolean	deleteDirectory (java.io.File path)
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**()

(continued from last page)

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

copyFile2

```
public static void copyFile2(java.io.File in,  
                             java.io.File out)  
    throws java.io.IOException
```

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

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_BUFFERSIZE</u> Size of temporary buffer needed for flv reading (byte[]) Value: 13
public static final	<u>FLV_CHUNKHEADER_FIRSTBYTE</u> Header values: first byte of packet data Value: 3
public static final	<u>FLV_CHUNKHEADER_HEADERSIZE</u> Size of packet header (byte[]) Value: 11
public static final	<u>FLV_CHUNKHEADER_ISIZE</u> Header values: packet size Value: 1
public static final	<u>FLV_CHUNKHEADER_IMECODE</u> Header values: timecode (milliseconds) Value: 2
public static final	<u>FLV_CHUNKHEADER_ITYPE</u> Header values: packet type Value: 0
public static final	<u>FLV_CHUNKHEADER_SECONDBYTE</u> Header values: second byte of packet data Value: 4
public static final	<u>FLV_CHUNKHEADER_VALUESIZE</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 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>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 <u>AMFPacket</u>	<u>readChunk</u> (java.io.DataInput is) Read a packets worth of .flv data from an InputStream and return as an AMFPacket
static <u>AMFPacket</u>	<u>readChunk</u> (java.io.InputStream is) Read a packets worth of .flv data from an InputStream and return as an AMFPacket

static void	<u>readChunkHeader</u> (java.io.RandomAccessFile is, byte[] buffer, int[] values) Read packet header.
static boolean	<u>readHeader</u> (java.io.DataInput is) Read file header.
static boolean	<u>readHeader</u> (java.io.InputStream is) Read file header.
static void	<u>readPrevChunkHeader</u> (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	<u>streamCodecToString</u> (int codec) Get a printable string representation of the stream codecs defined as IVHost.CODEC_STREAM_*
static <u>AMFPacket</u>	<u>updateOnCuePointTimecode</u> (<u>AMFPacket</u> packet, long timecode)
static java.nio.ByteBuffer	<u>updateOnCuePointTimecode</u> (java.nio.ByteBuffer data, int dataType, long timecode)
static int	<u>videoCodecStringToId</u> (String codecString) Parse a string to get the codec ID defined by IVHost.CODEC_VIDEO_*
static String	<u>videoCodecToMetaDataString</u> (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	<u>videoCodecToString</u> (int codec) Get a printable string representation of the video codecs defined as IVHost.CODEC_VIDEO_*
static void	<u>writeChunk</u> (java.io.DataOutput ds, java.nio.ByteBuffer data, int size, long timecode, byte type)
static void	<u>writeChunk</u> (java.io.OutputStream ds, java.nio.ByteBuffer data, int size, long timecode, byte type) Write a packets worth of data.
static void	<u>writeDuration</u> (java.io.File file, double duration) Write the duration to an existing .flv file.
static void	<u>writeHeader</u> (java.io.OutputStream ds, double duration, int audiocodecid, int videocodecid, String createdBy, java.util.Map extraMetadata) Write file header including onMetaData packet.
static void	<u>writeHeader</u> (java.io.OutputStream ds, double duration, java.util.Map extraMetadata) Write file header including onMetaData packet.
static void	<u>writePackets</u> (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_IMECODE

```
public static final int FLV_CHUNKHEADER_IMECODE
```

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**

(continued from last page)

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
```

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

FLV_TCINDEXAUDIO

```
public static final int FLV_TCINDEXAUDIO
```

(continued from last page)

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

(continued from last page)

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_*

Returns:codec name

frameTypeToString

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

(continued from last page)

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

(continued from last page)

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.

Parameters:

is - RandomAccessFile

buffer - temporary buffer byte[FLV_CHUNKHEADER_BUFFERSIZE]

values - header values long[FLV_CHUNKHEADER_VALUESIZE]

(continued from last page)

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 void 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.CONTENT_TYPE_*

writeChunk

```
public static void 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(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:

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

(continued from last page)

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)
```

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:

(continued from last page)

packet - packet

Returns:

true if onMetaData packet

getOnMetadataData

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

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 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	splitQueryStr (String queryStr) Split a query string into a map and URL decode the values

static String

[statusCodeToStr](#)(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

Parameters:

`inUrl` - url (will work with http and https)

(continued from last page)

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:

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

com.wowza.util Interface IBandwidthThrottler

public interface **IBandwidthThrottler**
extends

Method Summary

long	<u>getBytesAllocation</u> (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.

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

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
----------------	---

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

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	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

(continued from last page)

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:

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:

(continued from last page)

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)

equals

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

Checks if 2 strings are equals, accounting for null cases.

Parameters:

s1 - string one

s2 - string 2

Returns:

true is the 2 strings are equal

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.

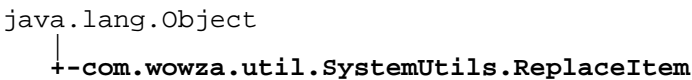
getSystemTime

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

Get system time in nanoseconds.

com.wowza.util

Class SystemUtils.ReplaceItem



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 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

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 Media Server 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	<u>DATA_TYPE_AMF3_NUMBER</u> Value: 5
public static final	<u>DATA_TYPE_AMF3_OBJECT</u> Value: 10
public static final	<u>DATA_TYPE_AMF3_STRING</u> Value: 6
public static final	<u>DATA_TYPE_AMF3_UNDEFINED</u> Value: 0
public static final	<u>DATA_TYPE_AMF3_XML_LEGACY</u> Value: 7
public static final	<u>DATA_TYPE_AMF3_XML_TOP</u> Value: 11
public static final	<u>DATA_TYPE_ARRAY</u> Value: 10
public static final	<u>DATA_TYPE_AS_OBJECT</u> Value: 13
public static final	<u>DATA_TYPE_BOOLEAN</u> Value: 1
public static final	<u>DATA_TYPE_BYTEARRAY</u> Value: 33
public static final	<u>DATA_TYPE_CUSTOM_CLASS</u> Value: 16
public static final	<u>DATA_TYPE_DATE</u> Value: 11
public static final	<u>DATA_TYPE_INTEGER</u> Value: 32
public static final	<u>DATA_TYPE_LONG_STRING</u> Value: 12
public static final	<u>DATA_TYPE_MIXED_ARRAY</u> Value: 8
public static final	<u>DATA_TYPE_MOVIE_CLIP</u> 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	<code>serialize</code> (java.io.DataOutputStream out) Serialize object to output stream
abstract void	<code>serialize</code> (java.io.DataOutputStream out, <code>AMFDataContextSerialize</code> context) Serialize object to output stream
abstract void	<code>serialize</code> (java.io.DataOutputStream out, int objectEncoding) Serialize object to output stream
abstract byte[]	<code>serialize</code> (int objectEncoding) Serial object to byte array
void	<code>setType</code> (int type) Sets the the data type for this object
static int	<code>skipByte</code> (java.nio.ByteBuffer data) Skip forward one byte in the byte buffer
static boolean	<code>testNextByte</code> (java.nio.ByteBuffer data, int test) Peek at the next value in data to see if its the test value
static boolean	<code>triggerAMF3Switch</code> (<code>AMFData</code> 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:

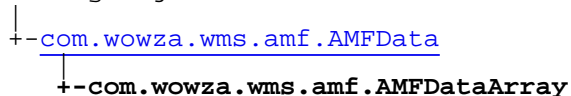
data - AMF object

Returns:

true if the object is serialized differently in AMF3

com.wowza.wms.amf Class AMFDataArray

java.lang.Object



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[]	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	set() (int index, AMFData data) Set an array item
void	set() (int index, boolean data) Set an boolean value (will be wrapped in an AMFDataItem object)
void	set() (int index, java.util.Date data) Set an date value (will be wrapped in an AMFDataItem object)
void	set() (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

(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 a 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

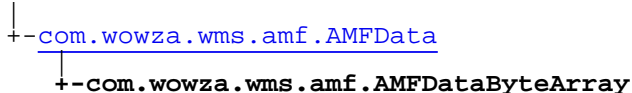
```
public String toString()
```

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 AMDDDataByteArray 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 AMDDDataByteArray 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 AMDDDataByteArray 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 AMDDDataByteArray 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

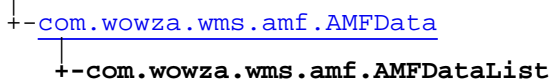
```
public byte[] serialize(AMFDataContextSerialize context)
```

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	<code>getFloat(int index)</code> Get item at index return as float
int	<code>getInt(int index)</code> Get item at index return as int
long	<code>getLong(int index)</code> Get item at index return as long
<code>AMFDataObj</code>	<code>getObject(int index)</code> Get item at index return as AMFDataObj
short	<code>getShort(int index)</code> Get item at index return as short
String	<code>getString(int index)</code> Get item at index return as String
int	<code>getType(int index)</code> Get type of item at index.
Object	<code>getValue()</code>
<code>AMFData</code>	<code>remove(int index)</code> Remove an element from the AMFDataList object
byte[]	<code>serialize()</code>
byte[]	<code>serialize(AMFDataContextSerialize context)</code>
byte[]	<code>serialize(AMFDataContextSerialize context, byte[] prepend)</code>
void	<code>serialize(java.io.DataOutputStream out)</code>
void	<code>serialize(java.io.DataOutputStream out, AMFDataContextSerialize context)</code>
void	<code>serialize(java.io.DataOutputStream out, AMFDataContextSerialize context, byte[] prepend)</code>
void	<code>serialize(java.io.DataOutputStream out, int objectEncoding)</code>
byte[]	<code>serialize(int objectEncoding)</code>
void	<code>set(int index, AMFData data)</code> Set an array item
void	<code>set(int index, boolean data)</code> Set an boolean value (will be wrapped in an AMFDataItem object)
void	<code>set(int index, java.util.Date data)</code> Set an date value (will be wrapped in an AMFDataItem object)
void	<code>set(int index, double data)</code> 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](#), [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](#), [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](#), [getShort](#), [getShort](#), [getString](#), [getString](#), [getTrait](#), [getValue](#), [put](#), [put](#), [put](#), [put](#), [put](#), [put](#), [put](#), [remove](#), [remove](#), [serialize](#), [serialize](#), [serialize](#), [serialize](#), [serialize](#), [serialize](#), [setClassName](#), [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](#), [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](#), [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	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	<u>containsKey</u> (String name)
void	<u>deserialize</u> (java.nio.ByteBuffer data)
void	<u>deserialize</u> (java.nio.ByteBuffer data, <u>AMFDataContextDeserialize</u> context)
<u>AMFData</u>	<u>get</u> (int index)
<u>AMFData</u>	<u>get</u> (String name)
boolean	<u>getBoolean</u> (int index)
boolean	<u>getBoolean</u> (String name)
byte	<u>getBytes</u> (int index)
byte	<u>getBytes</u> (String name)
String	<u>getClassName</u> ()
java.util.Date	<u>getDate</u> (int index)
java.util.Date	<u>getDate</u> (String name)
double	<u>getDouble</u> (int index)
double	<u>getDouble</u> (String name)
float	<u>getFloat</u> (int index)
float	<u>getFloat</u> (String name)
int	<u>getInt</u> (int index)
int	<u>getInt</u> (String name)
String	<u>getKey</u> (int index)
java.util.List	<u>getKeys</u> ()
long	<u>getLong</u> (int index)
long	<u>getLong</u> (String name)
<u>AMFDataObj</u>	<u>getObject</u> (int index)

AMFDataObj	getObject (String name)
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)
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 , 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

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

DECODE_TRAITS_EXT

public static final int **DECODE_TRAITS_EXT**

Constant value: 3

DECODE_TRAITS

public static final int **DECODE_TRAITS**

(continued from last page)

Constant value: **4**

members

protected java.util.Map **members**

order

protected java.util.List **order**

trait

protected com.wowza.wms.amf.AMFDataTrait **trait**

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

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:

(continued from last page)

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)
```

(continued from last page)

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)
```

getDate

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

getObject

```
public AMFDataObj getObject(String name)
```

getString

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

(continued from last page)

getInt

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

getLong

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

getShort

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

getByte

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

getDouble

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

getFloat

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

getBoolean

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

getDate

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

getObject

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

(continued from last page)

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

getValue

```
public Object getValue()
```

Convert object to Java native class

(continued from last page)

toString

```
public String toString()
```

Return object as formatted string

getClassName

```
public String getClassName()
```

setClassName

```
public void setClassName(String className)
```

getTrait

```
public AMFDataTrait getTrait()
```

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	AMFDEBUGHEADERSIZE
public static final	AMFFORCETYPE1 Value: true

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

AMFFORCETYPE1

public static final boolean **AMFFORCETYPE1**

Constant value: **true**

AMFDEBUGHEADERSIZE

public static boolean **AMFDEBUGHEADERSIZE**

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)

Parameters:

(continued from last page)

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)
```

Set packet size

(continued from last page)

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)

setTimecode

```
public void setTimecode(long timecode)
```

(continued from last page)

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:

absolute timecode

(continued from last page)

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

isByteContainerFull

```
public boolean isByteContainerFull()
```

(continued from last page)

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)
```

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 **AMFPacket**
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	AMFPacket() Create new empty packet
public	AMFPacket(int type, int src, int size) 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
AMFPacket	clone()
long	getAbsTimecode() Get absolute timecode (milliseconds)
byte[]	getData() Get data as byte[]
java.nio.ByteBuffer	getDataBuffer() Get data as ByteBuffer
int	getFirstByte() Get first byte of data (used to peek into packet)
int	getMissing() Get the number of bytes remaining unfilled in the packet
int	getSecondByte() 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>isVideo()</code> Is this an audio packet IVHost.CONTENTTYPE_VIDEO
void	<code>setAbsTimecode(long absTimecode)</code> Set absolute timecode (milliseconds)
void	<code>setDataBuffer(byte[] data)</code> Set the data buffer to a byte array
void	<code>setDataBuffer(java.nio.ByteBuffer data)</code> Set the data for this packet
void	<code>setSeq(long seq)</code> Set packet sequence.
void	<code>setSize(int size)</code> Set packet size
void	<code>setSrc(int src)</code> Set stream id (0 if not stream data)
void	<code>setTimecode(long timecode)</code> Set timecode (milliseconds) relative,
void	<code>setTimecodes(long timecode, long absTimecode)</code> Set both relative and absolute timecode in one call (milliseconds)
void	<code>setType(int type)</code> Set content type IVHost.CONTENTTYPE_*
String	<code>toString()</code> Return object as formatted string
void	<code>truncatePacket(int newSize)</code>

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.CONTENT*TYPE_*

src - stream id

size - packet size (bytes)

Methods

clone

```
public AMFPacket clone()
```

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()
```

(continued from last page)

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

(continued from last page)

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) 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)

(continued from last page)

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()
```

(continued from last page)

Get data as ByteBuffer

Returns:

data as ByteBuffer

getData

```
public byte[] getData()
```

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

(continued from last page)

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:

number of bytes unfilled in packet

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

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
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

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)
```

Set name of application

(continued from last page)

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_*)

Parameters:

counterIndex - counter index (see IVHost.COUNTER_*)

(continued from last page)

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( )
```

Get a list of application instance names

Returns:

(continued from last page)

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.toString());
    }
    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

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_*)

com.wowza.wms.application Interface IApplicationInstance

public interface **IApplicationInstance**
extends

IApplicationInstance: public interface to ApplicationInstance object

Field Summary

public static final	<u>DEFAULT_APPINSTANCE_NAME</u> Value: _definst_
public static final	<u>PROTOCOLUSAGE_CUPERTINO</u> Value: 7
public static final	<u>PROTOCOLUSAGE_RTMP</u> Value: 0
public static final	<u>PROTOCOLUSAGE_RTMP_E</u> Value: 4
public static final	<u>PROTOCOLUSAGE_RTMP_S</u> Value: 2
public static final	<u>PROTOCOLUSAGE_RTMP_T</u> Value: 1
public static final	<u>PROTOCOLUSAGE_RTMP_T_E</u> Value: 5
public static final	<u>PROTOCOLUSAGE_RTMP_T_S</u> Value: 3
public static final	<u>PROTOCOLUSAGE_RTP</u> Value: 6
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: 11

public static final	PROTOCOLUSAGE_WEBM Value: 10
---------------------	--

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	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	addMediaWriterListener (IMediaWriterActionNotify listener) Add a MediaWriter listener class.
void	addModuleListener (IModuleNotify moduleListener) Add module listener.
void	addPlayStreamByName (IMediaStream stream, String name) Add a media stream to the list of streams that are listening for a published stream
void	addPublisher (Publisher publisher) Add a server side publisher to this application instance
void	addRTPIncomingDatagramPortAll () Allow all incoming RTP UDP ports for this application instance
void	addRTPIncomingDatagramPortRange (int startPort, int endPort) Add a port range to the list of valid incoming RTP UDP ports
void	addRTPSession (RTPSession 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</u> (int index) 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</u> (int index) 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 <u>getClient(index)</u>)

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(String httpStreamName, boolean doCreate)</u> 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(int protocol)</u> Get the current number of HTTPStreamerSessions associated with this application instance by protocol.
int	<u>getHTTPStreamerSessionCount(int protocol, String streamName)</u> Get the current number of HTTPStreamerSessions associated with this application instance and stream name by protocol.
int	<u>getHTTPStreamerSessionCount(String streamName)</u> Get the current number of HTTPStreamerSessions associated with this application instance and stream name
java.util.Map	<u>getHTTPStreamerSessionCountsByName(int protocol)</u> 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	<u>getLiveStreamPacketizerList</u> () Get the comma separated list of LiveStreamPacketizers names being used by this application (see conf/LiveStreamPacketizers.xml)
<u>WMSProperties</u>	<u>getLiveStreamPacketizerProperties</u> () Get the property collection of LiveStreamPacketizer settings that are specific to this application instance
<u>ILiveStreamTranscoderControl</u>	<u>getLiveStreamTranscoderControl</u> () Get the Live Stream Transcoder Controller.
String	<u>getLiveStreamTranscoderList</u> () Get comma separated list of transcoders to use for this application instance
int	<u>getMaximumPendingReadBytes</u> () Set maximum number of bytes a client connection can have waiting to be written before the connection is terminated.
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.
int	<u>getMaxStorageDirDepth</u> () Maximum folder depth allowed for the StreamStorageDir and SharedObjectStorageDir paths
<u>WMSProperties</u>	<u>getMediaCasterProperties</u> () Get the property collection of media caster settings that are specific to this application instance
int	<u>getMediacasterRTPRTSPRTPTransportMode</u> () 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	getModuleInstance(String name) Get the instance of the module class for this application instance.
ModuleList	getModuleList() Get the list of loaded modules.
String	getName() Get applicationInstance name
int	getPingTimeout() Get ping timeout (milliseconds)
int	getPlayStreamCount(String streamName) Get the number of Flash players playing a given stream name
java.util.Map	getPlayStreamCountsByName() Get a map of stream names to number of Flash players playing the stream name
java.util.List	getPlayStreamsByName(String name) Get a list of media streams that are listening for published stream.
WMSProperties	getProperties() Get applicationInstance properties
boolean[]	getProtocolUsage() Get the protocols in use by this application instance (see IApplicationInstance.PROTOCOLUSAGE_*)
void	getProtocolUsage(boolean[] protocolsInUse) Get the protocols in use by this application instance (see IApplicationInstance.PROTOCOLUSAGE_*)
int	getPublisherCount() Get the current number of server side publishers
java.util.List	getPublishers() Get the set of server side publishers

java.util.List	<u>getPublishStreamNames()</u> Get the list of live stream names currently being published.
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	<u>getRsoStoragePath()</u> Get the resolved storage path to the shared objects
int	<u>getRTPAVSyncMethod()</u> Get RTP audio/video sync method (RTPStream.AVSYNCMETHODS_SENDERREPORT, RTPStream.AVSYNCMETHODS_SYSTEMCLOCK, RTPStream.AVSYNCMETHODS_RTPTIMECODE)
int	<u>getRTPIdleFrequency()</u> Set the default RTP idle frequency (milliseconds)
int	<u>getRTPMaxRTCPWaitTime()</u> Get the maximum time to wait for RTCP packets (milliseconds)
String	<u>getRTPPlayAuthenticationMethod()</u> Get the RTP play authentication method (as defined in conf/Authentication.xml)
<u>WMSProperties</u>	<u>getRTPProperties()</u> Get the property collection of RTP settings that are specific to this application instance
String	<u>getRTPPublishAuthenticationMethod()</u> Get the RTP publish authentication method (as defined in conf/Authentication.xml)
int	<u>getRTPSessionCount()</u> Get the number of RTP sessions running under this application instance
int	<u>getRTPSessionCount(String streamName)</u> Get the number of RTP player streams playing a given stream name
java.util.Map	<u>getRTPSessionCountsByName()</u> Get a map of stream names and session counts of RTP sessions
java.util.List	<u>getRTPSessions()</u> Get a list of RTP sessions running under this application instance
java.util.List	<u>getRTPSessions(String streamName)</u> Get a list of RTP sessions running under this application instance playing a given stream name
String	<u>getRTSPBindIpAddress()</u> Set the IP address to which UDP ports will be bound for RTSP/RTP sessions
String	<u>getRTSPConnectionAddressType()</u> Get the connection IP address type (IP4) to used in the Session Description Protocol data exchanged for an RTSP/RTP session
String	<u>getRTSPConnectionIpAddress()</u> Get the connection IP address to used in the Session Description Protocol data exchanged for an RTSP/RTP session

int	<u>getRTSPMaximumPendingWriteBytes()</u> Get the maximum number of pending write bytes for an RTSP session
String	<u>getRTSPOriginAddressType()</u> Get the origin IP address type (IP4) to used in the Session Description Protocol data exchanged for an RTSP/RTP session
String	<u>getRTSPOriginIpAddress()</u> 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>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
String	<u>getTimeRunning()</u> Get time applicationInstance running
double	<u>getTimeRunningSeconds()</u> Get time running in seconds
LiveStreamTranscoderApplicationContext	<u>getTranscoderApplicationContext()</u> Get live stream transcoder application context
<u>WMSProperties</u>	<u>getTranscoderProperties()</u> Get the property collection of Transcoder settings that are specific to this application instance
int	<u>getValidationFrequency()</u> Get time between validation pings (milliseconds)
<u>IVHost</u>	<u>getVHost()</u> Get parent vHost
void	<u>incClientCountTotal()</u> Increment the total number of connected client counter by one
boolean	<u>isAcceptConnection()</u> Is auto accept connection on/off
boolean	<u>isRTPIncomingDatagramPortValid(int port)</u> Check a port number to be sure it is a valid RTP UDP port for this application instance
boolean	<u>isValidateFMLEConnections()</u> Returns true if validating FMLE connection (default is false)
void	<u>notifyDvrStreamManagerCreate(IDvrStreamManager dvrStoreManager)</u> Notify listeners that Dvr Application Store Manager has been created.
void	<u>notifyDvrStreamManagerDestroy(IDvrStreamManager dvrManager)</u> Notify listeners that Dvr Application Store Manager has been destroyed.
void	<u>notifyDvrStreamManagerInit(IDvrStreamManager dvrStoreManager)</u> Notify listeners that Dvr Application Store Manager has been initialized.
void	<u>notifyLiveStreamDvrRecorderCreate(ILiveStreamDvrRecorder dvr, String streamName)</u> Notify Dvr Recorder Create
void	<u>notifyLiveStreamDvrRecorderDestroy(ILiveStreamDvrRecorder dvr)</u> Notify DVR Recorder has been destroyed.
void	<u>notifyLiveStreamDvrRecorderInit(ILiveStreamDvrRecorder dvr, String streamName)</u> Notify DVR Recorder has been initialized.

void	<u>notifyLiveStreamPacketizerCreate(ILiveStreamPacketizer liveStreamPacketizer, String streamName)</u> Notify Live Stream Packetizer Create
void	<u>notifyLiveStreamPacketizerDestroy(ILiveStreamPacketizer liveStreamPacketizer)</u> Notify Live Stream Packetizer Destory
void	<u>notifyLiveStreamPacketizerInit(ILiveStreamPacketizer liveStreamPacketizer, String streamName)</u> Notify Live Stream Packetizer Init
void	<u>notifyLiveStreamTranscoderCreate(ILiveStreamTranscoder liveStreamTranscoder, IMediaStream stream)</u> Notify live stream transcoder create
void	<u>notifyLiveStreamTranscoderDestroy(ILiveStreamTranscoder liveStreamTranscoder, IMediaStream stream)</u> Notify live stream transcoder destroy
void	<u>notifyLiveStreamTranscoderInit(ILiveStreamTranscoder liveStreamTranscoder, IMediaStream stream)</u> Notify live stream transcoder init
void	<u>notifyMediaReaderClose(IMediaReader mediaReader, IMediaStream stream)</u> Notify media reader notifyMediaReaderClose
void	<u>notifyMediaReaderCreate(IMediaReader mediaReader)</u> Notify media reader notifyMediaReaderCreate
void	<u>notifyMediaReaderExtractMetaData(IMediaReader mediaReader, IMediaStream stream)</u> Notify media reader notifyMediaReaderExtractMetaData
void	<u>notifyMediaReaderInit(IMediaReader mediaReader, IMediaStream stream)</u> Notify media reader notifyMediaReaderInit
void	<u>notifyMediaReaderOpen(IMediaReader mediaReader, IMediaStream stream)</u> Notify media reader notifyMediaReaderOpen
void	<u>notifyMediaWriterOnFLVAddMetadata(IMediaStream stream, java.util.Map extraMetadata)</u> Notify all MediaWriter listeners of onFLVAddMetadata
void	<u>notifyMediaWriterOnWriteComplete(IMediaStream stream, java.io.File file)</u> Notify all MediaWriter listeners of onWriteComplete
void	<u>parseAllowDomains(String domainFilterStr)</u> Parse a comma delimited list of domain names used to control access to this application.
String	<u>readAppInstConfig(String sName)</u> Method to read xml config file..
void	<u>registerPlayRTPSession(RTPSession rtpSession)</u> Register an RTP session as a play session
void	<u>removeClientListener(IClientNotify clientListener)</u> Remove client listener.
void	<u>removeDvrRecorderListener(ILiveStreamDvrRecorderActionNotify listener)</u> Remove a Dvr Recorder listener (see: ILiveStreamDvrRecorderActionNotify)

void	<u>removeDvrStreamManagerListener</u> (<u>IDvrStreamManagerActionNotify</u> listener) Remove a Dvr Application Store Manager listener (see: IDvrStoreActionNotify)
void	<u>removeHTTPStreamerSession</u> (<u>IHTTPStreamerSession</u> httpStreamerSession) Remove a HTTPStreamerSession from this application instance
void	<u>removeLiveStreamPacketizerListener</u> (<u>ILiveStreamPacketizerActionNotify</u> liveStreamPacketizerListener) Remove a Live Stream Packetizer listener (see: ILiveStreamPacketizerActionNotify)
void	<u>removeLiveStreamTranscoderListener</u> (<u>ILiveStreamTranscoderNotify</u> liveStreamTranscoderListener) 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>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>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>setRTPMaXRTCPWaitTime</u> (int rtpMaxRTCPWaitTime) Set the maximum time to wait for RTCP packets (milliseconds)
void	<u>setRTPPlayAuthenticationMethod</u> (String rtpPlayAuthenticationMethod) Set the RTP play authentication method (as defined in conf/Authentication.xml)
void	<u>setRTPPublishAuthenticationMethod</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>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

(continued from last page)

PROTOCOLUSAGE_RTMP

```
public static final int PROTOCOLUSAGE_RTMP
```

Constant value: **0**

PROTOCOLUSAGE_RTMPT

```
public static final int PROTOCOLUSAGE_RTMPT
```

Constant value: **1**

PROTOCOLUSAGE_RTMPS

```
public static final int PROTOCOLUSAGE_RTMPS
```

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**

(continued from last page)

PROTOCOLUSAGE_SMOOTH

```
public static final int PROTOCOLUSAGE_SMOOTH
```

Constant value: **8**

PROTOCOLUSAGE_SANJOSE

```
public static final int PROTOCOLUSAGE_SANJOSE
```

Constant value: **9**

PROTOCOLUSAGE_WEBM

```
public static final int PROTOCOLUSAGE_WEBM
```

Constant value: **10**

PROTOCOLUSAGE_TOTAL

```
public static final int PROTOCOLUSAGE_TOTAL
```

Constant value: **11**

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:

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

(continued from last page)

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()
```

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

(continued from last page)

Returns:

parent vHost

getProperties

```
public WMSProperties getProperties()
```

Get applicationInstance properties

Returns:

applicationInstance properties

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

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

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

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)
```

(continued from last page)

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

(continued from last page)

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)
```

(continued from last page)

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

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

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

(continued from last page)

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.

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

(continued from last page)

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

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.

(continued from last page)

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()
```

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

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

(continued from last page)

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

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:

(continued from last page)

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:

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:

(continued from last page)

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

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

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

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

(continued from last page)

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.

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

(continued from last page)

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)
```

(continued from last page)

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

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)
```

Notify listeners that Dvr Application Store Manager has been created.

Parameters:

dvrStoreManager - Dvr Application Store Manager

notifyDvrStreamManagerInit

```
public void notifyDvrStreamManagerInit(IDvrStreamManager dvrStoreManager)
```

Notify listeners that Dvr Application Store Manager has been initialized.

(continued from last page)

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_*

(continued from last page)

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_*)

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

putAll

```
public void putAll(java.util.Map m)
```

cloneProperties

```
public static void cloneProperties(WMSProperties from, WMSProperties to)
```

Copy all properties from "from" properties object to "to" properties object.

Parameters:

from - source properties

(continued from last page)

to - destination properties

setProperty

```
public void setProperty(String name,  
                        Object value)
```

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:

(continued from last page)

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[]

(continued from last page)

toString

```
public String toString()
```

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	<code>getLastConnectRejectedByReasonStampString(int reason)</code> Get time (milliseconds) of the last rejected connection by reason to this object as formatted string.
java.util.Date	<code>getLastConnectRejectedByReasonTime(int reason)</code> Get date and time of last rejected connection by reason to this object as Date object.
String	<code>getLastConnectRejectedByReasonTimeString(int reason)</code> Get date and time of last rejected connection by reason to this object as formatted string.
long	<code>getLastConnectRejectedStamp()</code> Get time (milliseconds) of the last rejected connection to this object.
String	<code>getLastConnectRejectedStampString()</code> Get time (milliseconds) of the last rejected connection to this object as formatted string.
java.util.Date	<code>getLastConnectRejectedTime()</code> Get date and time of last rejected connection to this object as Date object.
String	<code>getLastConnectRejectedTimeString()</code> Get date and time of last rejected connection to this object as formatted string.
long	<code>getLastDisconnectStamp()</code> Get time (milliseconds) of the last disconnected connection to this object.
String	<code>getLastDisconnectStampString()</code> Get time (milliseconds) of the last disconnected connection to this object as formatted string.
java.util.Date	<code>getLastDisconnectTime()</code> Get date and time of last disconnected connection to this object as Date object.
String	<code>getLastDisconnectTimeString()</code> Get date and time of last disconnected connection to this object as Date object as formatted string.
long	<code>getTotal()</code> Get total number of connection attempts to this object.
long	<code>getTotalAccepted()</code> Get total number of accepted connections to this object.
long	<code>getTotalRejected()</code> Get total number of rejected connections to this object.
void	<code>incrementAccept(ConnectionHolder connectionHolder, java.util.Date date, long stamp, byte[] license)</code> Increment accepted connections.
void	<code>incrementReject(ConnectionHolder connectionHolder, int reason, java.util.Date date, long stamp, byte[] license)</code> Increment reject connection.
void	<code>rejectConnection(ConnectionHolder connectionHolder, int reason, byte[] license)</code> Reject connection.
void	<code>removeConnectionListener(IConnectionNotify connectionNotify)</code> Remove connection listener
void	<code>setConnectionValidator(IConnectionValidator connectionValidator)</code> 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 (same as FMS getAgent())
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)
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)
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
<u>AMFObj</u>	<u>getRespAMFAudioObj()</u> (<u>IMediaStream</u> stream) Get the audio response AMFObj for a given mediaStream
<u>AMFObj</u>	<u>getRespAMFDataObj()</u> (<u>IMediaStream</u> 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
IOPerformanceCounter	getTotalIOPerformanceCounter () Get performance counter for all bytes transferred to this client.
String	getUri () Get the full URI of the connection string

IVHost	getVHost() Get parent vHost
ClientWriteListener	getWriteListener() Object that tracks write operations
boolean	isAcceptConnection() Is auto accept connection
boolean	isConnected() Is this client connected
boolean	isEncrypted() Is this connection encrypted (RTMPE or RTMPTE)
boolean	isFlashMediaLiveEncoder() Returns true if this connection is the Flash Media Live Encoder
boolean	isFlashVersion10() Returns true if the Flash version is equal or greater than 10.x.x.x
boolean	isFlashVersion90115() Returns true if the Flash version is equal or greater than 9.0.115.x
boolean	isFlashVersionH264Capable() Returns true if the connected client is capable of playing H.264 video (Flash player 9.0.45.x or greater)
boolean	isLiveRepeater() Returns true if this connection is from the live stream repeater
boolean	isObjectEncodingAMF0() Is the object encoding for this client AMF0
boolean	isObjectEncodingAMF3() Is the object encoding for this client AMF3
boolean	isSecure() Is this connection protected by either SSL or encryption (RTMPE, RTMPTE, RTMPS)
boolean	isSSL() Is this connection SSL (RTMPS)
boolean	isValidFMLEConnections() 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	<u>redirectConnection</u> (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	<u>rejectConnection</u> () Reject connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected)
void	<u>rejectConnection</u> (<u>AMFData</u> errorObj) Reject connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected)
void	<u>rejectConnection</u> (String errorStr) Reject connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected)
void	<u>rejectConnection</u> (String description, <u>AMFData</u> errorObj) Reject connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected)
void	<u>rejectConnection</u> (String description, String errorStr) Reject connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected)
void	<u>reparentClient</u> (<u>IVHost</u> vhost) Move a client object to a new vhost.
void	<u>setAcceptConnection</u> (boolean acceptConnection) Set the default for accept connection
void	<u>setAcceptConnectionDescription</u> (String description) Call this method from onConnect to set the info.description property returned in NetConnection onStatus handler
void	<u>setAcceptConnectionExObj</u> (<u>AMFDataObj</u> acceptConnectionExObj) Call this method from onConnect to set the info.ex property returned in NetConnection onStatus handler
void	<u>setAcceptConnectionObj</u> (<u>AMFData</u> 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	<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>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	<u>setThreadContext</u> () Set the thread logging context to this client
void	<u>setValidateFMLEConnections</u> (boolean validateFMLEConnections) Returns true if validating FMLE connection (default is false)
void	<u>shutdownClient</u> () Gracefully shutdown a client.
int	<u>testFlashVersion</u> (int[] version) Test to see if the connected client flash version is equal to or greater than a given value.
void	<u>touch</u> () Update the last touch time for client

Fields

(continued from last page)

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: *

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

(continued from last page)

getClientId

```
public int getClientId()
```

Get client id. Assigned by server and connection time.

Returns:

client id

getFlashVer

```
public String getFlashVer()
```

Get client flash version (same as FMS getAgent())

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()
```

(continued from last page)

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()
```

(continued from last page)

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

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)`

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

(continued from last page)

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

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)

(continued from last page)

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)

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:

(continued from last page)

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)
```

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)

getVHost

```
public IVHost getVHost( )
```

Get parent vHost

(continued from last page)

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 Media Server 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

isValidateFMLEConnections

```
public boolean isValidateFMLEConnections()
```

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

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

com.wowza.wms.client Interface IConnectionValidator

public interface **IConnectionValidator**
extends

IConnectionValidator: callback for validating connections

Method Summary

boolean	validateConnection (ConnectionCounter connectionCounter, ConnectionHolder connectionHolder, byte[] license) Triggered on connection
---------	--

Methods

validateConnection

```
public boolean validateConnection(ConnectionCounter connectionCounter,  
    ConnectionHolder connectionHolder,  
    byte[] license)
```

Triggered on connection

Parameters:

connectionCounter - connection counter
connectionHolder - connection holder

Returns:

isValid

com.wowza.wms.client Interface ILicenseNotify

public interface **ILicenseNotify**
extends

Method Summary

void	<code>onAcceptLicense</code> (LicenseCounter licenseCounter, LicenseHolder licenseHolder, java.util.Date date, long stamp) Triggered when license accepted
void	<code>onDisconnect</code> (LicenseCounter licenseCounter, LicenseHolder licenseHolder, boolean isValid, java.util.Date date, long stamp) Triggered when client disconnected
void	<code>onRejectLicense</code> (LicenseCounter licenseCounter, LicenseHolder licenseHolder, int reason, java.util.Date date, long stamp) Triggered when license rejected

Methods

onAcceptLicense

```
public void onAcceptLicense(LicenseCounter licenseCounter,  
    LicenseHolder licenseHolder,  
    java.util.Date date,  
    long stamp)
```

Triggered when license accepted

Parameters:

licenseCounter
licenseHolder
date
stamp

onRejectLicense

```
public void onRejectLicense(LicenseCounter licenseCounter,  
    LicenseHolder licenseHolder,  
    int reason,  
    java.util.Date date,  
    long stamp)
```

Triggered when license rejected

Parameters:

licenseCounter
licenseHolder
reason
date
stamp

onDisconnect

```
public void onDisconnect(LicenseCounter licenseCounter,  
    LicenseHolder licenseHolder,  
    boolean isValid,  
    java.util.Date date,  
    long stamp)
```

Triggered when client disconnected

Parameters:

- licenseCounter
- licenseHolder
- isValid
- date
- stamp

com.wowza.wms.client Interface ILicenseValidator

public interface **ILicenseValidator**
extends

Method Summary

boolean	validateLicense (LicenseCounter licenseCounter, LicenseHolder licenseHolder, byte[] license)
---------	--

Methods

validateLicense

```
public boolean validateLicense(LicenseCounter licenseCounter,  
    LicenseHolder licenseHolder,  
    byte[] license)
```

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)
```

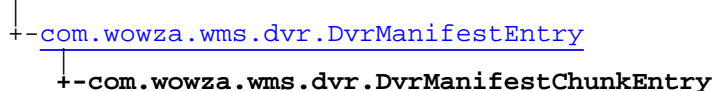
shouldDeleteArchivedStream

```
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	serialize (java.io.DataOutputStream out)
void	setEncryptions (DvrEncryptionInfoHolder encryptions) Set associated encryptions.
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

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()
```

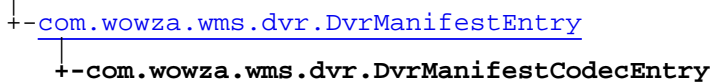
toString

```
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

java.lang.Object
└─com.wowza.wms.dvr.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.

Method Summary

void	<u>deserialize</u> (byte[] data) Deserialize.
int	<u>getEndIndex</u> () Get end index.
IndexRange	<u>getRange</u> () Get index range.
int	<u>getStartIndex</u> () Get start index.
int	<u>getType</u> () Get range type
boolean	<u>isEmpty</u> ()
boolean	<u>isInRange</u> (int index) Is index in range.
byte[]	<u>serialize</u> () Serialize range.
void	<u>setRange</u> (IndexRange range) Set index range.
String	<u>toString</u> ()

Methods inherited from class java.lang.Object

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

Methods

getType

public int **getType**()

(continued from last page)

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.

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.

(continued from last page)

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.

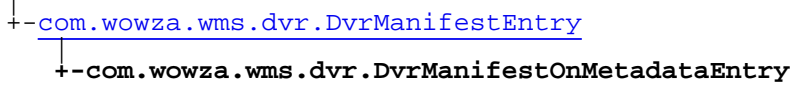
toString

```
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
  |
  +- com.wowza.wms.dvr.DvrManifestEntry
      |
      +- com.wowza.wms.dvr.DvrManifestTimeMapEntry
  
```

```

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	determineUtcTime (long currentUtc, long aChunkStartUtc, long aFirstPacketUtc, AMFPacket aFirstPacket, long vChunkStartUtc, long vFirstPacketUtc, AMFPacket vFirstPacket)
void	init (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

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_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_RECORDING_LENGTH Default value for DVR Property "dvrMaximumRecordingLength". Value: 108000
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.plugin.mediacache.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: 50
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	DEFAULT_PROPERTY_SMOOTH_MANIFEST_LIVE_CAN_SEEK Value: true
public static final	DEFAULT_PROPERTY_SMOOTH_MANIFEST_MAJOR_VERSION Value: 2
public static final	DEFAULT_PROPERTY_SMOOTH_MANIFEST_MINOR_VERSION Value: 1
public static final	DEFAULT_PROPERTY_SMOOTH_MANIFEST_RECORDED_CAN_PAUSE Value: true

public static final	DEFAULT_PROPERTY_SMOOTH_MANIFEST_RECORDED_CAN_SEEK Value: true
public static final	DEFAULT_PROPERTY_SMOOTH_MANIFEST_RECORDED_SPECIFY_DURATION Value: true
public static final	DEFAULT_PROPERTY_STORAGE_DIRECTORY Default value for DVR Property "dvrStorageDirectory". Value: <code>\${com.wowza.wms.context.VHostConfigHome}/dvr</code>
public static final	DEFAULT_PROPERTY_TEXT_READER_CLASS Default value for DVR Property "dvrTextReaderClass". Value: <code>com.wowza.wms.dvr.impl.io.DvrTextFileReader</code>
public static final	DEFAULT_PROPERTY_TEXT_WRITER_CLASS Default value for DVR Property "dvrTextWriterClass". Value: <code>com.wowza.wms.dvr.impl.io.DvrTextFileWriter</code>
public static final	DVR_DEFAULT_FILESTORE The default DVR store ID: "dvrfilestorage". Value: <code>dvrfilestorage</code>
public static final	DVR_DEFAULT_RECORDER_ID The default DVR recorder ID: "dvrrecorder". Value: <code>dvrrecorder</code>
public static final	DVR_REPEATER_PACKETIZER_ID The default DVR streaming repeater ID: "dvrstreamingrepeater". Value: <code>dvrstreamingrepeater</code>
public static final	DVR_STREAMING_PACKETIZER_ID The default DVR streaming packetizer ID: "dvrstreamingpacketizer". Value: <code>dvrstreamingpacketizer</code>
public static final	DVR_WINDOW_DURATION_UNLIMITED Value: 0
public static final	MEDIACACHE_PREFIX The MediaCache prefix for dvr repeater Value: <code>dvrorigin</code>
public static final	MIMETYPE_VIDEO_MP4 Constant for mime type "video/mp4" Value: <code>video/mp4</code>
public static final	PROPERTY_ALLOWABLE_AV_PACKET_DELTA DVR Property "dvrAllowableAVPacketDelta": for specifying how much audio and video packets may diverge before triggering an error. Value: <code>dvrAllowableAVPacketDelta</code>
public static final	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. Value: <code>dvrAppendDiscontinuityDelta</code>

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_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_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> 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_PLAYER_ADAPTER</u> HTTP Streamer Property "dvrDebugMbrPlayerAdapter": for turning on DVR MBR Player Adapter debug logging. Value: dvrDebugMbrPlayerAdapter
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_RECORDING_LENGTH</u> DVR Property "dvrMaximumRecordingLength": The maximum recording length in seconds. Value: dvrMaximumRecordingLength
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	<u>PROPERTY_PLAYLIST_REQUEST_DELEGATE</u> DVR Property "dvrPlaylistRequestDelegate": used to over-ride the class that generates a playlist request. Value: dvrPlaylistRequestDelegate
public static final	<u>PROPERTY_RECORD_AUDIO</u> DVR Property "recordAudio": for determining if audio should be recorded. Value: recordAudio
public static final	<u>PROPERTY_RECORD_DATA</u> DVR Property "recordData": for determining if data should be recorded. Value: recordData
public static final	<u>PROPERTY_RECORD_VIDEO</u> DVR Property "recordVideo": for determining if video should be recorded. Value: recordVideo
public static final	<u>PROPERTY_RECORDINGS_LOADER_CLASS</u> DVR Property "dvrRecordingsLoaderClass": for controlling the class responsible for loading DVR recordings. Value: dvrRecordingsLoaderClass
public static final	<u>PROPERTY_REPEATER_HEARTBEAT_DURATION</u> DVR Property "dvrRepeaterHeartbeatDuration": for time in ms that origin pings edges Value: dvrRepeaterHeartbeatDuration
public static final	<u>PROPERTY_REPEATER_SHARED_SECRET</u> DVR Property "dvrEncryptionSharedSecret": for encryption shared secret between origins and edges. Value: dvrEncryptionSharedSecret
public static final	<u>PROPERTY_SANJOSE_ABST_DURATION_TOLERANCE</u> DVR Property "dvrSanJosePlaylistAbstDurationEqualityTolerance": used to over-ride the tolerance when determining equal chunks lengths. Value: dvrSanJosePlaylistAbstDurationTolerance
public static final	<u>PROPERTY_SANJOSE_ABST_TIMESCALE</u> DVR Property "dvrSanJosePlaylistAbstTimescale": used to over-ride the time-scale for abst files. Value: dvrSanJosePlaylistAbstTimescale

public static final	<u>PROPERTY_SANJOSE_PLAYLIST_DELIVERYTYPE</u> DVR Property "dvrSanJosePlaylistDeliveryType": used to over-ride the mime type for DVR San Jose F4m playlists. Value: dvrSanJosePlaylistDeliveryType
public static final	<u>PROPERTY_SANJOSE_PLAYLIST_LIVE_STREAMTYPE</u> DVR Property "dvrSanJosePlaylistLiveStreamType": used to over-ride the stream type for live DVR in San Jose f4m playlists. Value: dvrSanJosePlaylistLiveStreamType
public static final	<u>PROPERTY_SANJOSE_PLAYLIST_MIMETYPE</u> 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_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	SANJOSE_F4M_VERSION_1_0 Constant for San Jose f4m version "1.0". Value: 1.0
public static final	SANJOSE_F4M_VERSION_2_0 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, 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_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.

(continued from last page)

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:

[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](#)

(continued from last page)

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](#)

[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**

(continued from last page)

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**

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
```

(continued from last page)

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](#)

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**

(continued from last page)

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".

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
```

(continued from last page)

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](#)

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**

(continued from last page)

See Also:

[PROPERTY_PACKET_DELTA_TO_RESET_TIME](#)

PROPERTY_PACKET_DELTA_TO_NOTIFY

```
public static final java.lang.String PROPERTY_PACKET_DELTA_TO_NOTIFY
```

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
```

(continued from last page)

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:

[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](#)

(continued from last page)

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:

[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.

(continued from last page)

Constant value: **1500**

See Also:

[PROPERTY_CHUNK_DURATION_MINIMUM](#)

PROPERTY_CHUNK_MEMORY_CACHESIZE

```
public static final java.lang.String PROPERTY_CHUNK_MEMORY_CACHESIZE
```

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_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**

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 min seconds. Recording stops when it reaches this value.

Add the property to Application/DVR/Properties section of Application.xml

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.plugin.mediache.impl.MediaCacheRandomAccessReader**

See Also:

[PROPERTY_MEDIACACHE_READER_CLASS](#)

PROPERTY_SMOOTH_MANIFEST_LIVE_CAN_SEEK

```
public static final java.lang.String PROPERTY_SMOOTH_MANIFEST_LIVE_CAN_SEEK
```

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**

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
```

(continued from last page)

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
```

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_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:

(continued from last page)

[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**

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
```

(continued from last page)

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:

[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
```

(continued from last page)

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**

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. Units are in chunks. The default value is 10 chunks. 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_DEBUG_METHODS

```
public static final java.lang.String PROPERTY_DEBUG_METHODS
```

(continued from last page)

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**

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**

(continued from last page)

PROPERTY_DEBUG_MBR_PLAYER_ADAPTER

```
public static final java.lang.String PROPERTY_DEBUG_MBR_PLAYER_ADAPTER
```

HTTP Streamer 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_ALIGNMENT

```
public static final java.lang.String PROPERTY_DEBUG_MBR_ALIGNMENT
```

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
```

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

Add this to Application/DVR/Properties section of Application.xml

Constant value: **dvrPlaylistRequestDelegate**

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**

(continued from last page)

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
```

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.

(continued from last page)

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".

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**

(continued from last page)

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
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: **50**

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
```

(continued from last page)

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
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

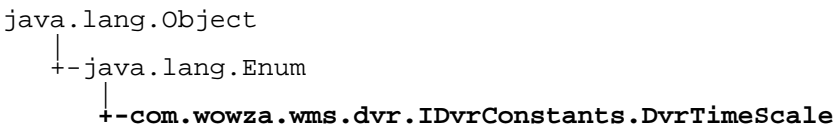
```
public static final java.lang.String PROPERTY_CUPERTINO_ON_CHUNK_START_RESET_COUNTER
```

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**

com.wowza.wms.dvr
Class 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, 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 final 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	ENCRYPTION_TYPE_CUPERTINO Cupertino AES encryption type. Value: 1
public static final	ENCRYPTION_TYPE_PLAYREADY Playready encryption type. Value: 2

Method Summary

Object	clone()
int	getEncryptionType() Serialize encryption info.
int	getSerializeSize() Return number of bytes required to serialize this encryption info.
byte[]	serialize() Serialize the encryption info, returning a byte buffer containing the serialized data.
int	serialize(byte[] buffer, int pos) 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.
Constant value: **2**

Methods

(continued from last page)

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:

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	<u>QUERY_PARAM_DVR</u> Value: DVR
---------------------	---

Method Summary

int	<u>calcBitrate</u> (<u>IDvrStreamStore</u> store, int type, long t)
String	<u>determineStreamVersion</u> (<u>IDvrStreamManager</u> dvrMgr, String baseStreamName)
DvrPlaylistRequest	<u>getDvrPlaylistRequest</u> (<u>IHTTPStreamerApplicationContext</u> appContext, <u>IDvrStreamStore</u> store, java.util.Map queryMap)
DvrPlaylistRequest	<u>getDvrPlaylistRequest</u> (<u>IHTTPStreamerApplicationContext</u> appContext, java.util.List stores, java.util.Map queryMap)
IDvrMbrPlaylistAlignment	<u>getPlaylistAlignment</u> (<u>IHTTPStreamerSession</u> httpStreamerSession)
boolean	<u>isPlaylistReady</u> (<u>IDvrStreamStore</u> store, DvrPlaylistRequest playlistRequest)
boolean	<u>isPlaylistReady</u> (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 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	<code>getCurrentTime()</code> Get current DVR time.
long	<code>getLastPurgeTime()</code> Get the DVR time when the last purge occurred.
void	<code>init(IDvrStreamStore store)</code> Initialize the controller.
boolean	<code>isPurgingEnabled()</code> Is Purging enabled for this controller.
void	<code>setCurrentDvrTime(long newDvrTime)</code> 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)	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	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	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.
long	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](#)

[getApplicationInstance](#), [getLiveStreamPacketizerId](#), [getProperties](#), [getRepeaterLastSequence](#), [getStartStream](#), [handlePacket](#), [init](#), [isActive](#), [isPacketizeAudio](#), [isPacketizeData](#), [isPacketizeVideo](#), [isRepeaterEdge](#), [resetStream](#), [setLiveStreamPacketizerId](#), [setPacketizeAudio](#), [setPacketizeData](#), [setPacketizeVideo](#), [setRepeaterEdge](#), [shutdown](#), [startStream](#), [touch](#)

Methods

getContextStr

```
public String getContextStr()
```

Get stream context string, useful for logging.

(continued from last page)

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.

(continued from last page)

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()
```

Get the storage directory.

Returns:

storage directory

(continued from last page)

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()
```

(continued from last page)

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()
```

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\(\)](#).

(continued from last page)

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)
```

Delete archived stream store.

Parameters:

vStreamName - versioned stream Name (e.g. myStream.0)

(continued from last page)

addManifestEntries

```
public void addManifestEntries(String vStreamName,  
    java.util.List entries)
```

Add to stream stores manifest.

Parameters:

vStreamName - versioned stream Name (e.g. myStream.0)

entries - list of manifest entries.

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 long 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)
```

Remove listener of DVR store lifecycle events.

Parameters:

listener - listener

(continued from last page)

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)
```

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)
```

stateChange

```
public void stateChange(String vStreamName,  
                        DvrStreamStoreState state)
```

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)
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>isLoaded</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	<u>setCanPlay</u> (boolean canPlay) Set the Stream Stores ability to play.

void	<u>setCanRecord</u> (boolean canRecord) Set the Stream Stores ability to record.
void	<u>setHasEncryption</u> (boolean hasEncryption) Set whether the store has encryption.
void	<u>shutdown</u> ()
boolean	<u>startRecording</u> () Request that recording start.
boolean	<u>stopRecording</u> () Request that stream recording stop.
long	<u>storeChunks</u> (int vDuration, DvrPacketHolder vPackets, int aDuration, DvrPacketHolder aPackets, int dDuration, DvrPacketHolder dPackets)
boolean	<u>storeOnMetadata</u> (long pt, long utc, <u>AMFPacket</u> metaPacket)

Methods

init

```
public void init()
```

Initialize DVR stream store.

reset

```
public void reset()
```

shutdown

```
public void shutdown()
```

getProperties

```
public WMSProperties getProperties()
```

Get stream store properties.

Returns:

properties

getDvrStorageWindowSeconds

```
public int getDvrStorageWindowSeconds()
```

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\(IDvrStreamStore\)](#)

(continued from last page)

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:

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

(continued from last page)

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 long 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)
```

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)
```

(continued from last page)

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(HTTPServerVersion.class).error("HTMLServerVersion:
"+e.toString());
        }
    }
}
```

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

(continued from last page)

HTTPProvider2Base

```
public HTTPProvider2Base()
```

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

setProperty

```
public void setProperty(WMSProperties properties)
```

Set properties

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:

(continued from last page)

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.toString());
        }
    }
}

```

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 Media Server Pro")
int	<code>getServerPort()</code> Get the port this request was received on
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'

Parameters:

name - header name

(continued from last page)

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()
```

Get the full request URI

(continued from last page)

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

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 Media Server Pro")

Returns:

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

getHeaderBytes

```
public byte[] getHeaderBytes()
```

(continued from last page)

Returns the header as bytes

Returns:

header as bytes

getMsgBytes

```
public byte[] getMsgBytes()
```

Return the message bytes

Returns:

message bytes

com.wowza.wms.http Interface IHTTPResponse

public interface **IHTTPResponse**
extends

Method Summary

boolean	<code>containsHeader</code> (String name) Returns true if reponse 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 reponse.
void	<code>setHeader</code> (String name, String value) Set header value
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 reponse. You can then write directly into the output stream.

Returns:

output stream for the reponse

`containsHeader`

public boolean **containsHeader**(String name)

Returns true if reponse header contains parameter name

(continued from last page)

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

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)
```

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:

(continued from last page)

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

Package

com.wowza.wms.httpstreamer.cupertinostreaming.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:

[IHTTPStreamerSession](#)

public class **HTTPStreamerSessionCupertino**
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, isPlayLogged, lastRequest, liveStreamingPacketizer, lock, mediaCasterStreamLock, playDuration, playStart, properties, queryStr, redirectSession, redirectSessionBody, redirectSessionCode, redirectSessionContentType, redirectSessionURL, referrer, serverIp, serverPort, sessionId, sessionProtocol, sessionTimeout, sessionType, stream, streamDomainStrSet, streamExt, streamName, streamNamePartMap, streamPosition, timeoutSession, totalIOPerformance2Last, totalIOPerformanceLast, uri, userAgent, userHTTPHeaders, userQueryStr, vhost

Fields inherited from interface [com.wowza.wms.httpstreamer.model.IHTTPStreamerSession](#)

[SESSIONPROTOCOL_COUNT](#), [SESSIONPROTOCOL_CUPERTINOSTREAMING](#), [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)
---------	---

static boolean	doesFileExist (IHTTPStreamerApplicationContext appContext, String rawStreamName, String streamExt, String streamName, IHTTPStreamerSession httpStreamerSession)
----------------	--

IHTTPStreamerCupertinoIndex	getIndex (IHTTPStreamerApplicationContext appContext, IHTTPStreamerSession httpStreamerSession, String rawStreamName, String streamExt, String streamName, long playStart, long playDuration)
---	--

void	logLiveChunk (LiveStreamPacketizerCupertinoChunk chunk)
void	logVODChunk (LiveStreamPacketizerCupertinoChunk chunk)
void	shutdown ()
void	updateLoggingValues ()

Methods inherited from class `com.wowza.wms.httpstreamer.model.HTTPStreamerSessionBase`

acceptSession, addIOPerformance, addIOPerformance2, addStreamDomainStr, addStreamDomainStrs, addUserHTTPHeaders, checkAndSetPlayLogged, clearLoggingValues, containsStreamDomainStr, containsStreamNameParts, doSessionRedirect, extractHTTPRequestInfo, getAppInstance, getConnectionHolder, getCookieStr, getDvrSessionInfo, getElapsedTime, getFileInfo, getHTTPDate, getHTTPHeader, getHTTPHeaderMap, getHTTPHeaderNames, getHTTPIntHeader, getHTTPStreamerAdapter, getIOPerformanceCounter, getIpAddress, getLastRequest, getLiveStreamingPacketizer, getLock, getPlayDuration, getPlayStart, getProperties, getQueryStr, getRedirectSessionBody, getRedirectSessionCode, getRedirectSessionContentType, getRedirectSessionURL, getReferrer, getServerIp, getServerPort, getSessionId, getSessionProtocol, getSessionTimeout, getSessionType, getStream, getStreamDomainStr, getStreamDomainStrList, getStreamExt, getStreamName, getStreamNameParts, getStreamPosition, getTimeRunning, getTimeRunningSeconds, getUri, getUserAgent, getUserHTTPHeaders, getUserQueryStr, getVHost, isAcceptSession, isActive, isFileInfo, isPlayLogged, isRedirectSession, isTimeout, isTimeoutSession, isValidated, isValidStreamDomainStr, lockRepeaterStreams, putFileInfo, putStreamNameParts, redirectSession, redirectSession, rejectSession, removeStreamDomainStr, setAcceptSession, setActive, setAppInstance, setCookieStr, setDvrSessionInfo, setHTTPStreamerAdapter, setIpAddress, setLiveStreamingPacketizer, setPlayDuration, setPlayLogged, setPlayStart, setQueryStr, setRedirectSession, setRedirectSessionBody, setRedirectSessionCode, setRedirectSessionContentType, setRedirectSessionURL, setReferrer, setServerIp, setServerPort, setSessionId, setSessionProtocol, setSessionTimeout, setSessionType, setStream, setStreamExt, setStreamName, setStreamPosition, setThreadContext, setTimeoutSession, setUri, setUserAgent, setUserHTTPHeader, setUserQueryStr, setValidated, setVHost, shutdown, 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](#), [addIOPerformance](#), [addIOPerformance2](#), [addStreamDomainStr](#),
[addStreamDomainStrs](#), [addUserHTTPHeaders](#), [checkAndSetPlayLogged](#), [clearLoggingValues](#),
[containsStreamDomainStr](#), [containsStreamNameParts](#), [doSessionRedirect](#),
[extractHTTPRequestInfo](#), [getAppInstance](#), [getConnectionHolder](#), [getCookieStr](#),
[getDvrSessionInfo](#), [getElapsedTime](#), [getFileInfo](#), [getHTTPHeader](#), [getHTTPHeaderMap](#),
[getHTTPHeaderNames](#), [getHTTPIntHeader](#), [getHTTPStreamerAdapter](#),
[getIOPerformanceCounter](#), [getIpAddress](#), [getLastRequest](#), [getLiveStreamingPacketizer](#),
[getLock](#), [getPlayDuration](#), [getPlayStart](#), [getProperties](#), [getQueryStr](#),
[getRedirectSessionBody](#), [getRedirectSessionCode](#), [getRedirectSessionContentType](#),
[getRedirectSessionURL](#), [getReferrer](#), [getServerIp](#), [getServerPort](#), [getSessionId](#),
[getSessionProtocol](#), [getSessionTimeout](#), [getSessionType](#), [getStream](#), [getStreamExt](#),
[getStreamName](#), [getStreamNameParts](#), [getStreamPosition](#), [getTimeRunning](#),
[getTimeRunningSeconds](#), [getUri](#), [getUserAgent](#), [getUserHTTPHeaders](#), [getUserQueryStr](#),
[getVHost](#), [isAcceptSession](#), [isActive](#), [isFileInfo](#), [isPlayLogged](#), [isRedirectSession](#),
[isTimeout](#), [isTimeoutSession](#), [isValidated](#), [isValidStreamDomainStr](#),
[lockRepeaterStreams](#), [putFileInfo](#), [putStreamNameParts](#), [redirectSession](#),
[redirectSession](#), [rejectSession](#), [removeStreamDomainStr](#), [setAcceptSession](#), [setActive](#),
[setAppInstance](#), [setCookieStr](#), [setDvrSessionInfo](#), [setHTTPStreamerAdapter](#),
[setIpAddress](#), [setLiveStreamingPacketizer](#), [setPlayDuration](#), [setPlayLogged](#),
[setPlayStart](#), [setQueryStr](#), [setRedirectSession](#), [setRedirectSessionBody](#),
[setRedirectSessionCode](#), [setRedirectSessionContentType](#), [setRedirectSessionURL](#),
[setReferrer](#), [setServerIp](#), [setServerPort](#), [getSessionId](#), [setSessionProtocol](#),
[setSessionTimeout](#), [setSessionType](#), [setStream](#), [setStreamExt](#), [setStreamName](#),
[setStreamPosition](#), [setTimeoutSession](#), [setUri](#), [setUserAgent](#), [setUserHTTPHeader](#),
[setUserQueryStr](#), [setVHost](#), [shutdown](#), [touch](#), [updateLoggingValues](#),
[validStreamDomainToString](#)

Constructors

HTTPStreamerSessionCupertino

```
public HTTPStreamerSessionCupertino()
```

Methods

shutdown

```
public void shutdown()
```

containsIndex

```
public boolean containsIndex(String streamName)
```

(continued from last page)

doesFileExist

```
public static boolean doesFileExist(IHTTPStreamerApplicationContext appContext,  
    String rawStreamName,  
    String streamExt,  
    String streamName,  
    IHTTPStreamerSession httpStreamerSession)
```

getIndex

```
public IHTTPStreamerCupertinoIndex getIndex(IHTTPStreamerApplicationContext  
appContext,  
    IHTTPStreamerSession httpStreamerSession,  
    String rawStreamName,  
    String streamExt,  
    String streamName,  
    long playStart,  
    long playDuration)
```

updateLoggingValues

```
public void updateLoggingValues()
```

clearLoggingValues

```
public void clearLoggingValues()
```

logLiveChunk

```
public void logLiveChunk(LiveStreamPacketizerCupertinoChunk chunk)
```

logVODChunk

```
public void logVODChunk(LiveStreamPacketizerCupertinoChunk chunk)
```

com.wowza.wms.httpstreamer.cupertinostreaming.httpstreamer Interface IHTTPStreamerCupertinoVODActionNotify

All Subinterfaces:

[IHTTPStreamerCupertinoVODActionNotify2](#)

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](#)

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

Package

com.wowza.wms.httpstreamer.model

com.wowza.wms.httpstreamer.model Interface IHTTPStreamerAdapter

public interface **IHTTPStreamerAdapter**
extends

IHTTPStreamerAdapter: HTTP streaming adapter interface

Method Summary

boolean	<code>canHandle(String path)</code> Return true if can handle request
String	<code>getAdapterName()</code> Get the name of this adapter
HTTPStreamerItem	<code>getHTTPStreamerItem()</code> Get the HTTP streamer item associated with this adapter
String	<code>getID()</code> Get the id of this adapter
int	<code>getIdleFrequency()</code> Get the idle frequency (milliseconds) for HTTP requests.
<code>WMSProperties</code>	<code>getProperties()</code> Get properties
<code>IVHost</code>	<code>getVHost()</code> Get the vhost associated with this adapter
void	<code>init()</code> Initialize the HTTP streaming adapter
void	<code>service(org.apache.mina.common.io.Session session, RtmpRequestMessage req, RtmpResponseMessage resp)</code> Called to service each request
void	<code>setHTTPStreamerItem(HTTPStreamerItem httpStreamerItem)</code> Set the HTTP streamer item associated with this adapter
void	<code>setID(String id)</code> Set the id of this adapter
void	<code>setProperties(WMSProperties properties)</code> Set properties
void	<code>setVHost(IVHost vhost)</code> Set the vhost associated with this adapter
void	<code>shutdownSession(IApplicationInstance appInstance, IHTTPStreamerSession session)</code> 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 WMSProperties getProperties()
```

Get properties

Returns:

properties

setProperties

```
public void setProperties(WMSProperties properties)
```

Set properties

Parameters:

properties - properties

getHTTPStreamerItem

```
public HTTPStreamerItem getHTTPStreamerItem()
```

Get the HTTP streamer item associated with this adapter

Returns:

HTTP streamer item

setHTTPStreamerItem

```
public void setHTTPStreamerItem(HTTPStreamerItem httpStreamerItem)
```

(continued from last page)

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()
```

Get the name of this adapter

Returns:

name of this adapter

(continued from last page)

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

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.

Method Summary

IApplicationInstance	getAppInstance() Get application instance
WMSProperties	getProperties() Get properties
MediaCasterItem	getRepeaterMediaCasterDef() Get the live repeater media caster definition
String	getStreamTypeStr() Get stream type
IVHost	getVHost() Get vhost
void	init(IApplicationInstance appInstance, HTTPStreamerItem httpStreamerItem) Initialize context
boolean	isStreamDomainProtectionActive()
void	setRepeaterMediaCasterDef(MediaCasterItem repeaterMediaCasterDef) Set the live repeater media caster definition
void	setStreamDomainProtectionActive(boolean streamDomainProtectionActive)
void	setStreamTypeStr(String streamTypeStr) Set stream type

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

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)
```

(continued from last page)

Set stream type

Parameters:

streamTypeStr - stream type

isStreamDomainProtectionActive

```
public boolean isStreamDomainProtectionActive()
```

setStreamDomainProtectionActive

```
public void setStreamDomainProtectionActive(boolean streamDomainProtectionActive)
```

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 IHTTPStreamerSession

public interface **IHTTPStreamerSession**
extends

IHTTPStreamerSession: HTTP streaming session interface

Field Summary

public static final	SESSIONPROTOCOL_COUNT Value: 6
public static final	SESSIONPROTOCOL_CUPERTINOSTREAMING Value: 1
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	SESSIONTYPE_VOD Value: 2

Method Summary

void	<u>acceptSession()</u> Accept the HTTP session.
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>extractHTTPRequestInfo()</u> (<u>IHTTPRequest</u> req) Extract information from HTTP request
<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()</u> (String streamName) Get the file information if video on demand streaming
String	<u>getHTTPHeader()</u> (String name) Get a HTTP header value such as 'Content-Length'

java.util.Map	getHTTPHeaderMap() Get a copy of the HTTP request header map
java.util.Set	getHTTPHeaderNames() Get a Set of the header names
int	getHTTPIntHeader(String name) Get a HTTP header value such as 'Content-Length' and return as int
IHTTPStreamerAdapter	getHTTPStreamerAdapter() Get the HTTP streaming adapter associated with this HTTP session
IOPerformanceCounter	getIOPerformanceCounter() Get IO performance counter
String	getIpAddress() Get the IP address
long	getLastRequest() Get the last timestamp of the last Io request
String	getLiveStreamingPacketizer() Get the live stream packetizer name
Object	getLock() Get the synchronization lock for this HTTP session
long	getPlayDuration() Get the play duration (milliseconds) for video on demand playback.
long	getPlayStart() Get the play start time offset (milliseconds) for video on demand playback.
WMSProperties	getProperties() Get the properties associated with this session
String	getQueryStr() Get query string
byte[]	getRedirectSessionBody() Get redirect session body
int	getRedirectSessionCode() Get session redirect HTTP response code (default 302)
String	getRedirectSessionContentType() Get redirect session HTTP Content-Type
String	getRedirectSessionURL() Get redirect session URL
String	getReferrer() Get referrer
String	getServerIp() Get server IP address
int	getServerPort() Get server port

String	<code>getSessionId()</code> Get session id
int	<code>getSessionProtocol()</code> Get protocol, see SESSIONPROTOCOL_*
int	<code>getSessionTimeout()</code> Get the session timeout for this session (milliseconds)
int	<code>getSessionType()</code> Get session type: see SESSIONTYPE_*
<code>IMediaStream</code>	<code>getStream()</code> Get the IMediaStream associated with this HTTP session
String	<code>getStreamExt()</code> Get stream extension
String	<code>getStreamName()</code> Get stream name
HTTPStreamerStreamNameParts	<code>getStreamNameParts(String streamName)</code> Break the stream name into parts
long	<code>getStreamPosition()</code> Get stream position
String	<code>getTimeRunning()</code> Get the time this session has been running (milliseconds)
double	<code>getTimeRunningSeconds()</code> Get the time this session has been running (seconds)
String	<code>getUri()</code> Get the URI associated with initial request
String	<code>getUserAgent()</code> Get user agent
java.util.Map	<code>getUserHTTPHeaders()</code> Get user HTTP header.
String	<code>getUserQueryString()</code> This query string will be added to URLs used in HTTP streaming
<code>IVHost</code>	<code>getVHost()</code> Get vhost
boolean	<code>isAcceptSession()</code> Return true if this session has not been rejected
boolean	<code>isActive()</code> Is this session active, false after shutdown
boolean	<code>isFileInfo(String streamName)</code> Return true if the is file information for a given stream name
boolean	<code>isPlayLogged()</code> true, if play has been logged

boolean	<u>isRedirectSession()</u> Is session redirect
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(String streamDomainStr)</u> Internal user, is stream name valid for HTTP session
void	<u>lockRepeaterStreams(java.util.List streamNames, String liveStreamPacketizer, String liveStreamRepeater, String streamTypeStr)</u> Internal user, lock in reapter streams
void	<u>putFileInfo(String streamName, HTTPStreamerFileInfo fileInfo)</u> Set the file information
void	<u>putStreamNameParts(String streamName, HTTPStreamerStreamNameParts streamNameParts)</u> Add stream name to stream name parts
void	<u>redirectSession(String redirectSessionURL)</u> Redirect session
void	<u>redirectSession(String redirectSessionURL, int redirectSessionCode)</u> Redirect session
void	<u>rejectSession()</u> Reject this HTTP session.
void	<u>removeStreamDomainStr(String streamDomainStr)</u> Internal user, remove stream name
void	<u>setAcceptSession(boolean acceptSession)</u> Set to false to reject session
void	<u>setActive(boolean isActive)</u> Set session active
void	<u>setAppInstance(IApplicationInstance appInstance)</u> Set the application instance associated with this HTTP session
void	<u>setCookieStr(String cookieStr)</u> Set cookie string
void	<u>setDvrSessionInfo(DvrSessionInfo dvr)</u> This information is used to manage the connection to the DVR store.
void	<u>setHTTPStreamerAdapter(IHTTPStreamerAdapter httpStreamerAdapter)</u> Set the HTTP streaming adapter associated with this HTTP session
void	<u>setIpAddress(String ipAddress)</u> 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>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>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>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>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**

SESSIONTYPE_VOD

```
public static final int SESSIONTYPE_VOD
```

Constant value: **2**

(continued from last page)

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
```

Constant value: **5**

SESSIONPROTOCOL_COUNT

```
public static final int SESSIONPROTOCOL_COUNT
```

(continued from last page)

Constant value: **6**

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)
```

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();

(continued from last page)

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()
```

(continued from last page)

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()
```

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()
```

(continued from last page)

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)
```

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

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()
```

(continued from last page)

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

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()
```

(continued from last page)

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

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

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

(continued from last page)

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()
```

(continued from last page)

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)
```

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

Parameters:

name - header name

Returns:

header value

getHTTPHeaderNames

```
public java.util.Set getHTTPHeaderNames()
```

Get a Set of the header names

(continued from last page)

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)

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

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()
```

(continued from last page)

Get redirect session URL

Returns:

redirect session URL

setRedirectSessionURL

```
public void setRedirectSessionURL(String redirectSessionURL)
```

Set redirect session URL

Parameters:

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

(continued from last page)

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)

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 **ILiveStreamPacketizerChunkWriter**

public interface **ILiveStreamPacketizerChunkWriter**
extends

Method Summary

void	<u>addToChunk</u> (LiveStreamPacketizerPacketHolder holder)
void	<u>endChunk</u> (long timecode)
boolean	<u>isPacketizeAudio</u> ()
boolean	<u>isPacketizeData</u> ()
boolean	<u>isPacketizeVideo</u> ()
boolean	<u>isValidAudioCodec</u> (int codec)
boolean	<u>isValidVideoCodec</u> (int codec)
void	<u>resetStream</u> ()
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)

Methods

startChunk

```
public void startChunk(int streamMode,  
    int videoCodecId,  
    int audioCodecId,  
    long timecode)
```

endChunk

```
public void endChunk(long timecode)
```

(continued from last page)

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)
```

isValidVideoCodec

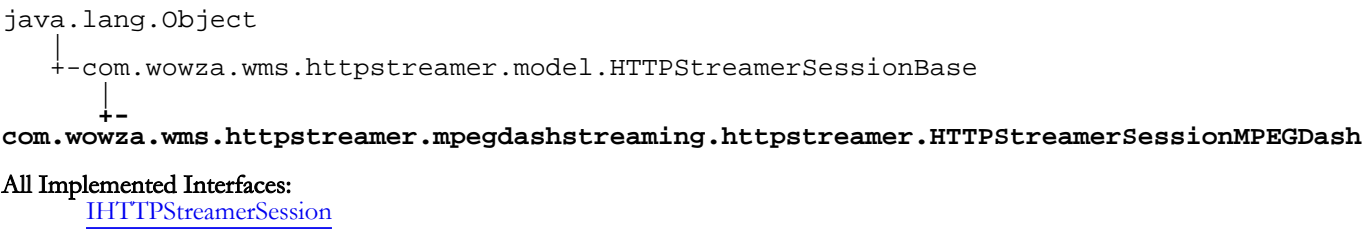
```
public boolean isValidVideoCodec(int codec)
```

Package

com.wowza.wms.httpstreamer.mpegdashstreaming.httpstreamer

com.wowza.wms.httpstreamer.mpegdashstreaming.httpstreamer

Class HTTPStreamerSessionMPGEDash



```
public class HTTPStreamerSessionMPGEDash
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, isPlayLogged, lastRequest, liveStreamingPacketizer, lock, mediaCasterStreamLock, playDuration, playStart, properties, queryStr, redirectSession, redirectSessionBody, redirectSessionCode, redirectSessionContentType, redirectSessionURL, referrer, serverIp, serverPort, sessionId, sessionProtocol, sessionTimeout, sessionType, stream, streamDomainStrSet, streamExt, streamName, streamNamePartMap, streamPosition, timeoutSession, totalIOPerformance2Last, totalIOPerformanceLast, uri, userAgent, userHTTPHeaders, userQueryStr, vhost

Fields inherited from interface com.wowza.wms.httpstreamer.model.IHTTPStreamerSession
SESSIONPROTOCOL_COUNT , SESSIONPROTOCOL_CUPERTINOSTREAMING , SESSIONPROTOCOL_DVRCHUNKSTREAMING , SESSIONPROTOCOL_MPEGDASHSTREAMING , SESSIONPROTOCOL_SANJOSESTREAMING , SESSIONPROTOCOL_SMOOTHSTREAMING , SESSIONPROTOCOL_UNKNOWN , SESSIONPROTOCOL_WEBMSTREAMING , SESSIONTYPE_LIVE , SESSIONTYPE_LIVEDVR , SESSIONTYPE_UNKNOWN , SESSIONTYPE_VOD

Constructor Summary	
public	HTTPStreamerSessionMPGEDash()

Method Summary	
void	logLiveChunk (MPGEDashDashChunk chunk)

Methods inherited from class com.wowza.wms.httpstreamer.model.HTTPStreamerSessionBase

```
acceptSession, addIOPerformance, addIOPerformance2, addStreamDomainStr,
addStreamDomainStrs, addUserHTTPHeaders, checkAndSetPlayLogged, clearLoggingValues,
containsStreamDomainStr, containsStreamNameParts, doSessionRedirect,
extractHTTPRequestInfo, getAppInstance, getConnectionHolder, getCookieStr,
getDvrSessionInfo, getElapsedTime, getFileInfo, getHTTPDate, getHTTPHeader,
getHTTPHeaderMap, getHTTPHeaderNames, getHTTPIntHeader, getHTTPStreamerAdapter,
getIOPerformanceCounter, getIpAddress, getLastRequest, getLiveStreamingPacketizer,
getLock, getPlayDuration, getPlayStart, getProperties, getQueryStr,
getRedirectSessionBody, getRedirectSessionCode, getRedirectSessionContentType,
getRedirectSessionURL, getReferrer, getServerIp, getServerPort, getSessionId,
getSessionProtocol, getSessionTimeout, getSessionType, getStream, getStreamDomainStr,
getStreamDomainStrList, getStreamExt, getStreamName, getStreamNameParts,
getStreamPosition, getTimeRunning, getTimeRunningSeconds, getUri, getUserAgent,
getUserHTTPHeaders, getUserQueryStr, getVHost, isAcceptSession, isActive, isFileInfo,
isPlayLogged, isRedirectSession, isTimeout, isTimeoutSession, isValidated,
isValidStreamDomainStr, lockRepeaterStreams, putFileInfo, putStreamNameParts,
redirectSession, redirectSession, rejectSession, removeStreamDomainStr,
setAcceptSession, setActive, setAppInstance, setCookieStr, setDvrSessionInfo,
setHTTPStreamerAdapter, setIpAddress, setLiveStreamingPacketizer, setPlayDuration,
setPlayLogged, setPlayStart, setQueryStr, setRedirectSession, setRedirectSessionBody,
setRedirectSessionCode, setRedirectSessionContentType, setRedirectSessionURL,
setReferrer, setServerIp, setServerPort, setSessionId, setSessionProtocol,
setSessionTimeout, setSessionType, setStream, setStreamExt, setStreamName,
setStreamPosition, setThreadContext, setTimeoutSession, setUri, setUserAgent,
setUserHTTPHeader, setUserQueryStr, setValidated, setVHost, shutdown, 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](#), [addIOPerformance](#), [addIOPerformance2](#), [addStreamDomainStr](#),
[addStreamDomainStrs](#), [addUserHTTPHeaders](#), [checkAndSetPlayLogged](#), [clearLoggingValues](#),
[containsStreamDomainStr](#), [containsStreamNameParts](#), [doSessionRedirect](#),
[extractHTTPRequestInfo](#), [getAppInstance](#), [getConnectionHolder](#), [getCookieStr](#),
[getDvrSessionInfo](#), [getElapsedTime](#), [getFileInfo](#), [getHTTPHeader](#), [getHTTPHeaderMap](#),
[getHTTPHeaderNames](#), [getHTTPIntHeader](#), [getHTTPStreamerAdapter](#),
[getIOPerformanceCounter](#), [getIpAddress](#), [getLastRequest](#), [getLiveStreamingPacketizer](#),
[getLock](#), [getPlayDuration](#), [getPlayStart](#), [getProperties](#), [getQueryStr](#),
[getRedirectSessionBody](#), [getRedirectSessionCode](#), [getRedirectSessionContentType](#),
[getRedirectSessionURL](#), [getReferrer](#), [getServerIp](#), [getServerPort](#), [getSessionId](#),
[getSessionProtocol](#), [getSessionTimeout](#), [getSessionType](#), [getStream](#), [getStreamExt](#),
[getStreamName](#), [getStreamNameParts](#), [getStreamPosition](#), [getTimeRunning](#),
[getTimeRunningSeconds](#), [getUri](#), [getUserAgent](#), [getUserHTTPHeaders](#), [getUserQueryStr](#),
[getVHost](#), [isAcceptSession](#), [isActive](#), [isFileInfo](#), [isPlayLogged](#), [isRedirectSession](#),
[isTimeout](#), [isTimeoutSession](#), [isValidated](#), [isValidStreamDomainStr](#),
[lockRepeaterStreams](#), [putFileInfo](#), [putStreamNameParts](#), [redirectSession](#),
[redirectSession](#), [rejectSession](#), [removeStreamDomainStr](#), [setAcceptSession](#), [setActive](#),
[setAppInstance](#), [setCookieStr](#), [setDvrSessionInfo](#), [setHTTPStreamerAdapter](#),
[setIpAddress](#), [setLiveStreamingPacketizer](#), [setPlayDuration](#), [setPlayLogged](#),
[setPlayStart](#), [setQueryStr](#), [setRedirectSession](#), [setRedirectSessionBody](#),
[setRedirectSessionCode](#), [setRedirectSessionContentType](#), [setRedirectSessionURL](#),
[setReferrer](#), [setServerIp](#), [setServerPort](#), [getSessionId](#), [setSessionProtocol](#),
[setSessionTimeout](#), [setSessionType](#), [setStream](#), [setStreamExt](#), [setStreamName](#),
[setStreamPosition](#), [setTimeoutSession](#), [setUri](#), [setUserAgent](#), [setUserHTTPHeader](#),
[setUserQueryStr](#), [setVHost](#), [shutdown](#), [touch](#), [updateLoggingValues](#),
[validStreamDomainToString](#)

Constructors

HTTPStreamerSessionMPEGDash

```
public HTTPStreamerSessionMPEGDash()
```

Methods

logLiveChunk

```
public void logLiveChunk(MPEGDashDashChunk chunk)
```

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:

[IHTTPStreamerSession](#)

public class **HTTPStreamerSessionSanJose**
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, isPlayLogged, lastRequest, liveStreamingPacketizer, lock, mediaCasterStreamLock, playDuration, playStart, properties, queryStr, redirectSession, redirectSessionBody, redirectSessionCode, redirectSessionContentType, redirectSessionURL, referrer, serverIp, serverPort, sessionId, sessionProtocol, sessionTimeout, sessionType, stream, streamDomainStrSet, streamExt, streamName, streamNamePartMap, streamPosition, timeoutSession, totalIOPerformance2Last, totalIOPerformanceLast, uri, userAgent, userHTTPHeaders, userQueryStr, vhost

Fields inherited from interface [com.wowza.wms.httpstreamer.model.IHTTPStreamerSession](#)

[SESSIONPROTOCOL_COUNT](#), [SESSIONPROTOCOL_CUPERTINOSTREAMING](#), [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)
IHTTPStreamerSanJoseIndex	getIndexLive(IHTTPStreamerApplicationContext appContext, IHTTPStreamerSession httpStreamerSession, String rawStreamName, String streamExt, String streamName)
boolean	isFirstABSTRequest (String streamName)
void	logLiveChunk (LiveStreamPacketizerSanJoseChunk chunk)
void	logVODChunk (LiveStreamPacketizerSanJoseChunk chunk)
void	shutdown ()
void	updateLoggingValues ()

Methods inherited from class `com.wowza.wms.httpstreamer.model.HTTPStreamerSessionBase`

acceptSession, addIOPerformance, addIOPerformance2, addStreamDomainStr, addStreamDomainStrs, addUserHTTPHeaders, checkAndSetPlayLogged, clearLoggingValues, containsStreamDomainStr, containsStreamNameParts, doSessionRedirect, extractHTTPRequestInfo, getAppInstance, getConnectionHolder, getCookieStr, getDvrSessionInfo, getElapsedTime, getFileInfo, getHTTPDate, getHTTPHeader, getHTTPHeaderMap, getHTTPHeaderNames, getHTTPIntHeader, getHTTPStreamerAdapter, getIOPerformanceCounter, getIpAddress, getLastRequest, getLiveStreamingPacketizer, getLock, getPlayDuration, getPlayStart, getProperties, getQueryStr, getRedirectSessionBody, getRedirectSessionCode, getRedirectSessionContentType, getRedirectSessionURL, getReferrer, getServerIp, getServerPort, getSessionId, getSessionProtocol, getSessionTimeout, getSessionType, getStream, getStreamDomainStr, getStreamDomainStrList, getStreamExt, getStreamName, getStreamNameParts, getStreamPosition, getTimeRunning, getTimeRunningSeconds, getUri, getUserAgent, getUserHTTPHeaders, getUserQueryStr, getVHost, isAcceptSession, isActive, isFileInfo, isPlayLogged, isRedirectSession, isTimeout, isTimeoutSession, isValidated, isValidStreamDomainStr, lockRepeaterStreams, putFileInfo, putStreamNameParts, redirectSession, redirectSession, rejectSession, removeStreamDomainStr, setAcceptSession, setActive, setAppInstance, setCookieStr, setDvrSessionInfo, setHTTPStreamerAdapter, setIpAddress, setLiveStreamingPacketizer, setPlayDuration, setPlayLogged, setPlayStart, setQueryStr, setRedirectSession, setRedirectSessionBody, setRedirectSessionCode, setRedirectSessionContentType, setRedirectSessionURL, setReferrer, setServerIp, setServerPort, setSessionId, setSessionProtocol, setSessionTimeout, setSessionType, setStream, setStreamExt, setStreamName, setStreamPosition, setThreadContext, setTimeoutSession, setUri, setUserAgent, setUserHTTPHeader, setUserQueryStr, setValidated, setVHost, shutdown, 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](#), [addIOPerformance](#), [addIOPerformance2](#), [addStreamDomainStr](#),
[addStreamDomainStrs](#), [addUserHTTPHeaders](#), [checkAndSetPlayLogged](#), [clearLoggingValues](#),
[containsStreamDomainStr](#), [containsStreamNameParts](#), [doSessionRedirect](#),
[extractHTTPRequestInfo](#), [getAppInstance](#), [getConnectionHolder](#), [getCookieStr](#),
[getDvrSessionInfo](#), [getElapsedTime](#), [getFileInfo](#), [getHTTPHeader](#), [getHTTPHeaderMap](#),
[getHTTPHeaderNames](#), [getHTTPIntHeader](#), [getHTTPStreamerAdapter](#),
[getIOPerformanceCounter](#), [getIpAddress](#), [getLastRequest](#), [getLiveStreamingPacketizer](#),
[getLock](#), [getPlayDuration](#), [getPlayStart](#), [getProperties](#), [getQueryStr](#),
[getRedirectSessionBody](#), [getRedirectSessionCode](#), [getRedirectSessionContentType](#),
[getRedirectSessionURL](#), [getReferrer](#), [getServerIp](#), [getServerPort](#), [getSessionId](#),
[getSessionProtocol](#), [getSessionTimeout](#), [getSessionType](#), [getStream](#), [getStreamExt](#),
[getStreamName](#), [getStreamNameParts](#), [getStreamPosition](#), [getTimeRunning](#),
[getTimeRunningSeconds](#), [getUri](#), [getUserAgent](#), [getUserHTTPHeaders](#), [getUserQueryStr](#),
[getVHost](#), [isAcceptSession](#), [isActive](#), [isFileInfo](#), [isPlayLogged](#), [isRedirectSession](#),
[isTimeout](#), [isTimeoutSession](#), [isValidated](#), [isValidStreamDomainStr](#),
[lockRepeaterStreams](#), [putFileInfo](#), [putStreamNameParts](#), [redirectSession](#),
[redirectSession](#), [rejectSession](#), [removeStreamDomainStr](#), [setAcceptSession](#), [setActive](#),
[setAppInstance](#), [setCookieStr](#), [setDvrSessionInfo](#), [setHTTPStreamerAdapter](#),
[setIpAddress](#), [setLiveStreamingPacketizer](#), [setPlayDuration](#), [setPlayLogged](#),
[setPlayStart](#), [setQueryStr](#), [setRedirectSession](#), [setRedirectSessionBody](#),
[setRedirectSessionCode](#), [setRedirectSessionContentType](#), [setRedirectSessionURL](#),
[setReferrer](#), [setServerIp](#), [setServerPort](#), [getSessionId](#), [setSessionProtocol](#),
[setSessionTimeout](#), [setSessionType](#), [setStream](#), [setStreamExt](#), [setStreamName](#),
[setStreamPosition](#), [setTimeoutSession](#), [setUri](#), [setUserAgent](#), [setUserHTTPHeader](#),
[setUserQueryStr](#), [setVHost](#), [shutdown](#), [touch](#), [updateLoggingValues](#),
[validStreamDomainToString](#)

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)
```

updateLoggingValues

```
public void updateLoggingValues()
```

clearLoggingValues

```
public void clearLoggingValues()
```

logLiveChunk

```
public void logLiveChunk(LiveStreamPacketizerSanJoseChunk chunk)
```

logVODChunk

```
public void logVODChunk(LiveStreamPacketizerSanJoseChunk chunk)
```

(continued from last page)

isFirstABSTRequest

```
public boolean isFirstABSTRequest(String streamName)
```

addFirstABSTRequest

```
public void addFirstABSTRequest(String streamName)
```

Package

**com.wowza.wms.httpstreamer.smoothstream
ing.httpstreamer**

com.wowza.wms.httpstreamer.smoothstreaming.httpstreamer Class HTTPStreamerSessionSmoothStreamer

java.lang.Object

└─com.wowza.wms.httpstreamer.model.HTTPStreamerSessionBase

└─

com.wowza.wms.httpstreamer.smoothstreaming.httpstreamer.HTTPStreamerSessionSmoothStreamer

All Implemented Interfaces:

[IHTTPStreamerSession](#)

```
public class HTTPStreamerSessionSmoothStreamer
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, isPlayLogged, lastRequest, liveStreamingPacketizer, lock, mediaCasterStreamLock, playDuration, playStart, properties, queryStr, redirectSession, redirectSessionBody, redirectSessionCode, redirectSessionContentType, redirectSessionURL, referrer, serverIp, serverPort, sessionId, sessionProtocol, sessionTimeout, sessionType, stream, streamDomainStrSet, streamExt, streamName, streamNamePartMap, streamPosition, timeoutSession, totalIOPerformance2Last, totalIOPerformanceLast, uri, userAgent, userHTTPHeaders, userQueryStr, vhost

Fields inherited from interface [com.wowza.wms.httpstreamer.model.IHTTPStreamerSession](#)

[SESSIONPROTOCOL_COUNT](#), [SESSIONPROTOCOL_CUPERTINOSTREAMING](#), [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)
SmoothStreamingLivePlaylist	getLivePlaylist()

void	<u>logLiveFragment</u> (SmoothStreamerFragmentId fragmentId, PacketFragmentList fragmentData)
void	<u>logVODFragment</u> (SmoothStreamerFragmentId fragmentId, PacketFragmentList fragmentData)
void	<u>setLivePlaylist</u> (SmoothStreamingLivePlaylist livePlaylist)
void	<u>shutdown</u> ()
void	<u>updateLoggingValues</u> ()

Methods inherited from class `com.wowza.wms.httpstreamer.model.HTTPStreamerSessionBase`

acceptSession, addIOPerformance, addIOPerformance2, addStreamDomainStr, addStreamDomainStrs, addUserHTTPHeaders, checkAndSetPlayLogged, clearLoggingValues, containsStreamDomainStr, containsStreamNameParts, doSessionRedirect, extractHTTPRequestInfo, getAppInstance, getConnectionHolder, getCookieStr, getDvrSessionInfo, getElapsedTime, getFileInfo, getHTTPDate, getHTTPHeader, getHTTPHeaderMap, getHTTPHeaderNames, getHTTPIntHeader, getHTTPStreamerAdapter, getIOPerformanceCounter, getIpAddress, getLastRequest, getLiveStreamingPacketizer, getLock, getPlayDuration, getPlayStart, getProperties, getQueryStr, getRedirectSessionBody, getRedirectSessionCode, getRedirectSessionContentType, getRedirectSessionURL, getReferrer, getServerIp, getServerPort, getSessionId, getSessionProtocol, getSessionTimeout, getSessionType, getStream, getStreamDomainStr, getStreamDomainStrList, getStreamExt, getStreamName, getStreamNameParts, getStreamPosition, getTimeRunning, getTimeRunningSeconds, getUri, getUserAgent, getUserHTTPHeaders, getUserQueryStr, getVHost, isAcceptSession, isActive, isFileInfo, isPlayLogged, isRedirectSession, isTimeout, isTimeoutSession, isValidated, isValidStreamDomainStr, lockRepeaterStreams, putFileInfo, putStreamNameParts, redirectSession, redirectSession, rejectSession, removeStreamDomainStr, setAcceptSession, setActive, setAppInstance, setCookieStr, setDvrSessionInfo, setHTTPStreamerAdapter, setIpAddress, setLiveStreamingPacketizer, setPlayDuration, setPlayLogged, setPlayStart, setQueryStr, setRedirectSession, setRedirectSessionBody, setRedirectSessionCode, setRedirectSessionContentType, setRedirectSessionURL, setReferrer, setServerIp, setServerPort, setSessionId, setSessionProtocol, setSessionTimeout, setSessionType, setStream, setStreamExt, setStreamName, setStreamPosition, setThreadContext, setTimeoutSession, setUri, setUserAgent, setUserHTTPHeader, setUserQueryStr, setValidated, setVHost, shutdown, 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](#), [addIOPerformance](#), [addIOPerformance2](#), [addStreamDomainStr](#), [addStreamDomainStrs](#), [addUserHTTPHeaders](#), [checkAndSetPlayLogged](#), [clearLoggingValues](#), [containsStreamDomainStr](#), [containsStreamNameParts](#), [doSessionRedirect](#), [extractHTTPRequestInfo](#), [getAppInstance](#), [getConnectionHolder](#), [getCookieStr](#), [getDvrSessionInfo](#), [getElapsedTime](#), [getFileInfo](#), [getHTTPHeader](#), [getHTTPHeaderMap](#), [getHTTPHeaderNames](#), [getHTTPIntHeader](#), [getHTTPStreamerAdapter](#), [getIOPerformanceCounter](#), [getIpAddress](#), [getLastRequest](#), [getLiveStreamingPacketizer](#), [getLock](#), [getPlayDuration](#), [getPlayStart](#), [getProperties](#), [getQueryStr](#), [getRedirectSessionBody](#), [getRedirectSessionCode](#), [getRedirectSessionContentType](#), [getRedirectSessionURL](#), [getReferrer](#), [getServerIp](#), [getServerPort](#), [getSessionId](#), [getSessionProtocol](#), [getSessionTimeout](#), [getSessionType](#), [getStream](#), [getStreamExt](#), [getStreamName](#), [getStreamNameParts](#), [getStreamPosition](#), [getTimeRunning](#), [getTimeRunningSeconds](#), [getUri](#), [getUserAgent](#), [getUserHTTPHeaders](#), [getUserQueryStr](#), [getVHost](#), [isAcceptSession](#), [isActive](#), [isFileInfo](#), [isPlayLogged](#), [isRedirectSession](#), [isTimeout](#), [isTimeoutSession](#), [isValidated](#), [isValidStreamDomainStr](#), [lockRepeaterStreams](#), [putFileInfo](#), [putStreamNameParts](#), [redirectSession](#), [redirectSession](#), [rejectSession](#), [removeStreamDomainStr](#), [setAcceptSession](#), [setActive](#), [setAppInstance](#), [setCookieStr](#), [setDvrSessionInfo](#), [setHTTPStreamerAdapter](#), [setIpAddress](#), [setLiveStreamingPacketizer](#), [setPlayDuration](#), [setPlayLogged](#), [setPlayStart](#), [setQueryStr](#), [setRedirectSession](#), [setRedirectSessionBody](#), [setRedirectSessionCode](#), [setRedirectSessionContentType](#), [setRedirectSessionURL](#), [setReferrer](#), [setServerIp](#), [setServerPort](#), [setSessionId](#), [setSessionProtocol](#), [setSessionTimeout](#), [setSessionType](#), [setStream](#), [setStreamExt](#), [setStreamName](#), [setStreamPosition](#), [setTimeoutSession](#), [setUri](#), [setUserAgent](#), [setUserHTTPHeader](#), [setUserQueryStr](#), [setVHost](#), [shutdown](#), [touch](#), [updateLoggingValues](#), [validStreamDomainToString](#)

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)
```

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)
```

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, isPlayLogged, lastRequest, liveStreamingPacketizer, lock, mediaCasterStreamLock, playDuration, playStart, properties, queryStr, redirectSession, redirectSessionBody, redirectSessionCode, redirectSessionContentType, redirectSessionURL, referrer, serverIp, serverPort, sessionId, sessionProtocol, sessionTimeout, sessionType, stream, streamDomainStrSet, streamExt, streamName, streamNamePartMap, streamPosition, timeoutSession, totalIOPerformance2Last, totalIOPerformanceLast, uri, userAgent, userHTTPHeaders, userQueryStr, vhost

Fields inherited from interface [com.wowza.wms.httpstreamer.model.IHTTPStreamerSession](#)

[SESSIONPROTOCOL_COUNT](#), [SESSIONPROTOCOL_CUPERTINOSTREAMING](#), [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, addIOPerformance, addIOPerformance2, addStreamDomainStr,
addStreamDomainStrs, addUserHTTPHeaders, checkAndSetPlayLogged, clearLoggingValues,
containsStreamDomainStr, containsStreamNameParts, doSessionRedirect,
extractHTTPRequestInfo, getAppInstance, getConnectionHolder, getCookieStr,
getDvrSessionInfo, getElapsedTime, getFileInfo, getHTTPDate, getHTTPHeader,
getHTTPHeaderMap, getHTTPHeaderNames, getHTTPIntHeader, getHTTPStreamerAdapter,
getIOPerformanceCounter, getIpAddress, getLastRequest, getLiveStreamingPacketizer,
getLock, getPlayDuration, getPlayStart, getProperties, getQueryStr,
getRedirectSessionBody, getRedirectSessionCode, getRedirectSessionContentType,
getRedirectSessionURL, getReferrer, getServerIp, getServerPort, getSessionId,
getSessionProtocol, getSessionTimeout, getSessionType, getStream, getStreamDomainStr,
getStreamDomainStrList, getStreamExt, getStreamName, getStreamNameParts,
getStreamPosition, getTimeRunning, getTimeRunningSeconds, getUri, getUserAgent,
getUserHTTPHeaders, getUserQueryStr, getVHost, isAcceptSession, isActive, isFileInfo,
isPlayLogged, isRedirectSession, isTimeout, isTimeoutSession, isValidated,
isValidStreamDomainStr, lockRepeaterStreams, putFileInfo, putStreamNameParts,
redirectSession, redirectSession, rejectSession, removeStreamDomainStr,
setAcceptSession, setActive, setAppInstance, setCookieStr, setDvrSessionInfo,
setHTTPStreamerAdapter, setIpAddress, setLiveStreamingPacketizer, setPlayDuration,
setPlayLogged, setPlayStart, setQueryStr, setRedirectSession, setRedirectSessionBody,
setRedirectSessionCode, setRedirectSessionContentType, setRedirectSessionURL,
setReferrer, setServerIp, setServerPort, setSessionId, setSessionProtocol,
setSessionTimeout, setSessionType, setStream, setStreamExt, setStreamName,
setStreamPosition, setThreadContext, setTimeoutSession, setUri, setUserAgent,
setUserHTTPHeader, setUserQueryStr, setValidated, setVHost, shutdown, 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](#), [addIOPerformance](#), [addIOPerformance2](#), [addStreamDomainStr](#),
[addStreamDomainStrs](#), [addUserHTTPHeaders](#), [checkAndSetPlayLogged](#), [clearLoggingValues](#),
[containsStreamDomainStr](#), [containsStreamNameParts](#), [doSessionRedirect](#),
[extractHTTPRequestInfo](#), [getAppInstance](#), [getConnectionHolder](#), [getCookieStr](#),
[getDvrSessionInfo](#), [getElapsedTime](#), [getFileInfo](#), [getHTTPHeader](#), [getHTTPHeaderMap](#),
[getHTTPHeaderNames](#), [getHTTPIntHeader](#), [getHTTPStreamerAdapter](#),
[getIOPerformanceCounter](#), [getIpAddress](#), [getLastRequest](#), [getLiveStreamingPacketizer](#),
[getLock](#), [getPlayDuration](#), [getPlayStart](#), [getProperties](#), [getQueryStr](#),
[getRedirectSessionBody](#), [getRedirectSessionCode](#), [getRedirectSessionContentType](#),
[getRedirectSessionURL](#), [getReferrer](#), [getServerIp](#), [getServerPort](#), [getSessionId](#),
[getSessionProtocol](#), [getSessionTimeout](#), [getSessionType](#), [getStream](#), [getStreamExt](#),
[getStreamName](#), [getStreamNameParts](#), [getStreamPosition](#), [getTimeRunning](#),
[getTimeRunningSeconds](#), [getUri](#), [getUserAgent](#), [getUserHTTPHeaders](#), [getUserQueryStr](#),
[getVHost](#), [isAcceptSession](#), [isActive](#), [isFileInfo](#), [isPlayLogged](#), [isRedirectSession](#),
[isTimeout](#), [isTimeoutSession](#), [isValidated](#), [isValidStreamDomainStr](#),
[lockRepeaterStreams](#), [putFileInfo](#), [putStreamNameParts](#), [redirectSession](#),
[redirectSession](#), [rejectSession](#), [removeStreamDomainStr](#), [setAcceptSession](#), [setActive](#),
[setAppInstance](#), [setCookieStr](#), [setDvrSessionInfo](#), [setHTTPStreamerAdapter](#),
[setIpAddress](#), [setLiveStreamingPacketizer](#), [setPlayDuration](#), [setPlayLogged](#),
[setPlayStart](#), [setQueryStr](#), [setRedirectSession](#), [setRedirectSessionBody](#),
[setRedirectSessionCode](#), [setRedirectSessionContentType](#), [setRedirectSessionURL](#),
[setReferrer](#), [setServerIp](#), [setServerPort](#), [getSessionId](#), [setSessionProtocol](#),
[setSessionTimeout](#), [setSessionType](#), [setStream](#), [setStreamExt](#), [setStreamName](#),
[setStreamPosition](#), [setTimeoutSession](#), [setUri](#), [setUserAgent](#), [setUserHTTPHeader](#),
[setUserQueryStr](#), [setVHost](#), [shutdown](#), [touch](#), [updateLoggingValues](#),
[validStreamDomainToString](#)

Constructors

HTTPStreamerSessionWebM

```
public HTTPStreamerSessionWebM()
```

Methods

getSessionTracker

```
public WebMPacketizerSessionTracker getSessionTracker()
```

setSessionTracker

```
public void setSessionTracker(WebMPacketizerSessionTracker sessionTracker)
```

shutdown

```
public void shutdown()
```

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	<u>HOSTNAME_KEY</u> hostname string literal. Value: hostname
public static final	<u>ISO8601_FORMAT</u> ISO8601 string literal. Value: ISO8601
public static final	<u>ISO8601_PATTERN</u> SimpleTimePattern for ISO8601. Value: yyyy-MM-dd HH:mm:ss,SSS
public static final	<u>JNDI_CONTEXT_NAME</u> JNDI context name string literal. Value: java:comp/env/log4j/context-name
public static final	<u>LOG4J_ID_KEY</u> log4jid string literal. Value: log4jid
public static final	<u>LOG4J_PACKAGE_NAME</u> log4j package name string literal. Value: org.apache.log4j
public static final	<u>RECEIVER_NAME_KEY</u> receiver string literal. Value: receiver
public static final	<u>TEMP_CONSOLE_APPENDER_NAME</u> TEMP_CONSOLE_APPENDER string literal. Value: TEMP_CONSOLE_APPENDER
public static final	<u>TEMP_LIST_APPENDER_NAME</u> TEMP_LIST_APPENDER string literal. Value: TEMP_LIST_APPENDER
public static final	<u>TIMESTAMP_RULE_FORMAT</u> 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

If on Windows, also edit `[install-dir]/bin/WowzaMediaServerPro-Service.conf` and add `wrapper.java.additional.[n]=-Dcom.wowza.wms.logging.LogNotify=[full-path-to-your-ILogNotify-class]` to the **Java Additional Parameters** section where **[n]** is the next number in the list of active parameters. For example for the class above if the last active additional parameter is 6, the entry would look like this:

wrapper.java.additional.7=-

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	<code>onLog(org.apache.log4j.Level level, String comment, IMediaStream stream, String category, String event, int status, String context)</code> 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

(continued from last page)

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 (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	error (String comment)
void	error (String comment, IMediaStream stream)
void	error (String comment, IMediaStream stream, String category, String event, int status, String context)
void	error (String comment, String category, String event)

void	error (String comment, String category, String event, int status, String context)
void	fatal (String comment)
void	fatal (String comment, IMediaStream stream)
void	fatal (String comment, IMediaStream stream, String category, String event, int status, String context)
void	fatal (String comment, String category, String event)
void	fatal (String comment, String category, String event, int status, String context)
static WMSLogger	getLogger (String name)
void	info (String comment)
void	info (String comment, IMediaStream stream)
void	info (String comment, IMediaStream stream, String category, String event, int status, String context)
void	info (String comment, String category, String event)
void	info (String comment, String category, String event, int status, String context)
boolean	isDebugEnabled ()
boolean	isEnabledFor (org.apache.log4j.Priority level)
boolean	isErrorEnabled ()
boolean	isInfoEnabled ()
boolean	isTraceEnabled ()
boolean	isWarnEnabled ()
void	log (org.apache.log4j.Level level, String comment)
void	log (org.apache.log4j.Level level, String comment, IMediaStream stream)
void	log (org.apache.log4j.Level level, String comment, IMediaStream stream, String category, String event)
void	log (org.apache.log4j.Level level, String comment, IMediaStream stream, String category, String event, int status, String context)

void	log (org.apache.log4j.Level level, String comment, String category, String event)
void	warn (String comment)
void	warn (String comment, IMediaStream stream)
void	warn (String comment, IMediaStream stream, String category, String event, int status, String context)
void	warn (String comment, String category, String event)
void	warn (String comment, String category, String event, int status, String context)

Methods inherited from class org.apache.log4j.Logger

getLogger, getLogger, getLogger, getRootLogger, 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)
```

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)
```

(continued from last page)

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(String comment)
```

info

```
public void info(String comment,
               IMediaStream stream)
```

info

```
public void info(String comment,
               String category,
               String event)
```

(continued from last page)

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(String comment)
```

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)
```

fatal

```
public void fatal(String comment)
```

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(String comment)
```

debug

```
public void debug(String comment,  
    IMediaStream stream)
```

(continued from last page)

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)
```

warn

```
public void warn(String comment)
```

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: <code>_Server_</code>

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_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_HTTPSSMOOTH Value: https (smooth)
public static final	PROTO_HTTPSSTREAMER 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	STAT_connect_application_not_available Value: 302

public static final	<u>STAT_connect_application_not_found</u> Value: 404
public static final	<u>STAT_connect_bad_gateway</u> Value: 502
public static final	<u>STAT_connect_internal_error</u> Value: 500
public static final	<u>STAT_connect_license_limit</u> Value: 413
public static final	<u>STAT_connect_pending_wating</u> Value: 100
public static final	<u>STAT_connect_redirect</u> Value: 302
public static final	<u>STAT_connect_rejected_by_application</u> Value: 401
public static final	<u>STAT_connect_rejected_by_module</u> Value: 403
public static final	<u>STAT_connect_resource_limit</u> Value: 409
public static final	<u>STAT_connect_service_unavailable</u> Value: 503
public static final	<u>STAT_connect_successful</u> Value: 200
public static final	<u>STAT_connect_unknown_protocol</u> 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	STAT_play_rejected_by_application Value: 401
public static final	STAT_play_rejected_by_module Value: 403
public static final	STAT_play_stream_not_found Value: 404
public static final	STAT_play_successful Value: 200
public static final	STAT_play_unsupported_media_type Value: 415
public static final	STAT_publish_bad_request Value: 400
public static final	STAT_publish_in_use Value: 409
public static final	STAT_publish_internal_error Value: 500
public static final	STAT_publish_rejected_by_application Value: 401
public static final	STAT_publish_successful Value: 200
public static final	STAT_publish_unsupported_media_type Value: 415
public static final	STAT_stop_client_disconnect 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

(continued from last page)

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**

(continued from last page)

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**

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
```

(continued from last page)

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
```

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**

(continued from last page)

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**

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
```

(continued from last page)

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
```

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**

(continued from last page)

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**

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
```

(continued from last page)

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
```

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**

(continued from last page)

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**

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
```

(continued from last page)

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
```

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**

(continued from last page)

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**

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
```

(continued from last page)

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
```

Constant value: **rtmpte (HTTP-1.1)**

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_HTTPDVRCHUNK

```
public static final java.lang.String PROTO_HTTPDVRCHUNK
```

Constant value: **http (dvr)**

PROTO_HTTPSTREAMER

```
public static final java.lang.String PROTO_HTTPSTREAMER
```

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)**

(continued from last page)

PROTO_HTTPSDVRCHUNK

```
public static final java.lang.String PROTO_HTTPSDVRCHUNK
```

Constant value: **https (dvr)**

FD_ALL

```
public static final java.lang.String FD_ALL
```

CAT_ALL

```
public static final java.lang.String CAT_ALL
```

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 ID3Frames

java.lang.Object

└─com.wowza.wms.media.mp3.model.idtags.ID3Frames

public class **ID3Frames**
extends Object

Field Summary

public static final	ID3FOOTER_SIZE Value: 10
public static final	ID3HEADER_SIZE Value: 10
public static final	ID3HEADER_VERSION Value: 1024
public static final	ID3HEADERFLAGS_DEFAULT Value: 0
public static final	ID3HEADERFLAGS_EXPERIMENTAL Value: 32
public static final	ID3HEADERFLAGS_EXTENDED Value: 64
public static final	ID3HEADERFLAGS_FOOTERPRESENT Value: 16
public static final	ID3HEADERFLAGS_UNSYNC Value: 128

Constructor Summary

public	ID3Frames()
--------	-----------------------------

Method Summary

void	clear()
java.util.List	getFrameMapIds()

<code>java.util.List</code>	<code>getFrames()</code>
<code>Object</code>	<code>getLock()</code>
<code>int</code>	<code>getSize()</code>
<code>boolean</code>	<code>isEmpty()</code>
<code>void</code>	<code>putFrame</code> (<code>IID3V2Frame</code> frame)
<code>IID3V2Frame</code>	<code>removeFrame</code> (<code>IID3V2Frame</code> frame)
<code>byte[]</code>	<code>serialize()</code>
<code>byte[]</code>	<code>serialize</code> (boolean includeHeader, boolean includeFooter, int flags)
<code>static int</code>	<code>serializeFooter</code> (byte[] buffer, int offset, int flags, int size)
<code>static int</code>	<code>serializeHeader</code> (byte[] buffer, int offset, int flags, int size)
<code>int</code>	<code>serializeTags</code> (byte[] buffer, int offset)

Methods inherited from class `java.lang.Object`

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

Fields

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**

(continued from last page)

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**

ID3HEADERFLAGS_FOOTERPRESENT

```
public static final int ID3HEADERFLAGS_FOOTERPRESENT
```

Constant value: **16**

Constructors

ID3Frames

```
public ID3Frames()
```

Methods

getLock

```
public Object getLock()
```

clear

```
public void clear()
```

(continued from last page)

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()
```

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)
```

(continued from last page)

serialize

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

serialize

```
public byte[] serialize(boolean includeHeader,  
                        boolean includeFooter,  
                        int flags)
```

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](#)

[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_ISO_8859_1](#), [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	<u>setPictureData</u> (byte[] pictureData)
void	<u>setPictureDataAsURL</u> (String urlStr)
void	<u>setPictureType</u> (int pictureType)
void	<u>setTextEncoding</u> (int textEncoding)

Methods inherited from class [com.wowza.wms.media.mp3.model.idtags.ID3V2FrameBase](#)

[byteStringLen](#), [deserializeFrame](#), [deserializeString](#), [getFlags](#), [getIdStr](#), [getMapIdStr](#), [getSize](#), [serialize](#), [serializeString](#), [serializeStringLen](#), [setFlags](#), [setIdStr](#), [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:

[ID3V2FrameURLLink](#), [ID3V2FrameTextInformation](#), [ID3V2FrameSynchronizedText](#),
[ID3V2FrameRecommendedBufferSize](#), [ID3V2FrameRawBytes](#), [ID3V2FramePrivate](#), [ID3V2FrameLinkedInformation](#),
[ID3V2FrameComment](#), [ID3V2FrameAttachedPicture](#)

public abstract class **ID3V2FrameBase**

extends Object

implements [IID3V2Frame](#)

Field Summary

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	<u>TAG_TIT3</u> Value: TIT3
public static final	<u>TAG_TKEY</u> Value: TKEY
public static final	<u>TAG_TLAN</u> Value: TLAN
public static final	<u>TAG_TLEN</u> Value: TLEN
public static final	<u>TAG_TMCL</u> Value: TMCL
public static final	<u>TAG_TMED</u> Value: TMED
public static final	<u>TAG_TMOO</u> Value: TMOO
public static final	<u>TAG_TOAL</u> Value: TOAL
public static final	<u>TAG_TOFN</u> Value: TOFN
public static final	<u>TAG_TOLY</u> Value: TOLY
public static final	<u>TAG_TOPE</u> Value: TOPE
public static final	<u>TAG_TOWN</u> Value: TOWN
public static final	<u>TAG_TPE1</u> Value: TPE1
public static final	<u>TAG_TPE2</u> Value: TPE2
public static final	<u>TAG_TPE3</u> Value: TPE3
public static final	<u>TAG_TPE4</u> 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_ISO_8859_1 Value: 0
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)
static ID3V2FrameBase	deserializeFrame (byte[] buffer, int offset, int len)
static String	deserializeString (byte[] buffer, int offset, int len)
int	getFlags ()

String	getIdStr()
String	getMapIdStr()
int	getSize()
int	serialize (byte[] buffer, int offset)
static int	serializeString (String value, byte[] buffer, int offset)
static int	serializeStringLen (String value)
void	setFlags (int flags)
void	setIdStr (String idStr)
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**

(continued from last page)

TEXTENCODING_UTF16BE

```
public static final int TEXTENCODING_UTF16BE
```

Constant value: **2**

TEXTENCODING_UTF8

```
public static final int TEXTENCODING_UTF8
```

Constant value: **3**

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

byteStringLen

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

serializeStringLen

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

trimTrailingZero

```
public static String trimTrailingZero(String value)
```

serializeString

```
public static int serializeString(String value,  
    byte[] buffer,  
    int offset)
```

deserializeString

```
public static String deserializeString(byte[] buffer,  
    int offset,  
    int len)
```

deserializeFrame

```
public static ID3V2FrameBase deserializeFrame(byte[] buffer,  
    int offset,  
    int len)
```

serialize

```
public int serialize(byte[] buffer,  
    int offset)
```

getIdStr

```
public String getIdStr()
```

(continued from last page)

setIdStr

```
public void setIdStr(String idStr)
```

getFlags

```
public int getFlags()
```

setFlags

```
public void setFlags(int flags)
```

getSize

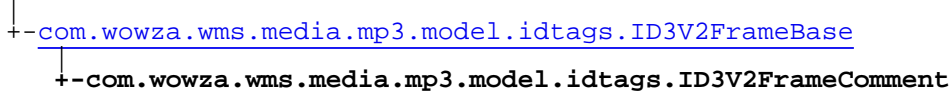
```
public int getSize()
```

getMapIdStr

```
public String getMapIdStr()
```

com.wowza.wms.media.mp3.model.idtags Class ID3V2FrameComment

java.lang.Object



All Implemented Interfaces:

[IID3V2Frame](#)

```

public class ID3V2FrameComment
extends ID3V2FrameBase
  
```

Fields inherited from class [com.wowza.wms.media.mp3.model.idtags.ID3V2FrameBase](#)

[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_ISO_8859_1](#), [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	getDescription ()
int	getTextEncoding ()
String	getValue ()
int	serializeBody (byte[] buffer, int offset)

void	<u>setDescription</u> (String description)
void	<u>setTextEncoding</u> (int textEncoding)
void	<u>setValue</u> (String value)

Methods inherited from class [com.wowza.wms.media.mp3.model.idtags.ID3V2FrameBase](#)

[byteStringLen](#), [deserializeFrame](#), [deserializeString](#), [getFlags](#), [getIdStr](#), [getMapIdStr](#), [getSize](#), [serialize](#), [serializeString](#), [serializeStringLen](#), [setFlags](#), [setIdStr](#), [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

ID3V2FrameComment

```
public ID3V2FrameComment(String idStr,
                          int flags)
```

ID3V2FrameComment

```
public ID3V2FrameComment(String idStr)
```

ID3V2FrameComment

```
public ID3V2FrameComment()
```

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()
```

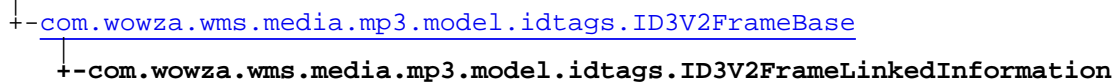
setDescription

```
public void setDescription(String description)
```

com.wowza.wms.media.mp3.model.idtags

Class 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](#)

[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_ISO_8859_1](#), [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](#), [deserializeFrame](#), [deserializeString](#), [getFlags](#), [getIdStr](#), [getMapIdStr](#), [getSize](#), [serialize](#), [serializeString](#), [serializeStringLen](#), [setFlags](#), [setIdStr](#), [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()
```

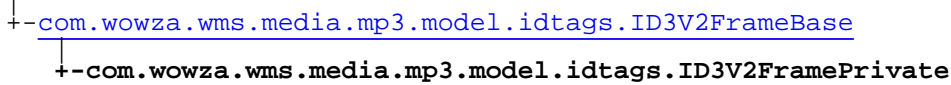
setData

```
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](#)

[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_ISO_8859_1](#), [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 ()
int	serializeBody (byte[] buffer, int offset)

Methods inherited from class [com.wowza.wms.media.mp3.model.idtags.ID3V2FrameBase](#)

[byteStringLen](#), [deserializeFrame](#), [deserializeString](#), [getFlags](#), [getIdStr](#), [getMapIdStr](#),
[getSize](#), [serialize](#), [serializeString](#), [serializeStringLen](#), [setFlags](#), [setIdStr](#),
[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)
```

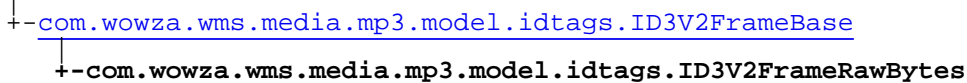
getBodySize

```
public int getBodySize()
```

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](#)

[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_ISO_8859_1](#), [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](#), [deserializeFrame](#), [deserializeString](#), [getFlags](#), [getIdStr](#), [getMapIdStr](#), [getSize](#), [serialize](#), [serializeString](#), [serializeStringLen](#), [setFlags](#), [setIdStr](#), [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](#)

[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_ISO_8859_1](#), [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](#), [deserializeFrame](#), [deserializeString](#), [getFlags](#), [getIdStr](#), [getMapIdStr](#), [getSize](#), [serialize](#), [serializeString](#), [serializeStringLen](#), [setFlags](#), [setIdStr](#), [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)
```

getBodySize

```
public int getBodySize()
```

(continued from last page)

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](#)

[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_ISO_8859_1](#), [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](#), [deserializeFrame](#), [deserializeString](#), [getFlags](#), [getIdStr](#), [getMapIdStr](#), [getSize](#), [serialize](#), [serializeString](#), [serializeStringLen](#), [setFlags](#), [setIdStr](#), [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**

(continued from last page)

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()
```

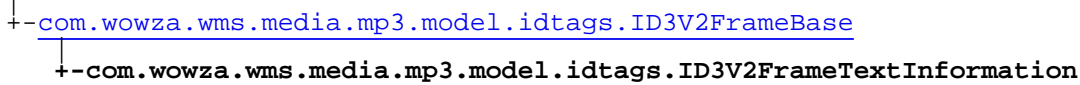
setValue

```
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](#)

Field Summary

protected	addTrailingZero
-----------	---------------------------------

Fields inherited from class [com.wowza.wms.media.mp3.model.idtags.ID3V2FrameBase](#)

[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_ISO_8859_1](#), [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](#), [deserializeFrame](#), [deserializeString](#), [getFlags](#), [getIdStr](#), [getMapIdStr](#), [getSize](#), [serialize](#), [serializeString](#), [serializeStringLen](#), [setFlags](#), [setIdStr](#), [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

addTrailingZero

protected boolean **addTrailingZero**

Constructors

ID3V2FrameTextInformation

```
public ID3V2FrameTextInformation(String idStr,  
                                int flags)
```

ID3V2FrameTextInformation

```
public ID3V2FrameTextInformation(String idStr)
```

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

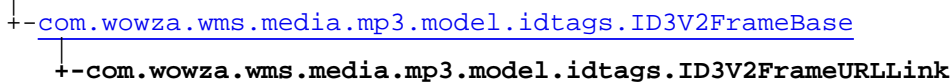
```
public int getTextEncoding()
```

setTextEncoding

```
public void setTextEncoding(int textEncoding)
```

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](#)

[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_ISO_8859_1](#), [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 ()
int	getTextEncoding ()
String	getURL ()
int	serializeBody (byte[] buffer, int offset)
void	setTextEncoding (int textEncoding)
void	setURL (String value)

Methods inherited from class [com.wowza.wms.media.mp3.model.idtags.ID3V2FrameBase](#)

[byteStringLen](#), [deserializeFrame](#), [deserializeString](#), [getFlags](#), [getIdStr](#), [getMapIdStr](#), [getSize](#), [serialize](#), [serializeString](#), [serializeStringLen](#), [setFlags](#), [setIdStr](#), [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()
```

getURL

```
public String getURL()
```


(continued from last page)

setURL

```
public void setURL(String value)
```

getTextEncoding

```
public int getTextEncoding()
```

setTextEncoding

```
public void setTextEncoding(int textEncoding)
```

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	deserializeBody (byte[] buffer, int offset, int len)
int	getBodySize ()
int	getFlags ()
String	getIdStr ()
String	getMapIdStr ()
int	getSize ()
int	serialize (byte[] buffer, int offset)
int	serializeBody (byte[] buffer, int offset)
void	setFlags (int flags)
void	setIdStr (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.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)
<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
Object	<u>getStreamIsRunningLock</u> () Get stream running lock
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
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.Session session)</u> sessionClosed callback
void	<u>sessionOpened(org.apache.mina.common.Session session)</u> sessionOpened callback
void	<u>setAppInstance(IApplicationInstance appInstance)</u> Set the application instance this media caster is associated 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>setStreamTimeout(int streamTimeout)</u> Set the watchdog stream timeout (milliseconds)
void	<u>shutdown(boolean isAppInstanceShutdown)</u> Shutdown media caster
void	<u>unregisterPlayer(IMediaStreamPlay player)</u> Unregister a player with this media caster

Fields

(continued from last page)

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**

STREAMTIMEOUTREASON_GOOD

```
public static final int STREAMTIMEOUTREASON_GOOD
```

Constant value: **100**

(continued from last page)

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,  
    IApplicationInstance 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

(continued from last page)

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)
```

Set the application instance this media caster is associated with

Parameters:

appInstance - application instance this media caster is associated with

(continued from last page)

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)
```

(continued from last page)

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)

Parameters:

streamTimeout - stream timeout

(continued from last page)

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

isStream

```
public boolean isStream()
```

(continued from last page)

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

getStreamIsRunningLock

```
public Object getStreamIsRunningLock( )
```

Get stream running lock

Returns:

stream running lock

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)
```

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

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. NOTE: This is not implemented yet (coming soon)

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. NOTE: This is not implemented yet (coming soon)

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	<u>MediaCasterItem</u> (String name, String streamType, String baseClass) Media caster item constructor
--------	--

Method Summary

void	<u>clearProperty</u> (String name) Clear property
String	<u>getBaseClass</u> () Get base class
int	<u>getConnectionTimeout</u> () Get connection timeout (milliseconds)
String	<u>getDescription</u> () Get description
static String	<u>getIdString</u> (String name, String liveStreamPacketizer, String liveStreamRepeater) Get id string for this media caster item (not used - returns name unchanged)
int	<u>getKeepAliveTime</u> () Get keep alive time (milliseconds)
String	<u>getName</u> () Get name
<u>WMSProperties</u>	<u>getProperties</u> () Get properties collection
String	<u>getProperty</u> (String name) Get property value
String	<u>getStreamType</u> () Get the stream type
String	<u>idStringToName</u> (String idString) Convert name to id string (not used - returns id string unchanged)
static MediaCasterStreamId	<u>parseIdString</u> (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	MediaCasterList() Constructor
--------	--

Method Summary

MediaCasterItem	getMediaCasterDef(String name) Get media caster definition by name
java.util.Map	getMediaCasterDefs() Get map of media caster items
java.util.List	getMediaCasterNames() 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

<u>IApplicationInstance</u>	<u>getApplicationInstance()</u> Get the parent application instance for this map
edu.emory.mathcs.backport.java.util.concurrent.locks.WMSReadWriteLock	<u>getLock()</u> Get the read/write lock for this interface
<u>MediaCasterStreamItem</u>	<u>getMediaCaster(String streamName)</u> Get media caster item based on given stream name
<u>MediaCasterStreamItem</u>	<u>getMediaCaster(String streamName, String liveStreamPacketizer, String liveStreamRepeater)</u> Get media caster item based on given stream name
int	<u>getMediaCasterCount()</u> Get the number of mediacasters current running
java.util.List	<u>getMediaCasterNames()</u> Get a list of all the currently running media caster names
<u>MediaCasterStreamManager</u>	<u>getStreamManager()</u> Get the stream manager interface for managing the starting and stopping of streams
void	<u>registerPlayer(IMediaStreamPlay player, MediaCasterItem mediaCasterDef)</u> Register a player to a media caster item (internal use)
void	<u>release(MediaCasterStreamItem mediaCasterStreamItem)</u> Decrement lock count on media caster item
void	<u>release(MediaCasterStreamItem mediaCasterStreamItem, boolean removeIfZero)</u>
void	<u>remove(MediaCasterStreamItem mediaCasterStreamItem)</u>
void	<u>shutdown(boolean isAppInstanceShutdown)</u> Shutdown this media caster and close all running media casters
String	<u>streamSrcToMediaCaster(long streamSrc)</u>
void	<u>unregisterPlayer(IMediaStreamPlay player, MediaCasterItem mediaCasterDef)</u> 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)

Create a new mediacaster map

(continued from last page)

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

doWatchdog

```
public void doWatchdog()
```

Do periodic idle time processing

shutdown

```
public void shutdown(boolean isAppInstanceShutdown)
```

Shutdown this media caster and close all running media casters

Parameters:

isAppInstanceShutdown - is this due to application shutdown

(continued from last page)

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

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.

(continued from last page)

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)
```

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:

(continued from last page)

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)
```

remove

```
public void remove(MediaCasterStreamItem mediaCasterStreamItem)
```

(continued from last page)

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)

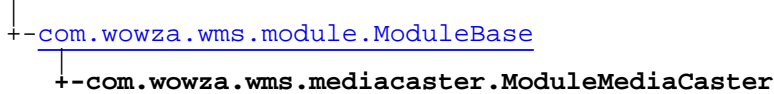
Parameters:

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 exists 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	onCodecConfigAAC (com.wowza.wms.media.aac.AACFrame frame, byte[] buffer, long offset)
void	onFrameAAC (com.wowza.wms.media.aac.AACFrame frame, byte[] buffer, long offset)
void	onFrameMP3 (int frequency, int samplesPerFrame, int channels, byte[] syncHeader, byte[] packetHeader, byte[] frameData)
void	onHeaderData (java.util.Map headerMap)
void	onMetaData (java.util.Map metaMap)
void	onTrim ()

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

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)

MediaListSegment	removeSegment (int index)
void	removeSegment (MediaListSegment mediaListSegment)
void	reset ()
void	setName (String name)
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**

segments

protected `java.util.List` **segments**

properties

protected `com.wowza.wms.application.WMSProperties` **properties**

lock

protected `java.lang.Object` **lock**

Constructors

MediaList

public **MediaList**()

Methods

(continued from last page)

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()
```

(continued from last page)

getFirstSegment

```
public MediaListSegment getFirstSegment()
```

getSegment

```
public MediaListSegment getSegment(int index)
```

getLock

```
public Object getLock()
```

getProperties

```
public WMSPProperties getProperties()
```

getProperties

```
public WMSPProperties getProperties(boolean write)
```

getName

```
public String getName()
```

setName

```
public void setName(String name)
```

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	height
protected	lock
protected	mediaListSegment
protected	name
protected	playDuration
protected	playStart
protected	properties
protected	type
protected	videoCodecId
protected	width
protected	wowzaAudioOnly

Constructor Summary

public	MediaListRendition()
--------	--------------------------------------

Method Summary

String	<u>getAudioCodecId()</u>
int	<u>getBitrateAudio()</u>
int	<u>getBitrateTotal()</u>
int	<u>getBitrateVideo()</u>
String	<u>getCodecId()</u>
int	<u>getHeight()</u>
Object	<u>getLock()</u>
<u>MediaListSegment</u>	<u>getMediaListSegment()</u>
String	<u>getName()</u>
long	<u>getPlayDuration()</u>
long	<u>getPlayStart()</u>
<u>WMSProperties</u>	<u>getProperties()</u>
<u>WMSProperties</u>	<u>getProperties</u> (boolean write)
int[]	<u>getSize()</u>
int	<u>getType()</u>
String	<u>getTypeAsString()</u>
String	<u>getVideoCodecId()</u>
int	<u>getWidth()</u>
boolean	<u>isWowzaAudioOnly()</u>
void	<u>setAudioCodecId</u> (String audioCodecId)
void	<u>setBitrateAudio</u> (int bitrateAudio)
void	<u>setBitrateTotal</u> (int bitrateTotal)
void	<u>setBitrateVideo</u> (int bitrateVideo)

void	setHeight (int height)
void	setLock (Object lock)
void	setMediaListSegment (MediaListSegment mediaListSegment)
void	setName (String name)
void	setPlayDuration (long playDuration)
void	setPlayStart (long playStart)
void	setSize (int width, int height)
void	setType (int type)
void	setVideoCodecId (String videoCodecId)
void	setWidth (int width)
void	setWowzaAudioOnly (boolean wowzaAudioOnly)
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**

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**

videoCodecId

protected java.lang.String **videoCodecId**

audioCodecId

protected java.lang.String **audioCodecId**

wowzaAudioOnly

protected boolean **wowzaAudioOnly**

width

protected int **width**

height

protected int **height**

(continued from last page)

playStart

protected long **playStart**

playDuration

protected long **playDuration**

Constructors

MediaListRendition

public **MediaListRendition**()

Methods

toString

public String **toString**()

toSMILString

public String **toSMILString**()

getTypeAsString

public String **getTypeAsString**()

getLock

public Object **getLock**()

setLock

public void **setLock**(Object lock)

getMediaListSegment

public [MediaListSegment](#) **getMediaListSegment**()

(continued from last page)

setMediaListSegment

```
public void setMediaListSegment(MediaListSegment mediaListSegment)
```

getProperties

```
public WMSProperties getProperties()
```

getProperties

```
public WMSProperties getProperties(boolean write)
```

getBitrateTotal

```
public int getBitrateTotal()
```

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)
```

(continued from last page)

getVideoCodecId

```
public String getVideoCodecId()
```

setVideoCodecId

```
public void setVideoCodecId(String videoCodecId)
```

getCodecId

```
public String getCodecId()
```

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()
```

(continued from last page)

setType

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

getWidth

```
public int getWidth()
```

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)
```

(continued from last page)

getPlayDuration

```
public long getPlayDuration()
```

setPlayDuration

```
public void setPlayDuration(long playDuration)
```

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 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 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_PLAYLIST</u> Value: 2

Method Summary

void	<u>onHTTPCupertinoEncryptionKeyCreateLive</u> (<u>IApplicationInstance</u> appInstance, String streamName, byte[] encKey) Called when live stream key is requested (per-published stream)
void	<u>onHTTPCupertinoEncryptionKeyCreateVOD</u> (<u>HTTPStreamerSessionCupertino</u> httpSession, byte[] 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, byte[] encKeyData) Called when a key data is requested.
void	<u>onHTTPCupertinoEncryptionKeyLiveChunk</u> (<u>ILiveStreamPacketizer</u> liveStreamPacketizer, String 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	<u>onHTTPCupertinoEncryptionKeyVODChunk</u> (<u>HTTPStreamerSessionCupertino</u> httpSession, <u>IHTTPStreamerCupertinoIndex</u> index, <u>CupertinoEncInfo</u> encInfo, long chunkId, int mode) Called when video on demand key is requested (per-session).

Fields

(continued from last page)

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**

Methods

onHTTPCupertinoEncryptionKeyRequest

```
public void onHTTPCupertinoEncryptionKeyRequest(HTTPStreamerSessionCupertino
httpSession,
IHTTPRequest req,
IHTTPResponse resp)
```

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)
```

(continued from last page)

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)

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	onHTTPCupertinoStreamingSessionCreate (HTTPStreamerSessionCupertino httpCupertinoStreamingSession) Invoked when an HTTP Cupertino Streaming session is created.
void	onHTTPCupertinoStreamingSessionDestroy (HTTPStreamerSessionCupertino httpCupertinoStreamingSession) 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 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 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

[onResult](#)([IClient](#) client, long pingTime, int pingId, boolean result)

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

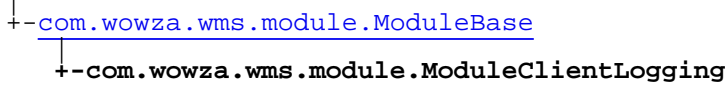
Parameters:

(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)
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)
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);

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

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:

(continued from last page)

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:

(continued from last page)

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:

(continued from last page)

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);

Parameters:

client - client
function - function
params - params (no params)

(continued from last page)

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)

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);`

(continued from last page)

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)
```

(continued from last page)

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)
```

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)
```

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

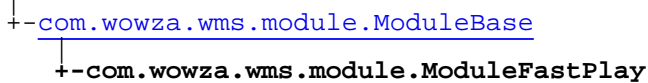
params - params (streamName:String)

FCUnSubscribe

```
public static void FCUnSubscribe(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

com.wowza.wms.module Class ModuleFastPlay

java.lang.Object



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>streamMonitorVideoTCPostTolerance</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>streamMonitorAudioTCPostTolerance</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>streamMonitorDataTCPostTolerance</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

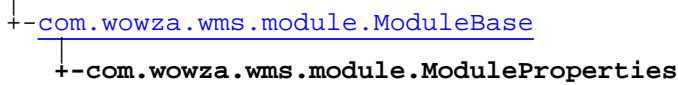
```
public void onValidateMediaCastersStop(IApplicationInstance appInstance)
```

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.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

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	onTeardown (RTPSession rtspSession, com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.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

(continued from last page)

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	getRTSPTunnelingSession (String sessionId) Get RTSP/RTP tunneling session by session id
RTPSessions	getSessions () Get RTP sessions
RTPPacketizerItem	getStreamPacketizerItem (IApplicationInstance appInstance, int codecId) Get stream packetizer for a given codec id.
com.wowza.wms.rtp.transport.UDPTransportManager	getUDPTransportManager () Get the UDP transport manager
IVHost	getVHost () Get vhost
RTPPacketizerItem	getVideoPacketizerItem (IApplicationInstance appInstance, int codecId) Get video packetizer for a given codec id.
void	init () Initialize
void	putAudioPacketizerItem (int codecId, RTPPacketizerItem rtpPacketizerInfo) Set the audio packetizer for a given codec id
void	putRTSPTunnelingSession (String sessionId, com.wowza.wms.rtsp.RTSP TunnelingSession rtspTunnelingSession) Remove RTSP/RTP tunneling session by session id
void	putStreamPacketizerItem (int codecId, RTPPacketizerItem rtpPacketizerInfo) Set the stream packetizer for a given codec id
void	putVideoPacketizerItem (int codecId, RTPPacketizerItem rtpPacketizerInfo) Set the video packetizer for a given codec id
boolean	releaseSocketAddress (java.net.SocketAddress socketAddress) Release UDP socket address
void	releaseUDPPortPair (int port) Release port pair
com.wowza.wms.rtsp.RTSP TunnelingSession	removeRTSPTunnelingSession (String sessionId) Remove RTSP/RTP tunneling session by session id
void	shutdown () Shutdown
void	shutdownRTPSession (RTPSession rtpSession) Gracefully and forcefully shutdown and RTP session
RTPStream	uncacheRTPStream (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	ioPerformanceCounter
protected	ioSession
protected	ip
protected	isAnnounce
protected	isConnected
protected	isDescribe
protected	isSessionValid
protected	lastAuthenticateMethod
protected	lock
protected	loggedConnect
protected	properties
protected	queryStr
protected	redirectSession
protected	redirectSessionCode
protected	redirectSessionMessage
protected	redirectSessionURL
protected	referrer
protected	rtpWriteListener
protected	rtspPlayRangeStart
protected	rtspPlayRangeStop
protected	rtspTunnelingSessionId
protected	serverIp
protected	serverPort
protected	sessionId

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	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	SDPLOCATION_STREAM Value: 0
public static final	SDPLOCATION_VIDEO Value: 2
public static final	STREAMINFO_SESSIONATTRIBUTES Value: attributes

public static final	STREAMINFO_SESSIONBANDWIDTH Value: bandwidth
public static final	STREAMINFO_SESSIONCONNECTIONDATA Value: connectiondata
public static final	STREAMINFO_SESSIONEMAILADDRESS Value: emailaddress
public static final	STREAMINFO_SESSIONINFORMATION Value: information
public static final	STREAMINFO_SESSIONNAME Value: name
public static final	STREAMINFO_SESSIONPHONENUMBER Value: phonenummer
public static final	STREAMINFO_SESSIONPORIGIN Value: origin
public static final	STREAMINFO_SESSIONPROTOCOLVERSION Value: protocolversion
public static final	STREAMINFO_SESSIONREPEATTIMES Value: repeattimes
public static final	STREAMINFO_SESSIONTIMEZONES Value: timezones
public static final	STREAMINFO_SESSIONTIMING Value: timing
public static final	STREAMINFO_SESSIONURI Value: uri

Constructor Summary

public	RTPStream (RTPContext context, String streamId) Constructor
--------	---

Method Summary

void	addStreamAttribute (String key, String value) Add a name value pair to the stream attribute collection
void	addStreamInfo (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) Create IMediaStream, Internal use.
String	<u>describe</u> (RTPSession rtspSession, int isStreamPacketizer, RTPRequestStatus status) Execute describe command
String	<u>describe</u> (RTPSession 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()</u> (double startTime, int videoSeq, int audioSeq) Get the RTP info

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()</u> (String trackId) Get track by id
String	<u>getTrackId()</u> (String seq) 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()</u> (boolean isMulticast) 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.
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>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>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</u> (String appInstanceName) Set the application instance name
void	<u>setAppName</u> (String appName) Get the application name
void	<u>setAVSyncMethod</u> (int avSyncMethod) Set the audio/video sync method.
void	<u>setBlockUDPOut</u> (boolean blockUDPOut)
void	<u>setCheckIpAddr</u> (boolean checkIpAddr) Are we checking the ip address of each incoming RTP packet
void	<u>setCheckSSRC</u> (boolean checkSSRC) Are we checking the ssrc values of each incoming RTP packet
void	<u>setForceMPEGTSOut</u> (boolean isForceMPEGTSOut)

void	<u>setForceRTSPInterleaved</u> (boolean isForceRTSPInterleaved) True if forcing RTSP interleaved
void	<u>setHost</u> (String host) Set host
void	<u>setLive</u> (boolean isLive) Is live stream
void	<u>setMaxRTCPWaitTime</u> (int maxRTCPWaitTime) Set max time to wait for RTCP sender reports (milliseconds)
void	<u>setMode</u> (int mode) Set the play/publish mode
void	<u>setMPEGTSAudioBitrate</u> (int mpegtsAudioBitrate)
void	<u>setMPEGTSAudioLanguage</u> (String mpegtsAudioLanguage)
void	<u>setMPEGTSAudioPID</u> (int mpegtsAudioPID) Set the audio PID id if MPEG-TS stream
void	<u>setMPEGTSOut</u> (boolean isMPEGTSOut) Is MPEG-TS out
void	<u>setMPEGTSProgramID</u> (int mpegtsProgramId)
void	<u>setMPEGTSVideoBitrate</u> (int mpegtsVideoBitrate)
void	<u>setMPEGTSVideoPID</u> (int mpegtsVideoPID) Set the video PID id if MPEG-TS stream
void	<u>setOutHost</u> (String outHost) Set out host
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)
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	<u>touch</u> () Touch the stream so that it does not timeout
String	<u>transportFindBestMatch</u> (String transport) Based on a trasport string from SETUP command find best match
void	<u>unlockRepeaterStreams</u> () 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**

(continued from last page)

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
```

(continued from last page)

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**

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
```

Constant value: **attributes**

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

(continued from last page)

Returns:

synchronization lock for this stream

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:

(continued from last page)

bytes

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

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)
```

Create IMediaStream, Internal use.

Parameters:

(continued from last page)

status - status

Returns:

true if successful

shutdown

```
public void shutdown(RTPRequestStatus status)
```

shutdown RTP stream, Internal use.

Parameters:

status - status

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()
```

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

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

(continued from last page)

attachToWMSSession

```
public void attachToWMSSession(RtmpSessionInfo wmsSessionInfo)
```

Attach to WMS session, Internal use

Parameters:

wmsSessionInfo - session info

detachFromWMSSession

```
public void detachFromWMSSession(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:

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()
```

(continued from last page)

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:

streamType - stream type

getAppName

```
public String getAppName()
```

Get the application name

Returns:

application name

setAppName

```
public void setAppName(String appName)
```

Get the application name

(continued from last page)

Parameters:

appName - application name

getAppName

```
public String getAppName()
```

Get the application instance name

Returns:

application instance name

setAppName

```
public void setAppName(String appName)
```

Set the application instance name

Parameters:

appName - 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

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()
```

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()
```

(continued from last page)

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

(continued from last page)

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)
```

(continued from last page)

Set the video PID id if MPEG-TS stream

Parameters:

mpegtVideoPID - 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

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

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)
```

(continued from last page)

Set out host

Parameters:

outHost - out host

idle

```
public void idle(org.apache.mina.common.IoSession 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

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

(continued from last page)

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

(continued from last page)

getNormalizedNTPTimecode

```
public long getNormalizedNTPTimecode(long timecode)
```

Turn a millisecond timcode 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

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

(continued from last page)

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

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

(continued from last page)

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()
```

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()
```

(continued from last page)

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()
```

(continued from last page)

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

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

(continued from last page)

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()
```

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()
```

(continued from last page)

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)
```

Set RTP destination

Parameters:

rtpDestination - RTP destination

getSDPLang

```
public String getSDPLang()
```

Get the SDP language

Returns:

SDP language

setSDPLang

```
public void setSDPLang(String sdpLang)
```

(continued from last page)

Set the SDP language

Parameters:

sdpLang - SDP language

isRTPIgnoreProfileLevelId

```
public boolean isRTPIgnoreProfileLevelId()
```

setRTPIgnoreProfileLevelId

```
public void setRTPIgnoreProfileLevelId(boolean rtpIgnoreProfileLevelId)
```

getUDPManagedDeliveryDelay

```
public int getUDPManagedDeliveryDelay()
```

setUDPManagedDeliveryDelay

```
public void setUDPManagedDeliveryDelay(int udpManagedDeliveryDelay)
```

getUDPManagedDeliveryCount

```
public int getUDPManagedDeliveryCount()
```

setUDPManagedDeliveryCount

```
public void setUDPManagedDeliveryCount(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)
```

Package

com.wowza.wms.server

com.wowza.wms.server Interface ICommandInterfaceCommand

public interface **ICommandInterfaceCommand**
extends

Method Summary

boolean	<u>canHandle</u> (CommandInterfaceRequestMessage req)
void	<u>invoke</u> (CommandInterfaceRequestMessage req, CommandInterfaceResponseMessage resp)

Methods

invoke

```
public void invoke(CommandInterfaceRequestMessage req,  
    CommandInterfaceResponseMessage resp)
```

canHandle

```
public boolean canHandle(CommandInterfaceRequestMessage req)
```

com.wowza.wms.server Interface IResponseListener

public interface **IResponseListener**
extends

IResponseListener: Internal use.

Method Summary

void	<u>onResponseWriteStart</u> (RtmpResponseMessage response)
void	<u>onResponseWriteStop</u> (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.

Method Summary

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_*)
int	getCoreHandlerPoolSize () Get the handler core thread pool size.
int	getCoreTransportPoolSize () Get the transport core thread pool size.
String	getDateStarted () Get the date and time the server was started.
java.util.Properties	getDynamicLogProperties () Get the dynamic log properties defined at the server level in conf/log4j.properties
ThreadPool	getHandlerThreadPool () Get the server handler thread pool.
IOPerformanceCounter	getIoPerformanceCounter () Get the server performance counter.
IOPerformanceCounter	getIoPerformanceCounter (int counterIndex) Get the server performance counter for a specific technology (see IVHost.COUNTER_*)

WMSProperties	getProperties() Get server level properties collection
ThreadPool	getThreadPool() Get the server handler thread pool.
String	getTimeRunning() Get a formatted String of how long the server has been running.
double	getTimeRunningSeconds() Get time running in seconds
ThreadPool	getTransportThreadPool() Get the server transport thread pool.
com.wowza.wms.transpo rt.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
com.wowza.wms.transpo rt.udp.UDPPortSharing Manager	getUDPPortSharingManager() Get the UDP port sharing manager.
String[]	getUserAgents() Get a pipe " " delimited list of user agents that the server recognizes as RTMPT client.
String	getVersion() Get server version number.
VHostList	getVHostList() Returns the interface to the VHostList for the server
boolean	isDynamicLogContextLoaded(String logContext) Returns true if the given dynamic log context is already loaded.
boolean	isSuspended() Is the server current suspended
String	readConfig(String sName) Method to read xml config file..
void	reloadVHostConfig() Reload the VHosts.xml file.
void	removeServerListener(IServerNotify serverListener) Remove server listener
void	setCommandInterfaceHostPort(HostPort commandInterfaceHostPort) Set the definition for the command interface.
void	setCoreHandlerPoolSize(int corePoolSize) Set the handler core thread pool size.
void	setCoreTransportPoolSize(int corePoolSize) Set the transport core thread pool size.
void	setDynamicLogProperties(java.util.Properties dynamicLogProperties) Set the dynamic log properties set at the server level

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	<u>unbindAllVHosts</u> () Unbind all virtual hosts (Calls IVHost.unbindAllHostPorts for each vhost)
boolean	<u>writeConfig</u> (String sName, String data) Method to write xml config file..

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.

(continued from last page)

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_*)

(continued from last page)

Returns:io performance counter

getConnectionCounter

```
public ConnectionCounter getConnectionCounter()
```

Get the server connection counter.

Returns:connection counter

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

(continued from last page)

Returns:

host port definition of command interface

setCommandInterfaceHostPort

```
public void setCommandInterfaceHostPort(HostPort commandInterfaceHostPort)
```

Set the definition for the command interface.

Parameters:

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

(continued from last page)

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

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()
```

(continued from last page)

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()
```

Get server level properties collection

Returns:

server level properties 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

com.wowza.wms.server Interface IServerNotify

All Subinterfaces:

[IServerNotify2](#)

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](#)

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 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 static	logNotifier
---------------	-----------------------------

Constructor Summary

public	Server()
--------	--------------------------

Method Summary

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)
IConnectionValidator	getConnectionValidator()
int	getCoreHandlerPoolSize()
int	getCoreTransportPoolSize()
int	getCryptoPoolActiveCount()
int	getCryptoPoolMaxSize()
long	getCurrentHeapSize()
String	getDateStarted()
java.util.Properties	getDynamicLogProperties()
String	getGUID()
ThreadPool	getHandlerThreadPool()
static Server	getInstance()
IOPerformanceCounter	getIoPerformanceCounter()
IOPerformanceCounter	getIoPerformanceCounter(int counterIndex)
JMXRemoteConfig	getJmxRemoteConfig()

LicenseCounter	getLicenseCounter (int index)
Server.LicenseSessionTracker	getLicenseTracker (int index)
ILicenseValidator	getLicenseValidator ()
long	getLiveThreads ()
long	getMaxHeapSize ()
long	getPeakThreads ()
WMSProperties	getProperties ()
boolean[]	getProtocolUsage ()
Server.ProtocolUsageSessionTracker	getProtocolUsageTracker ()
String	getServerGUID ()
String	getSessionGUID ()
ThreadPool	getThreadPool ()
String	getTimeRunning ()
double	getTimeRunningSeconds ()
ThreadPool	getTransportThreadPool ()
com.wowza.wms.transport.udp.UDPPortManager	getUDPPortManager ()
com.wowza.wms.transport.udp.UDPPortSharingManager	getUDPPortSharingManager ()
String[]	getUserAgents ()
String	getVersion ()
VHostList	getVHostList ()
boolean	isDynamicLogContextLoaded (String logContext)
boolean	isSuspended ()
boolean	isVHostRunning (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>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>setDynamicLogProperties</u> (java.util.Properties dynamicLogProperties)
void	<u>setIoPerformanceCounter</u> (<u>IOPerformanceCounter</u> ioPerformanceCounter)
void	<u>setUserAgents</u> (String[] userAgents)
static void	<u>start</u> ()
void	<u>startCommandInterface</u> ()
void	<u>startServer</u> ()
void	<u>startVHost</u> (String vhostName)
void	<u>startVHosts</u> ()
void	<u>stopAdminAgent</u> ()
void	<u>stopCommandInterface</u> ()
void	<u>stopServer</u> ()
void	<u>stopVHost</u> (String vhostName)
void	<u>stopVHosts</u> ()

void	<u>suspendAllVHosts()</u>
void	<u>suspendCommandInterface()</u>
void	<u>suspendServer()</u>
void	<u>unbindAllVHosts()</u>
void	<u>unregisterLiveStreamTranscoder(ILiveStreamTranscoder liveStreamTranscoder, byte[] license)</u>
<u>IOPerformanceCounter</u>	<u>updateIOPerformance()</u>
void	<u>updateLoggingDuration()</u>
boolean	<u>writeConfig</u> (String sName, String data)
static boolean	<u>writeXMLConfig</u> (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](#)

[addServerListener](#), [getAdminInterfaceObjectList](#), [getClientIdGenerator](#), [getCommandInterfaceCommandHandler](#), [getCommandInterfaceHostPort](#), [getConnectionCounter](#), [getConnectionCounter](#), [getCoreHandlerPoolSize](#), [getCoreTransportPoolSize](#), [getDateStarted](#), [getDynamicLogProperties](#), [getHandlerThreadPool](#), [getIoPerformanceCounter](#), [getIoPerformanceCounter](#), [getProperties](#), [getThreadPool](#), [getTimeRunning](#), [getTimeRunningSeconds](#), [getTransportThreadPool](#), [getUDPPortManager](#), [getUDPPortSharingManager](#), [getUserAgents](#), [getVersion](#), [getVHostList](#), [isDynamicLogContextLoaded](#), [isSuspended](#), [readConfig](#), [reloadVHostConfig](#), [removeServerListener](#), [setCommandInterfaceHostPort](#), [setCoreHandlerPoolSize](#), [setCoreTransportPoolSize](#), [setDynamicLogProperties](#), [setUserAgents](#), [startCommandInterface](#), [startVHost](#), [startVHosts](#), [stopAdminAgent](#), [stopCommandInterface](#), [stopVHost](#), [stopVHosts](#), [suspendAllVHosts](#), [suspendServer](#), [unbindAllVHosts](#), [writeConfig](#)

Fields

logNotifier

```
public static com.wowza.wms.logging.ILogNotify logNotifier
```

Constructors

(continued from last page)

Server

```
public Server()
```

Methods

getInstance

```
public static Server getInstance()
```

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()
```

(continued from last page)

startCommandInterface

```
public void startCommandInterface()
```

stopCommandInterface

```
public void stopCommandInterface()
```

startServer

```
public void startServer()
```

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()
```

(continued from last page)

suspendServer

```
public void suspendServer()
```

suspendAllVHosts

```
public void suspendAllVHosts()
```

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)
```

(continued from last page)

getConnectionCounter

```
public ConnectionCounter getConnectionCounter()
```

getConnectionCounter

```
public ConnectionCounterSimple getConnectionCounter(int counterIndex)
```

getConnectionValidator

```
public IConnectionValidator getConnectionValidator()
```

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)
```

(continued from last page)

getCommandInterface

```
public Object getCommandInterface()
```

setCommandInterface

```
public void setCommandInterface(Object commandInterface)
```

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()
```

(continued from last page)

getCoreTransportPoolSize

```
public int getCoreTransportPoolSize()
```

setCoreTransportPoolSize

```
public void setCoreTransportPoolSize(int corePoolSize)
```

getCoreHandlerPoolSize

```
public int getCoreHandlerPoolSize()
```

setCoreHandlerPoolSize

```
public void setCoreHandlerPoolSize(int corePoolSize)
```

getThreadPool

```
public ThreadPool getThreadPool()
```

getTransportThreadPool

```
public ThreadPool getTransportThreadPool()
```

getHandlerThreadPool

```
public ThreadPool getHandlerThreadPool()
```

getProperties

```
public WMSProperties getProperties()
```

getClientIdGenerator

```
public RandomIdGenerator getClientIdGenerator()
```

getDynamicLogProperties

```
public java.util.Properties getDynamicLogProperties()
```

(continued from last page)

setDynamicLogProperties

```
public void setDynamicLogProperties(java.util.Properties dynamicLogProperties)
```

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()
```

(continued from last page)

getCryptoPoolMaxSize

```
public int getCryptoPoolMaxSize()
```

getCryptoPoolActiveCount

```
public int getCryptoPoolActiveCount()
```

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)
```

(continued from last page)

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.

(continued from last page)

fs

```
public static String fs(byte[] kiIn,  
    String p)
```

Deprecated.

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()
```

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	SHARED_OBJECT_STATUS_DELETE shared object status: delete Value: 9
public static final	SHARED_OBJECT_STATUS_SUCCESS shared object status: success Value: 5

Method Summary

void	acquire() Increment the reference count to this shared object.
void	addClient(IClient client) Add a client to this shared object.
void	addSlotListener(ISharedObjectSlotNotify slotListener) Add a slot listener.
void	clear() Clear all properties from a shared object
void	close() Force close this shared object (not implemented)
boolean	containsProperty(String slotName) Returns true is slot/property name exists
boolean	containsSlot(String slotName) Returns true is slot/property name exists
void	deleteSlot(IClient client, String slotName) Remove a slot (property)
void	deleteSlot(String slotName) Remove slot (property)
void	disconnect(IClient client) Disconnect client from shared object.
void	flush() Flush (write to disk) shared object
java.util.List	getClients() Get a list of client that are connected to this shared object.
String	getName() Get shared object name
ISharedObjects	getParent() Get the shared object container to which this shared object belongs.
AMFData	getProperty(String slotName) Get slot (property) value.
int	getRefCount() 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()
```

Decrement 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 exists 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 is 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 appInstance, IMediaStream stream, String mediaReadType, String basePath, String mediaName, IHTTPStreamerSession httpStreamerSession)</code> Initialize mediaReader
boolean	<code>isOpen()</code> is file open
void	<code>open(String basePath, String name)</code> Open the file for reading
void	<code>setHTTPStreamerSession(IHTTPStreamerSession httpStreamerSession)</code> Set the HTTPStreamerSession associated with this media list reader
void	<code>setMediaReaderItem(MediaReaderItem mediaReaderItem)</code> Set the mediaReader item definition
void	<code>setProperties(WMSProperties properties)</code> 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>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**

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**

SEEK_KEYUP

```
public static final int SEEK_KEYUP
```

Seek direction: up to closets key frame
Constant value: **1**

(continued from last page)

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)
```

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

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

(continued from last page)

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

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

(continued from last page)

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:

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

setProperty

```
public void setProperty(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 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	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	addVideoCodecConfigPacket (long timecode, AMFPacket packet) Set video codec configuration packet (needed for H.264/AAC playback)
void	addVideoData (byte[] data, int offset, int size) Add data to 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	getDataType() Get the data packet type: (IVHost.CONTENTTYPE_DATA0 or IVHost.CONTENTTYPE_DATA3)
String	getDvrRecorder() Get the DVR Recorder for this stream
ILiveStreamDvrRecorder	getDvrRecorder(String name) Get the DVR Recorder interface to a stream by name
String	getDvrRecorderList() Get the comma separated list of DVR Recorder names being used by this stream (see conf/Dvr.xml)
String	getDvrRepeater() Get the DVR repeater name for this stream
ElapsedTimer	getElapsedTime() Get the interface to the elapse timer
String	getExt() Get media file extension
FastPlaySettings	getFastPlaySettings() Get current fastPlay settings
int	getHeaderSize() Get the last packet header size (debugging)
IHTTPStreamerSession	getHTTPStreamerSession() Get the HTTPStreamer session associated with this stream
AMFPacket	getLastKeyFrame() Get most recent video key frame
AMFPacket	getLastPacket() Get most recent live packet
String	getLiveStreamPacketizer() Get the live stream packetizer that this stream is using
ILiveStreamPacketizer	getLiveStreamPacketizer(String name) Get the LiveStreamPacketizer interface to a stream by name
String	getLiveStreamPacketizerList() Get the comma separated list of LiveStreamPacketizers names being used by this stream (see conf/LiveStreamPacketizers.xml)
String	getLiveStreamRepeater() Get the live stream repeater name for the stream
ILiveStreamTranscoder	getLiveStreamTranscoder(String name) Get a live stream transcoder for this stream by name
String	getLiveStreamTranscoderList() Get the comma separated list of LiveStreamTranscoders names being used by this stream (see conf/LiveStreamTranscoders.xml)
java.util.Map	getLiveStreamTranscoders() 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
<u>AMFObj</u>	<u>getRespAMFVideoObj()</u> Get video response channel object
<u>RTPStream</u>	<u>getRTPStream()</u> Get the RTP based stream this stream is associated with
int	<u>getSrc()</u> Get the stream id
java.io.File	<u>getStreamFileForRead()</u> Get the File object to read from a stream (get stream name, ext and query from stream object)
java.io.File	<u>getStreamFileForRead()</u> (String name, String ext, String query) Get the File object to read from a stream (specify name, ext and query)
java.io.File	<u>getStreamFileForWrite()</u> Get the File object to write to a stream (get stream name, ext and query from stream object)

java.io.File	<u>getStreamFileForWrite</u> (String name, String ext, String query) Get the File object to write to a stream (specify name, ext and query)
<u>MediaStreamMap</u>	<u>getStreams</u> () Get parent mediaStreamMap (owned by applicationInstance)
String	<u>getStreamType</u> () Get mediaStream streamType
String	<u>getUniqueStreamIdStr</u> () Get a string that uniquely identifies this stream
<u>AMFPacket</u>	<u>getVideoCodecConfigPacket</u> (long timecode) Get video codec configuration packet (needed for H.264/AAC playback)
int	<u>getVideoMissing</u> () Get number of video bytes missing from current audio packet
int	<u>getVideoSize</u> () Get the size of the current video packet that is being streamed from the client to the server
long	<u>getVideoTC</u> () Get last absolute video timecode (milliseconds) sent to mediaStream
void	<u>handleCallback</u> (com.wowza.wms.request.RequestFunction function) Routes request function to callback handler onStatus, onPlayStatus or [method/handler]
boolean	<u>idle</u> () Allow the mediaStream to perform idle work (not currently in use)
long	<u>incrementMediaInBytes</u> (long increment) Increment the number of mediaStream bytes received
long	<u>incrementMediaLossBytes</u> (long bytes, long count) Increment the number of mediaStream loss bytes sent and number of packets sent
long	<u>incrementMediaOutBytes</u> (long bytes, long count) Increment the number of mediaStream bytes sent and number of packets sent
void	<u>init</u> (<u>MediaStreamMap</u> parent, int src, <u>WMSProperties</u> properties) Initialize the mediaStream object after creation.
void	<u>initLiveStreamRepeating</u> (String liveStreamPacketizer, String liveStreamRepeater) Initialize this stream for live stream repeating
boolean	<u>isAppend</u> () Is append to media file (only valid if isRecord)
boolean	<u>isClustered</u> () not used
boolean	<u>isMediaCasterPlay</u> () Is MediaCaster play enabled (if true, will trigger MediaCaster startup)
boolean	<u>isMergeOnMetadata</u> () If true, merge incoming onMetadata events with the current onMetadata event data.
boolean	<u>isOpen</u> () Is mediaStream open

boolean	<u>isPlay()</u> Is the stream a play stream (vs a publish stream)
boolean	<u>isPlaying()</u> Is mediaStream playing (or paused - false)
boolean	<u>isPublishStreamReady()</u> (boolean checkAudio, boolean checkVideo) Returns true if the publishing stream contains enough video/audio data to start playback
boolean	<u>isReceiveAudio()</u> Is client currently receiving audio.
boolean	<u>isReceiveVideo()</u> Is client currently receiving video.
boolean	<u>isRecord()</u> Is this stream being recorded to a file
boolean	<u>isSendPlayStopLogEvent()</u> Get need to send a log event for stop
boolean	<u>isSendPublishStopLogEvent()</u> Get need to send a log event for publishing
boolean	<u>isSendRecordStopLogEvent()</u> Get need to send a log event for recording
boolean	<u>isTranscodeResult()</u> Is this stream the result of a transcode operation.
boolean	<u>isVideoH264SEIListenerEmpty()</u> Is H.264 SEI listener list empty.
double	<u>length()</u> 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>notifyVideoH264Packet</u> (<u>AMFPPacket</u> packet, <u>com.wowza.wms.media.h264.H264SEIMessages</u> 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>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
<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	<code>sendAMF3</code> (String handlerName) Call client side NetStream method/handler with no parameters.
void	<code>sendAMF3</code> (String handlerName, Object[] params) Call client side NetStream method/handler.
int	<code>sendControlBytes</code> (int controlType, java.io.OutputStream out) Send playback control bytes.
void	<code>sendDirect</code> (String handlerName) Call client side NetStream method/handler and send event to underlying stream (will record event)
void	<code>sendDirect</code> (String handlerName, Object[] params) Call client side NetStream method/handler and send event to underlying stream (will record event)
void	<code>sendDirectAMF3</code> (String handlerName) Call client side NetStream method/handler and send event to underlying stream (will record event).
void	<code>sendDirectAMF3</code> (String handlerName, Object[] params) Call client side NetStream method/handler and send event to underlying stream (will record event).
int	<code>sendLivePlaySeek</code> (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	<u>setAppend</u> (boolean isAppend) Set is append to media file (only valid if isRecord)
void	<u>setAudioSize</u> (int audioSize) Set the size of the current audio packet that is being streamed from the client to the server
void	<u>setAudioTC</u> (long audioTC) Set last absolute audio timecode (milliseconds) sent to mediaStream
void	<u>setAudioTC</u> (long audioTC, boolean isAbsolute) Set last absolute audio timecode (milliseconds) sent to mediaStream
void	<u>setBufferTime</u> (int bufferTime) Set buffer time for mediaStream (milliseconds)
void	<u>setClient</u> (<u>IClient</u> client) Set parent client connection
void	<u>setClustered</u> (boolean isClustered) not used
void	<u>setDataSize</u> (int dataSize) Set the size of the current data packet that is being streamed from the client to the server
void	<u>setDataTC</u> (long dataTC) Set last absolute data timecode (milliseconds) sent to mediaStream
void	<u>setDataTC</u> (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> (<u>IMediaStreamPlay</u> 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>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> (<u>RTPStream</u> 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	<u>setVideoTC</u> (long videoTC) Set last absolute video timecode (milliseconds) sent to mediaStream
void	<u>setVideoTC</u> (long videoTC, boolean isAbsolute) Set last absolute video timecode (milliseconds) sent to mediaStream
void	<u>shutdown</u> () shutdown or close this mediaStream
long	<u>size</u> () Get size (bytes) of media file pointed to by mediaStream
void	<u>startAudioPacket</u> () Called when an audio packet is first being populated with data
void	<u>startDataPacket</u> () Called when a data packet is first being populated with data
void	<u>startPublishing</u> () Start publishing live stream
void	<u>startVideoPacket</u> () Called when a video packet is first being populated with data
void	<u>stopName</u> (String name, String oldName, String ext, String queryStr, double playStart, double playLen, int playTransition) Stop stream name
void	<u>stopPublishing</u> () Stop publishing live stream

void	<code>switchName</code> (String name, String oldName, String ext, String queryStr, double playStart, double playLen, int playTransition) Switch to stream name
void	<code>trim</code> () Trim mediaStream.
void	<code>unregisterCallback</code> (String handlerName) Unregister a callback handler
void	<code>unregisterOnPlayStatus</code> (<code>IMediaStreamCallback</code> callback) Unregister onPlayStatus handler
void	<code>unregisterOnStatus</code> (<code>IMediaStreamCallback</code> callback) 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

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

(continued from last page)

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)

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

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

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

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 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	<code>onStreamStart</code> (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	<u>PAUSE_PAUSE</u> Pause type: pause Value: 1
public static final	<u>PAUSE_PLAY</u> Pause type: play Value: 0
public static final	<u>PAUSE_TOGGLE</u> Pause type: toggle Value: -1
public static final	<u>PLAYSIZES_AUDIO_BYTES</u> IMediaReader sizes array: audio byte count Value: 0
public static final	<u>PLAYSIZES_AUDIO_COUNT</u> IMediaReader sizes array: audio packet count Value: 1
public static final	<u>PLAYSIZES_DATA_BYTES</u> IMediaReader sizes array: data byte count Value: 4
public static final	<u>PLAYSIZES_DATA_COUNT</u> IMediaReader sizes array: data packet count Value: 5
public static final	<u>PLAYSIZES_LOSS_BYTES</u> IMediaReader sizes array: data byte count Value: 6
public static final	<u>PLAYSIZES_LOSS_COUNT</u> IMediaReader sizes array: data packet count Value: 7
public static final	<u>PLAYSIZES_SIZE</u> IMediaReader sizes array: size of sizes array long[PLAYSIZES_SIZE] Value: 8
public static final	<u>PLAYSIZES_VIDEO_BYTES</u> 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	<code>getDuration()</code> Get the recorded duration of the file in seconds
boolean	<code>isVersionFile()</code> Return true if the old file is to be versioned
boolean	<code>isWaitForVideoKeyFrame()</code> get wait for key frame
void	<code>putMetaData(String name, AMFData value)</code> Add metadata to the metadata packet.
void	<code>setMediaWriterItem(MediaWriterItem mediaWriterItem)</code> Set the media write definition
void	<code>setParent(IMediaStream parent)</code> Set the parent stream for this media write object
void	<code>setVersionFile(boolean versionFile)</code> Set to true if the old file is to be versioned
void	<code>setWaitForVideoKeyFrame(boolean waitForVideoKeyFrame)</code> Set to true if you want the recorder to skip opening frames until it hits a key frame
void	<code>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)</code> 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)

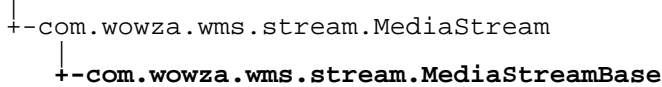
Parameters:

stream - stream
extraMetadata - additional metadata, add to this collection to add items to onMetadata event written to FLV file.

com.wowza.wms.stream

Class 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, audioBytes, audioSize, audioTC, BASE_STREAM_EXT, baseTC, bufferTime, callbacks, client, dataBytes, dataSize, dataTC, dataType, dvrLock, dvrMap, dvrPlayer, dvrRecorder, dvrRecorderList, dvrRepeater, elapsedTime, ext, fastPlaySettings, h264SEIListeners, headerSize, httpStreamerSession, ID3_STREAM_EXT, isAppend, isAudioBase, isClustered, isDataBase, isMediaCasterPlay, isOpen, isPlay, isPlaying, isRecord, isTranscodeResult, isVideoBase, liveStreamPacketizer, liveStreamPacketizerList, liveStreamRepeater, liveStreamTranscoderList, mediaIOPerformance, mergeOnMetadata, metaDataProvider, MP3_STREAM_EXT, MP4_STREAM_EXT, name, netConnection, 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, addClientListener, addClientListener, addClientListener, addDataData, addVideoCodecConfigPacket, addVideoData, 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, isPublishStreamReady, isReceiveAudio, isReceiveVideo, isRecord, isSendDirectMessages, isSendPlayStopLogEvent, isSendPublishStopLogEvent, isSendRecordStopLogEvent, isTranscodeResult, isVideoH264SEILListenerEmpty, length, notifyActionOnCodecInfoAudio, notifyActionOnCodecInfoVideo, notifyActionOnMetaData, notifyActionPause, notifyActionPauseRaw, notifyActionPlay, notifyActionPublish, notifyActionSeek, notifyActionStop, notifyActionUnPublish, notifyVideoH264Packet, packetComplete, processSendDirectMessages, publish, putDvrRecorder, putLiveStreamTranscoder, registerCallback, registerOnPlayStatus, registerOnStatus, removeClientListener, removeClientListener, removeClientListener, removeDvrRecorder, removeLiveStreamTranscoder, removeVideoH264SEILListener, reset, send, send, sendAMF3, sendAMF3, sendControlBytes, 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, setDataType, setDvrRecorder, setDvrRecorderList, setDvrRepeater, setExt, setFastPlaySettings, setHeaderSize, setHTTPStreamerSession, setIsPlaying, setJustName, setLiveStreamPacketizer, setLiveStreamPacketizerList, setLiveStreamRepeater, setLiveStreamTranscoderList, setMediaCasterPlay, setMergeOnMetadata, setMetaDataProvider, setName, setName, setName, setNetConnection, setOpen, setPlay, setPlayer, setPublishAudioCodecId, 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`](#), [`addClientListener`](#), [`addClientListener`](#), [`addClientListener`](#), [`addDataData`](#), [`addVideoCodecConfigPacket`](#), [`addVideoData`](#), [`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`](#), [`isPublishStreamReady`](#), [`isReceiveAudio`](#), [`isReceiveVideo`](#), [`isRecord`](#), [`isSendPlayStopLogEvent`](#), [`isSendPublishStopLogEvent`](#), [`isSendRecordStopLogEvent`](#), [`isTranscodeResult`](#), [`isVideoH264SEIListenerEmpty`](#), [`length`](#), [`notifyActionOnCodecInfoAudio`](#), [`notifyActionOnCodecInfoVideo`](#), [`notifyActionOnMetaData`](#), [`notifyActionPause`](#), [`notifyActionPauseRaw`](#), [`notifyActionPlay`](#), [`notifyActionPublish`](#), [`notifyActionSeek`](#), [`notifyActionStop`](#), [`notifyActionUnPublish`](#), [`notifyVideoH264Packet`](#), [`packetComplete`](#), [`publish`](#), [`putDvrRecorder`](#), [`putLiveStreamTranscoder`](#), [`registerCallback`](#), [`registerOnPlayStatus`](#), [`registerOnStatus`](#), [`removeClientListener`](#), [`removeClientListener`](#), [`removeClientListener`](#), [`removeDvrRecorder`](#), [`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`](#), [`setDataTC`](#), [`setDataType`](#), [`setDvrRecorder`](#), [`setDvrRecorderList`](#), [`setExt`](#), [`setFastPlaySettings`](#), [`setHeaderSize`](#), [`setHTTPStreamerSession`](#), [`setIsPlaying`](#), [`setLiveStreamPacketizer`](#), [`setLiveStreamPacketizerList`](#), [`setLiveStreamRepeater`](#), [`setLiveStreamTranscoderList`](#), [`setMediaCasterPlay`](#), [`setMergeOnMetadata`](#), [`setMetaDataProvider`](#), [`setName`](#), [`setName`](#), [`setName`](#), [`setNetConnection`](#), [`setOpen`](#), [`setPlay`](#), [`setPlayer`](#), [`setPublishAudioCodecId`](#), [`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

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,  
                WMSProperties 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	<u>appInstance</u>
protected	<u>dvrRecorders</u>
protected	<u>liveStreamPacketizers</u>
public static final	<u>MAXSTREAMINDEX</u> Value: 65536
protected	<u>mediaStreamListeners</u>
protected	<u>nameGroupId</u>
protected	<u>nameGroups</u>
protected	<u>nextStreamId</u>
protected	<u>packetizerLicenses</u>
protected	<u>streamLicenses</u>
protected	<u>streamLock</u>
protected	<u>streamNames</u>
protected	<u>streamNamesLock</u>
protected	<u>streamNameToGroup</u>
protected	<u>streams</u>

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.
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)
java.util.Set	getNameGroups ()

java.util.Set	<u>getNameGroups</u> (String streamName)
java.util.Set	<u>getNameGroupStreamNames</u> (String streamName)
long	<u>getNextNameGroupId</u> (MediaStreamMapGroup newGroup)
int	<u>getNextStreamIndex</u> () Reserve a clientless stream id for a new media stream.
int	<u>getNextStreamIndex</u> (<u>IClient</u> client) Reserve a stream for a client connection.
int	<u>getNextStreamIndex</u> (com.wowza.wms.netconnection.INetConnection netConnection) Reserve a stream for a netConnection connection.
java.util.List	<u>getPublishStreamNames</u> () Returns a List of published stream names
<u>IMediaStream</u>	<u>getStream</u> (<u>IClient</u> client, int index) Get a media stream reference by stream id.
<u>IMediaStream</u>	<u>getStream</u> (<u>IClient</u> client, int index, boolean doCreate) Get a media stream reference by stream id.
<u>IMediaStream</u>	<u>getStream</u> (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).
<u>IMediaStream</u>	<u>getStream</u> (com.wowza.wms.netconnection.INetConnection netConnection, int index, boolean doCreate) Get a media stream reference by stream id.
<u>IMediaStream</u>	<u>getStream</u> (String name) Get a media stream by stream name.
<u>IMediaStream</u>	<u>getStreamClientless</u> (int index, String streamTypeStr) Get a media stream reference by stream id.
edu.emory.mathcs.backport.java.util.concurrent.locks.WMSReadWriteLock	<u>getStreamListLock</u> () Get the underlying read/write lock associated with the list of streams
edu.emory.mathcs.backport.java.util.concurrent.locks.WMSReadWriteLock	<u>getStreamNameLock</u> () Get the underlying read/write lock associated with the stream names list
java.util.List	<u>getStreams</u> () Returns a list of IMediaStream objects
<u>IVHost</u>	<u>getVHost</u> () Get the parent vHost.
void	<u>notifyMediaStreamCreate</u> (<u>IMediaStream</u> mediaStream) Notify all media stream listeners that a new media stream object has been created.

void	<u>notifyMediaStreamDestroy</u> (<u>IMediaStream</u> mediaStream) Notify all media stream listeners that a media stream object is being destroyed.
void	<u>notifyPlayPublish</u> (<u>IMediaStream</u> 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	<u>notifyPlayUnpublish</u> (<u>IMediaStream</u> 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	<u>notifyPlayUnpublish</u> (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	<u>removeDvrRecorder</u> (String streamName)
<u>ILiveStreamDvrRecorder</u>	<u>removeDvrRecorder</u> (String streamName, String recorderName) Remove DVR Recorder
void	<u>removeLiveStreamPacketizer</u> (String streamName) Remove all live stream packetizers for this stream name
<u>ILiveStreamPacketizer</u>	<u>removeLiveStreamPacketizer</u> (String streamName, String packetizerName) Remove live stream packetizer
void	<u>removeMediaStreamListener</u> (<u>IMediaStreamNotify</u> mediaStreamListener) Remove a media stream listener.
MediaStreamMapGroup	<u>removeNameGroup</u> (int groupId)
MediaStreamMapGroup	<u>removeNameGroup</u> (MediaStreamMapGroup nameGroup)
void	<u>removeStream</u> (<u>IClient</u> client, int index) Remove a stream associated with a client connection
void	<u>removeStream</u> (com.wowza.wms.netconnection.INetConnection netConnection, int index) Remove a stream associated with a netConnection object
void	<u>removeStream</u> (int index) Remove a clientless media stream
void	<u>setStreamName</u> (<u>IMediaStream</u> stream, String name) Insert live media stream into the mediaStreamMap by name.
long	<u>streamToIndex</u> (<u>IMediaStream</u> 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

(continued from last page)

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
```

streamLock

```
protected edu.emory.mathcs.backport.java.util.concurrent.locks.WMSReadWriteLock  
streamLock
```

(continued from last page)

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**

dvrRecorders

protected java.util.Map **dvrRecorders**

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:

(continued from last page)

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()
```

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).

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

(continued from last page)

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

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

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

(continued from last page)

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

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)

broadcastPlayMessage

```
public 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.

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

(continued from last page)

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

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:

(continued from last page)

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)
```

(continued from last page)

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

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)
```

Package

com.wowza.wms.stream.livedvr

com.wowza.wms.stream.livedvr Interface IDvrStreamManagerActionNotify

public interface **IDvrStreamManagerActionNotify**
extends

IDvrActionNotify notify interface for dvr stores. See IApplicationInstance.addDvrStoreListener(IDvrStoreActionNotify storeListener)

Method Summary

void	<code>onDvrStreamManagerCreate</code> (<code>IDvrStreamManager</code> dvrMgr) Called when DVR stream manager created but before it is initialized.
void	<code>onDvrStreamManagerDestroy</code> (<code>IDvrStreamManager</code> dvrMgr) Called when DVR stream manager destroyed.
void	<code>onDvrStreamManagerInit</code> (<code>IDvrStreamManager</code> 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 **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>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.

Parameters:

streamName - stream name

(continued from last page)

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

(continued from last page)

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()
```

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

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
IMediaStream	getStartStream() Get the current stream that is being packetized
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	setPacketizeAudio (boolean packetizeAudio) Set to true to packetize audio
void	setPacketizeData (boolean packetizeVideo) Set to true to packetize data
void	setPacketizeVideo (boolean packetizeVideo) Set to true to packetize video
void	setRepeaterEdge (boolean isRepeaterEdge) Set is live repeater edge
void	shutdown () Called to shutdown the live stream packetizer
void	startStream (IMediaStream stream) Called when the stream starts
void	touch (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

getStartStream

```
public IMediaStream getStartStream()
```

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.

Method Summary

void	<u>close</u> (<u>IMediaStream</u> stream) Called when live stream transcoder is stream is closed
<u>IApplicationInstance</u>	<u>getAppInstance</u> () Get the application instance associated with this live stream transcoder.
String	<u>getContextStr</u> () Get the streaming context for this live stream transcoder.
LicenseHolder	<u>getLicenseHolder</u> () Get license holder.
LiveStreamTranscoderItem	<u>getLiveStreamTranscoderItem</u> () Get the definition for live stream transcoder.
<u>WMSProperties</u>	<u>getProperties</u> () Get the user properties
String	<u>getStreamName</u> () Get the stream name of the source stream.
String	<u>getTranscoderName</u> () Get the live stream transcoder name
void	<u>handleOnMetadata</u> (<u>IMediaStream</u> stream, <u>AMFPacket</u> 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, LiveStreamTranscoderItem 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	setAppInstance (IApplicationInstance appInstance) Set the application instance associated with this live stream transcoder.
void	setLiveStreamTranscoderItem (LiveStreamTranscoderItem liveStreamTranscoderItem) Set the definition for live stream transcoder.
void	setStreamName (String streamName) Set source stream name.
void	setTranscoderName (String transcoderName) Get the live stream transcoder name
void	shutdown (IMediaStream stream) Called when live stream transcoder is shutdown

Methods

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)

(continued from last page)

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

(continued from last page)

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)
```

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()
```

(continued from last page)

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()
```

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

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	<code>isLiveStreamTranscode</code> (String transcoder, <code>IMediaStream</code> 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	Playlist (String sName) Class constructor - A simple structure that maintains a list of playlist items.
--------	--

Method Summary

void	addItem (String sName, int start, int length) Appends an item to this playlist
java.util.List	getItems () Get the items in the playlist (returns a copy of the list)
String	getName () Returns the name of this playlist as defined in the XML definition file
boolean	getRepeat ()
boolean	open (Stream s) Opens this playlist on the given stream...
void	removeItem (int index) Remove an item from the list
void	setRepeat (boolean repeat)

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) Add audio data
void	addAudioData (byte[] data, int len, long timecode) Add audio data
void	addAudioData (byte[] data, long timecode) Add audio data
void	addAudioDataInc (byte[] data, int offset, int len)
void	addDataData (byte[] data, int offset, int len, long timecode) Add metadata
void	addDataData (byte[] data, int len, long timecode) Add metadata
void	addDataData (byte[] data, long timecode) Add metadata
void	addDataDataInc (byte[] data, int offset, int len)
void	addVideoData (byte[] data, int offset, int len, long timecode) Add video data
void	addVideoData (byte[] data, int len, long timecode) Add video data
void	addVideoData (byte[] data, long timecode) Add video data
void	addVideoDataInc (byte[] data, int offset, int len)
void	close () Close the publisher
static Publisher	createInstance (IApplicationInstance appInstance)
static Publisher	createInstance (IVHost vhost, String applicationName)
static Publisher	createInstance (IVHost vhost, String applicationName, String appInstanceName)
void	createStream () Create underlying IMediaStream object if not already created
void	flush () Flush the packets from the input buffer to the output buffer
IApplicationInstance	getAppInstance ()
String	getFileExtension () Get the file extension (default flv)
long	getLastAudioTimecode () Get last audio timecode written through this publisher (milliseconds).

long	getLastDataTimecode() Get last data timecode written through this publisher (milliseconds).
long	getLastVideoTimecode() Get last video timecode written through this publisher (milliseconds).
long	getMaxTimecode() Highest timecode written through this publisher (milliseconds).
IMediaStream	getStream() Get the media stream object
String	getStreamType()
void	publish() (String streamName) Publish a stream (null to stop publishing)
void	publish() (String streamName, String howToPublish) Start publishing a stream (streamName = null to stop).
void	setFileExtension() (String fileExtension) Set the file extension
void	setStreamType() (String streamType) Set the stream type (default live)
void	startAudioData() (int len, long timecode)
void	startDataData() (int len, long timecode)
void	startVideoData() (int len, long timecode)
void	unpublish()

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)

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

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)

(continued from last page)

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)

startVideoData

```
public void startVideoData(int len,  
    long timecode)
```

addVideoDataInc

```
public void addVideoDataInc(byte[] data,  
    int offset,  
    int len)
```

startAudioData

```
public void startAudioData(int len,  
    long timecode)
```

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)

(continued from last page)

addAudioData

```
public void addAudioData(byte[] data,  
    int len,  
    long timecode)
```

Add audio data

Parameters:

data - data
len - data length
timecode - absolute timecode (milliseconds)

addAudioData

```
public void addAudioData(byte[] data,  
    int offset,  
    int len,  
    long timecode)
```

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)
```

Add metadata

(continued from last page)

Parameters:

data - data
offset - offset
len - data length
timecode - absolute timecode (milliseconds)

startDataData

```
public void startDataData(int len,  
    long timecode)
```

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:

(continued from last page)

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).

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)
```

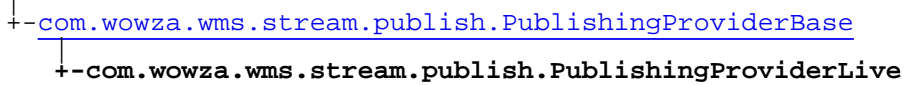
True to send onMetadata event on stream start

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	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)

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)
```

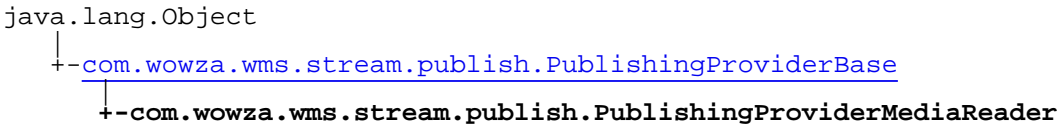
getStartOnPreviousBufferTime

```
public long getStartOnPreviousBufferTime()
```

setStartOnPreviousBufferTime

```
public void setStartOnPreviousBufferTime(long startOnPreviousBufferTime)
```

com.wowza.wms.stream.publish
Class PublishingProviderMediaReader



All Implemented Interfaces:
[IPublishingProvider](#)

public class **PublishingProviderMediaReader**
extends [PublishingProviderBase](#)
implements [IPublishingProvider](#)

PublishingProviderMediaReader: Publishing provider for media reader based stream (vod).

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 ()
long	getDuration () Get the target playback duration (milliseconds)
boolean	play (Publisher publisher)
boolean	seek (long timecode)
boolean	seek (long timecode, int seekType)
void	setDuration (long duration)
void	setRealTimeStartTime (long realTimeStartTime)

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

PublishingProviderMediaReader

```
public PublishingProviderMediaReader(Publisher publisher,  
                                   long timeoffset,  
                                   String streamName)
```

Constructor

Parameters:

`publisher` - publisher
`timeoffset` - timeoffset (milliseconds)
`streamName` - stream name

Methods

seek

```
public boolean seek(long timecode)
```

seek

```
public boolean seek(long timecode,  
                   int seekType)
```

play

```
public boolean play(Publisher publisher)
```

close

```
public void close()
```

getDuration

```
public long getDuration()
```

Get the target playback duration (milliseconds)

(continued from last page)

Returns:

playback duration (milliseconds)

setDuration

```
public void setDuration(long duration)
```

setRealTimeStartTime

```
public void setRealTimeStartTime(long realTimeStartTime)
```

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	getName() Returns the name of the playlist stream - the client would play this stream by this name.
java.util.List	getPlaylist() Get the current playlist
int	getPollingInterval() Get the polling interval (milliseconds)
Publisher	getPublisher()
boolean	getRepeat() Use this to determine if the playlist is auto-repeating
long	getStartLiveOnPreviousBufferTime() Get time in milliseconds to go back in live stream buffer to get previous key frame
int	getTimeOffsetBetweenItems() Get time in milliseconds to add to stream time between playlist items (default is zero)
boolean	isSendOnMetadata() True if sending onMetadata events
boolean	isStartLiveOnPreviousKeyFrame() Set to true to start live streams on most recent key frame (smoother switching)
boolean	isSwitchLog() Log when a playlist switch occurs
boolean	isTimesInMilliseconds() If true start time and duration and are milliseconds.
void	next()
void	next(int n)
void	play(int n)
boolean	play(String sPlaylist) Add a media item to the playlist as defined by an XML file ..
boolean	play(String name, int start, int length, boolean reset) Adds a media source item to this playlist -
void	previous()
void	previous(int n)
boolean	removeFromPlaylist(int index) Remove item from playlist based on index.
boolean	removeFromPlaylist(String name) Remove all items matching the given stream name from the playlist.
void	removeListener(IStreamActionNotify listener) Remove a listener

void	run() Overridden from class Runnable
void	setPollingInterval (int pollingInterval) Set the polling interval (milliseconds)
void	setRepeat (boolean repeat) Use this to make the playlist repeat or not...
void	setSendOnMetadata (boolean sendOnMetadata) True if sending onMetadata events
void	setStartLiveOnPreviousBufferTime (long startLiveOnPreviousBufferTime) Set time in milliseconds to go back in live stream buffer to get previous key frame
void	setStartLiveOnPreviousKeyFrame (boolean startLiveOnPreviousKeyFrame) Set to true to start live streams on most recent key frame (smoother switching)
void	setSwitchLog (boolean switchLog) Log when a playlist switch occurs
void	setTimeOffsetBetweenItems (int timeOffsetBetweenItems) Set time in milliseconds to add to stream time between playlist items (default is zero)
void	setTimesInMilliseconds (boolean timesInMilliseconds) If true start time and duration are in milliseconds.

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

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

(continued from last page)

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:

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

(continued from last page)

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

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

(continued from last page)

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

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.

(continued from last page)

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)
```

next

```
public void next()
```

previous

```
public void previous()
```

previous

```
public void previous(int n)
```

(continued from last page)

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:

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:

(continued from last page)

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

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

(continued from last page)

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

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	<code>onCalculateSourceAudioBitrate</code> (<code>LiveStreamTranscoder liveStreamTranscoder, long bitrate</code>) Called when the bitrate of the source audio stream is calculated
void	<code>onCalculateSourceVideoBitrate</code> (<code>LiveStreamTranscoder liveStreamTranscoder, long bitrate</code>) Called when the bitrate of the source video stream is calculated
void	<code>onInitAfterLoadTemplate</code> (<code>LiveStreamTranscoder liveStreamTranscoder</code>) Called just after transcoder template is loaded.
void	<code>onInitBeforeLoadTemplate</code> (<code>LiveStreamTranscoder liveStreamTranscoder</code>) Called just before transcoder template is loaded
void	<code>onInitStart</code> (<code>LiveStreamTranscoder liveStreamTranscoder, String streamName, String transcoderName, IApplicationInstance appInstance, LiveStreamTranscoderItem liveStreamTranscoderItem</code>) Call when live stream transcoder is first created
void	<code>onInitStop</code> (<code>LiveStreamTranscoder liveStreamTranscoder</code>) At the end of the initialization process
void	<code>onRegisterStreamNameGroup</code> (<code>LiveStreamTranscoder liveStreamTranscoder, TranscoderStreamNameGroup streamNameGroup</code>) Called after a stream name group is resolved and registered with MediaStreamMap
void	<code>onResetStream</code> (<code>LiveStreamTranscoder liveStreamTranscoder</code>) Called when the stream feeding the live stream transcoder switches.
void	<code>onSessionAudioDecodeCodecInfo</code> (<code>LiveStreamTranscoder liveStreamTranscoder, com.wowza.wms.media.model.MediaCodecInfoAudio codecInfoAudio</code>) Called when audio decoding information is available.
void	<code>onSessionAudioEncodeCodecInfo</code> (<code>LiveStreamTranscoder liveStreamTranscoder, TranscoderSessionAudioEncode sessionAudioEncode, com.wowza.wms.media.model.MediaCodecInfoAudio codecInfoAudio</code>) Called when audio encoding information is available.
void	<code>onSessionAudioEncodeCreate</code> (<code>LiveStreamTranscoder liveStreamTranscoder, TranscoderSessionAudioEncode sessionAudioEncode</code>) Called when audio encoder session is created.
void	<code>onSessionAudioEncodeInit</code> (<code>LiveStreamTranscoder liveStreamTranscoder, TranscoderSessionAudioEncode sessionAudioEncode</code>) 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 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	onAfterEncodeFrame (TranscoderSessionVideoEncode sessionVideoEncode, TranscoderStreamDestinationVideo destinationVideo, long frameCount)
void	onBeforeEncodeFrame (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

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()**

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 (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 void	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 RTTPushPublishSession	startRTTPull (IApplicationInstance appInstance, String streamName, boolean streamPacketizer, String ipAddress, int audioPort, int videoPort, boolean isRTPWrapped) Start pushing an RTP stream
static RTTPushPublishSession	startRTTPull (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.

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

decodeRangeHeader

```
public static double[] decodeRangeHeader(String rangeStr)
```

Decode RTP range header, Internal use.

Parameters:

rangeStr

Returns:

range values

(continued from last page)

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

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

(continued from last page)

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

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

loadConfigFile

```
public static void 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:

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 void	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 void loadConfigFile(StreamList streamDefs,  
    String fileURL)
```

Load Strreams.xml, Internal use.

Parameters:

streamDefs
fileURL

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.

Constructor Summary

public	HostPort() Create an empty HostPort object
--------	---

Method Summary

void	addHttpProvider(IHTTPProvider httpProvider)
void	addHttpProvider(IHTTPProvider2 httpProvider)
void	addHTTPStreamerAdapterID(String ID)
void	configureSocketAcceptor(org.apache.mina.transport.socket.nio.SocketAcceptorConfig socketConfig) Configure a socketAcceptor
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
java.util.List	getHttpProviders()
java.util.List	getHTTPStreamerAdapterIDs()
int	getPort() Get port
int	getProcessorCount() Get the number of threads to use to service this incoming port

HostPortSSLConfig	<code>getSSLConfig()</code>
String	<code>getSslFactoryClass()</code> Get full class name or SSLFactory class
boolean	<code>isSuspended()</code>
void	<code>setDomainName(String domainName)</code> Set domainName.
void	<code>setIpAddress(String ipAddress)</code> Set ipAddress for object.
void	<code>setPort(int port)</code> Set port
void	<code>setProcessorCount(int processorCount)</code> Set the number of threads to use to service this incoming port
void	<code>setSSLConfig(HostPortSSLConfig sslConfig)</code>
void	<code>setSslFactoryClass(String sslFactoryClass)</code> Set full class name of SSLFactory class
void	<code>setSuspended(boolean isSuspended)</code>
String	<code>toString()</code> Return object as formatted string
String	<code>toString(boolean mBeanSafe)</code> Return object as formatted string

Methods inherited from class `java.lang.Object`

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

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

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

(continued from last page)

Returns:

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:

(continued from last page)

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)
```

isSuspended

```
public boolean isSuspended()
```

setSuspended

```
public void setSuspended(boolean isSuspended)
```

getHTTPStreamerAdapterIDs

```
public java.util.List getHTTPStreamerAdapterIDs()
```

addHTTPStreamerAdapterID

```
public void addHTTPStreamerAdapterID(String ID)
```

getSSLConfig

```
public HostPortSSLConfig getSSLConfig()
```

(continued from last page)

setSSLConfig

```
public void setSSLConfig(HostPortSSLConfig sslConfig)
```

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	<code>onAcceptorCreate</code> (<code>HostPort</code> hostPort, java.util.Map acceptorMap) Triggered when a new acceptor is created
void	<code>onAcceptorDestroy</code> (<code>HostPort</code> 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_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_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	COUNTER_RTP 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	<u>LICENSECOUNTER_TRANSCODE_DECODECOUNTAUDIO</u> 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	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	closeHostPort (HostPort hostPort, boolean isSuspend) Close an individual HostPort
boolean	createApplication (String sName, String sStreamType, String sContentLoc) Method to create a new application

<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
<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>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	<u>getKeepAliveTimeout()</u> Get the RTMPT connection keep alive timeout
LiveStreamPacketizerList	<u>getLiveStreamPacketizerList()</u> Get the LiveStreamPacketizerList
LiveStreamTranscoderList	<u>getLiveStreamTranscoderList()</u> Get the LiveStreamTranscoderList

int	<u>getMaximumPendingReadBytes</u> () Set maximum number of bytes a client connection can have waiting to be written before the connection is terminated.
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>MediaCasterList</u>	<u>getMediaCasterList</u> () Get the list of media caster definitions (MediaCaster.xml)
<u>MediaCasterSettings</u>	<u>getMediaCasterSettings</u> () Get the media caster settings
int	<u>getMediaReaderContentType</u> (String mediaType) Get the content type of a media stream name prefix (see IMediaReader.CONTENTTYPE_*)
MediaReaderList	<u>getMediaReaders</u> () Get the media readers attached to vHost (MediaReaders.xml).
MediaWriterList	<u>getMediaWriters</u> () Get the media writers attached to vHost (MediaWriters.xml).
java.util.Map	<u>getMp3TagMap</u> () Get MP3 tag map attached to vHost (MP3Tags.xml).
String	<u>getName</u> () Get vHost name
HostPortConfig	<u>getNetConnectionHostPortConfig</u> () Get the socket configuration for server to server connections
int	<u>getNetConnectionIdleFrequency</u> () Get server to server idle frequency (milliseconds)
int	<u>getNetConnectionProcessorCount</u> () 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</u> (String key) Get virtual host property.
void	<u>getProtocolUsage</u> (boolean[] protocolsInUse) Get the protocols in use by this application instance (see IApplicationInstance.PROTOCOLUSAGE_*)
<u>RTPContext</u>	<u>getRTPContext</u> () Get the RTP (real time protocol) virtual host context

com.wowza.wms.rtp.transport.RTPUDPDatagramConfig	getRTPDatagramConfigIncoming() Get the RTP Datagram Socket configuration
com.wowza.wms.rtp.transport.RTPUDPDatagramConfig	getRTPDatagramConfigOutgoing() Get the RTP Datagram Socket configuration
int	getRTPIdleFrequency() Get default RTP idle frequency (milliseconds)
java.util.List	getStartupStreams() Get the list of streams to start at virtual host startup
StreamList	getStreamTypes() Get default stream type.
ThreadPool	getThreadPool() Get the VHost handler thread pool.
String	getTimeRunning() Get the time vHost has been running.
double	getTimeRunningSeconds() Get time running in seconds
ThreadPool	getTransportThreadPool() Get the VHost transport thread pool.
com.wowza.wms.rtp.transport.UDPTransportManager	getUDPTransportManager() Get the UDP transport manager.
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

String	<u>readVHostConfig</u> (String sName) Method to read xml config file..
void	<u>removeAcceptorListener</u> (<u>IAcceptorNotify</u> acceptorListener) Remove acceptor listener.
boolean	<u>removeApplication</u> (String sName) Method to remove an application
void	<u>removeApplicationListener</u> (<u>IApplicationNotify</u> applicationListener) Remove applation listener.
void	<u>removeClient</u> (int clientId) Remove client from vHost.
void	<u>removeIdleWorkerListener</u> (<u>IIIdleWorkerNotify</u> idleWorkerListener) Remove idleWorker listener
void	<u>reparentClient</u> (<u>IClient</u> client) Move a client object to a new vhost.
void	<u>setAdminInterfaceHostPort</u> (<u>HostPort</u> adminInterfaceHostPort) Set admin interface host port (not used)
void	<u>setApplicationTimeout</u> (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>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>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>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.
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	unbindAllHostPorts() Unbind all HostPorts and drop all connections
void	updateLoggingDuration() Internal: update the internal logging values.
boolean	writeVHostConfig(String sName, String data) Method to write xml config file..

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
```

(continued from last page)

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**

(continued from last page)

LICENSECOUNTER_TRANSCODE_DECODECOUNTAUDIO

```
public static final int LICENSECOUNTER_TRANSCODE_DECODECOUNTAUDIO
```

Constant value: **14**

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**

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**

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**

(continued from last page)

CODEC_VIDEO_MPEG4

```
public static final int CODEC_VIDEO_MPEG4
```

Constant value: **10**

CODEC_VIDEO_MPEG2

```
public static final int CODEC_VIDEO_MPEG2
```

Constant value: **11**

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_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
```

(continued from last page)

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**

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
```

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**

(continued from last page)

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_SHARED OBJECTS

public static final byte **CONTENTTYPE_SHARED OBJECTS**

(continued from last page)

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**

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**()

(continued from last page)

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.

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

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.

(continued from last page)

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

(continued from last page)

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)
```

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 ( )
```

(continued from last page)

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 file 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_*)

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.

(continued from last page)

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

getMp3TagMap

```
public java.util.Map getMp3TagMap()
```

Get MP3 tag map attached to vHost (MP3Tags.xml).

Returns:

MP3 tag map attached to vHost

updateLoggingDuration

```
public void updateLoggingDuration()
```

Internal: update the internal logging values.

getApplicationTimeout

```
public int getApplicationTimeout()
```

(continued from last page)

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:

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

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.

(continued from last page)

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)

getRTPIdleFrequency

```
public int getRTPIdleFrequency()
```

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)

(continued from last page)

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)
```

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(IIdleWorkerNotify idleWorkerListener)
```

(continued from last page)

Add idleWorker listener. An idleWorker listener will receive the following events: onIdleWorkerCreate, onIdleWorkerDestroy.

Parameters:

idleWorkerListener - idleWorker listener

removeIdleWorkerListener

```
public void removeIdleWorkerListener(IIdleWorkerNotify 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()
```

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

(continued from last page)

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

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

(continued from last page)

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:

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.

(continued from last page)

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

(continued from last page)

Returns:

LiveStreamTranscoderList

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

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

createApplication

```
public boolean createApplication(String sName,  
    String sStreamType,  
    String sContentLoc)
```

Method to create a new application

removeApplication

```
public boolean removeApplication(String sName)
```

Method to remove an application

readVHostConfig

```
public String readVHostConfig(String sName)
```

Method to read xml config file..

writeVHostConfig

```
public boolean writeVHostConfig(String sName,  
    String data)
```

Method to write xml config file..

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

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_*)

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 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	VHostList() Create empty vHostList
--------	---

Method Summary

void	addVHostItemListener() (IVHostItemNotify vHostItemListener) Add vHostItem listener.
java.util.List	getVHostItems() Get a list of vHostItems.
java.util.Map	getVHostMap() Get the Map of vHostItem definitions.
java.util.List	getVHostNames() Get a list of vHost names.
void	loadConfig()
void	notifyVHostItemCreate() (VHostItem vhostItem) Notify vHostItem listener of item create.
void	notifyVHostItemDestroy() (VHostItem vhostItem) Notify vHostItem listener of item destroy.
void	notifyVHostItemUpdate() (VHostItem vhostItem) Notify vHostItem listener of item update.
void	reloadConfig() Reload VHosts.xml file.
void	removeVHostItemListener() (IVHostItemNotify 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 598
- ABSOLUTE_TIME_PATTERN 598
- acceptConnection 320, 337
- ACCEPTORS_ACCEPTOR 1307
- ACCEPTORS_HANDLERADAPTER 1308
- acceptSession 561, 908
- acquire 744, 751, 752, 753, 1011, 1035
- acquireAndRelease 744, 752
- acquireMediaCaster 758
- acquireSocketAddress 876
- acquireUDPPortPair 876, 877
- actionListeners 906
- activateOptions 608
- add 77, 136, 137, 138, 171, 172, 173, 862, 1283
- addAcceptConnectionAttribute 351
- addAcceptorListener 1313
- addActionListener 917
- addApplicationInstanceListener 233
- addApplicationListener 1313
- addAudioCodecConfigPacket 1110
- addAudioData 1095, 1226, 1227
- addAudioDataInc 1226
- addBody 859
- addBouncyCastleSecurityProvider 98
- addBytes 859
- addChunk 204
- addClient 1011, 1035
- addClientListener 256, 1101, 1102
- addConnectionListener 319
- addContentDescriptor 697
- addData 215
- addDataData 1095, 1227
- addDataDataInc 1228
- addDataEx 218
- addDifference 77
- addDvrChunkListener 497
- addDvrRecorderListener 293
- addDvrRecordingListener 486
- addDvrStoreListener 486
- addDvrStreamManagerListener 294
- addFirstABSTRequest 584
- addHttpProvider 1281
- addHTTPStreamerAdapterID 1281
- addHTTPStreamerSession 281
- addIdleWorkerListener 1319
- addIOPerformance 557, 914
- addIOPerformance2 557, 914
- addItem 1214
- addLicense 1177
- addListener 1245
- addLiveStreamPacketizerListener 289
- addLiveStreamTranscoderListener 290
- addManifestEntries 485, 496
- addMediaCasterListener 260
- addMediaReaderListener 286
- addMediaStreamListener 258, 1174
- addMediaWriterListener 284
- addMember 198
- addMetaDataListener 766
- addModuleListener 264
- addNameGroup 1176
- addObject 153
- addOverlay 1258
- addPlayStreamByName 262
- addPublisher 279
- addRange 390
- addRendition 785
- addRepeaterHeartBeatItem 488
- addRTPIncomingDatagramPortAll 282
- addRTPIncomingDatagramPortRange 281
- addRTPSession 274
- addRTSPStream 909
- addSegment 772
- addServerListener 980, 997
- addSession 928
- addSessionListener 928
- addSharedObjectListener 258, 1023
- addSlotListener 1015, 1039
- addSorterPacket 1264
- addStartupStream 1326
- addStreamAttribute 947
- addStreamDomainStr 565
- addStreamDomainStrs 565
- addStreamInfo 947
- addStreamSrcToMediaCaster 754
- addString 152

addToCache 403
addToChunk 464, 574
addToManifest 451
addToPlaylist 1242
addTrack 944
addTrackId 945
addTrackInternal 944
addTrailingZero 701
addTrait 153
addUserHTTPHeaders 569
addVHostItemListener 1345
addVHostListener 1349
addVideoCodecConfigPacket 1110
addVideoData 1094, 1225
addVideoDataInc 1226
addVideoH264SEILListener 1118
adjustFirstPacketTCs 63
alphas 30
AMF3Utils 113
AMF_LEVEL0 124
AMF_LEVEL3 124
AMFData 125
AMFDataArray 134, 135
AMFDataByteArray 145, 146
AMFDataContextDeserialize 151
AMFDataContextSerialize 156
AMFDataItem 162, 163, 164
AMFDataList 170, 171
AMFDataMixedArray 182, 183
AMFDataObj 190, 191
AMFDataTrait 198
AMFDEBUGHEADERSIZE 203
AMFFORCETYPE1 203
AMFObj 203
AMFObjChunk 210
AMFPacket 213, 214
AMFUtils 11
announce 951
append 609
appendParamsToUrl 103
appInstance 847, 904, 1168
appInstanceShutdown 904
APPLICATION_KEY 597
applicationExists 1321
ARCHIVE_STRATEGY_APPEND 420
ARCHIVE_STRATEGY_DELETE 420
ARCHIVE_STRATEGY_VERSION 420
artifact 371, 393
assembleQueryStr 70
attachToWMMSession 950
audioCodecId 777
audioCodecStringToId 55
audioCodecToMetaDataString 56
audioCodecToString 56
audioCodecTypeToString 89
audioPacketizers 907
AUDIOSAMPLE_ACCESS_ALL 333
AUDIOSAMPLE_ACCESS_NONE 333
AUDIOSAMPLEACCESS 1078
authenticateHandler 509
authenticateHTTPProvider 310
authenticateHTTPProviderHandler 509
authenticatePlayHandler 904
authenticatePublishHandler 904
authenticateRTSP 311
authenticateSIP 312
AuthenticateUsernamePasswordProviderBase 306
authenticationMethod 509
AUTHMETHOD_PLAY 903
AUTHMETHOD_PUBLISH 903
AUTHMETHOD_UNKNOWN 903
AVSYNCMETHODS_RTPTIMECODE 941
AVSYNCMETHODS_SENDERREPORT 941
AVSYNCMETHODS_SYSTEMCLOCK 941
AVSYNCMETHODS_UNKNOWN 941
avSyncNameToId 957

B

badStreams 848
bind 889
bitrateAudio 777
bitrateTotal 777
bitrateVideo 777
booleanValue 165
broadcastPlayMessage 1174
broadcastGetObjectEncoding 1174
broadcastMsg 261
buffer 210
BufferUtils 30

- byteArrayToFile 41
- byteArrayToInt 32, 33
- byteArrayToIntSafeSync 706
- byteArrayToLong 31, 32
- byteArrayToLongSafeSync 706
- byteArrayToShort 33, 34
- byteArrayToString 31
- byteStringLen 676
- byteValue 165
- C
 - cacheRTPStream 877
 - calcBitrate 466
 - calcTotalPacketSize 218
 - calculateChunkGroupTime 402
 - call 340
 - CALLBACK_PARAM1 811
 - CALLBACK_PARAM10 812
 - CALLBACK_PARAM2 811
 - CALLBACK_PARAM3 811
 - CALLBACK_PARAM4 811
 - CALLBACK_PARAM5 811
 - CALLBACK_PARAM6 811
 - CALLBACK_PARAM7 811
 - CALLBACK_PARAM8 812
 - CALLBACK_PARAM9 812
 - canHandle 510, 517, 538, 973
 - canPlay 493
 - canRecord 484, 493
 - canRecordAudio 464, 1183
 - canRecordData 464, 1184
 - canRecordVideo 464, 1184
 - CAT_ALL 649
 - CAT_application 637
 - CAT_cupertino 638
 - CAT_dvrchunk 638
 - CAT_mpegdash 639
 - CAT_rtsp 638
 - CAT_sanjose 638
 - CAT_server 637
 - CAT_session 638
 - CAT_smoothstreaming 638
 - CAT_stream 638
 - CAT_transcoder 639
 - CAT_vhost 637
 - CAT_webm 638
 - checkAndSetPlayLogged 556
 - checkSendMetadata 952
 - cleanUp 608
 - clear 77, 653, 863, 1016, 1034, 1108, 1331
 - clearByteContainer 207
 - clearFastPlaySettings 342, 1098
 - clearIntData 152
 - clearLoggingValues 532, 559, 583, 589, 917, 1107
 - clearOverlay 1258
 - clearProperty 732, 1334
 - clearRTSPSessionExtraLines 969
 - clearSegments 772, 785
 - clearStreamName 1171
 - clearStreamSrcToMediaCaster 754
 - clearTracks 944
 - client 306
 - clone 77, 198, 214, 446, 1048
 - cloneProperties 300
 - close 7, 27, 608, 1011, 1040, 1052, 1059, 1097, 1142, 1202, 1209, 1228, 1233, 1236, 1243
 - closeAndWait 1243
 - closeHostPort 1323
 - closeStream 831
 - CODEC_AUDIO_AAC 1304
 - CODEC_AUDIO_G711_ALAW 1304
 - CODEC_AUDIO_G711_MULAW 1304
 - CODEC_AUDIO_MP3 1303
 - CODEC_AUDIO_MP3_8 1304
 - CODEC_AUDIO_NELLYMOSER 1304
 - CODEC_AUDIO_NELLYMOSER_16MONO 1303
 - CODEC_AUDIO_NELLYMOSER_8MONO 1303
 - CODEC_AUDIO_PCM_BE 1303
 - CODEC_AUDIO_PCM_LE 1303
 - CODEC_AUDIO_PCM_SWF 1303
 - CODEC_AUDIO_RESERVED 1304
 - CODEC_AUDIO_SPEEX 1304
 - CODEC_AUDIO_UNKNOWN 1303
 - CODEC_AUDIO_VORBIS 1304
 - CODEC_STREAM_MP2T 1305
 - CODEC_STREAM_UNKNOWN 1305
 - CODEC_TYPE 450
 - CODEC_VIDEO_H263 1302
 - CODEC_VIDEO_H264 1302

CODEC_VIDEO_MPEG2 1303
CODEC_VIDEO_MPEG4 1302
CODEC_VIDEO_SCREEN 1302
CODEC_VIDEO_SCREEN2 1302
CODEC_VIDEO_SPARK 1302
CODEC_VIDEO_UNKNOWN 1301
CODEC_VIDEO_VP6 1302
CODEC_VIDEO_VP6A 1302
CODEC_VIDEO_VP8 1302
CODES_HREF 598
compress 148
CONFIGURATOR_CLASS_KEY 597
configureSocketAcceptor 1281
ConnectionCounter 319
connectionHolder 905
containsDvrRecorder 278
containsHeader 525
containsHTTPStreamer 277
containsIndex 531, 582, 588
containsKey 191, 221
containsLiveStreamPacketizer 278
containsLiveStreamTranscoder 291
containsProperty 1016, 1036
containsSlot 1017, 1036
containsStreamDomainStr 565
containsStreamNameParts 563
containsTime 505
CONTENTTYPE_ACKBANDWIDTH 1306
CONTENTTYPE_AUDIO 1306
CONTENTTYPE_BUFFERSIZE 1305
CONTENTTYPE_CHORD 694
CONTENTTYPE_DATA 1306
CONTENTTYPE_DATA0 1306
CONTENTTYPE_DATA3 1306
CONTENTTYPE_EVENTS 694
CONTENTTYPE_FUNCTION 1307
CONTENTTYPE_FUNCTION0 1307
CONTENTTYPE_FUNCTION3 1307
CONTENTTYPE_LYRICS 694
CONTENTTYPE_MEDIA 1056
CONTENTTYPE_MEDIACHUNK 1307
CONTENTTYPE_MEDIALIST 1056
CONTENTTYPE_MOVEMENT 694
CONTENTTYPE_OTHER 694
CONTENTTYPE_PLAYCALLBACK 1307
CONTENTTYPE_SETBANDWIDTH 1306
CONTENTTYPE_SETCHUNKSIZE 1305
CONTENTTYPE_SHAREDOBJECTS 1306
CONTENTTYPE_SHAREDOBJECTS0 1307
CONTENTTYPE_SHAREDOBJECTS3 1307
CONTENTTYPE_TRANSCRIPTION 694
CONTENTTYPE_TRIVIA 695
CONTENTTYPE_UNKNOWN 1305
CONTENTTYPE_URLIMAGES 695
CONTENTTYPE_URLWEBPAGES 695
CONTENTTYPE_VIDEO 1306
CONTENTTYPE_WATCHDOG 1305
convertParams 11
COOKIEDATEFORMAT 69
cookieStr 904
copyFile 47
copyFile2 47
COUNTER_HTTPCUPERTINO 1301
COUNTER_HTTPDVRCHUNKS 1301
COUNTER_HTTPMPEGDASH 1301
COUNTER_HTTPSANJOSE 1301
COUNTER_HTTPSMOOTH 1301
COUNTER_HTTPWEBM 1301
COUNTER_RTMP 1300
COUNTER_RTP 1300
COUNTER_TOTAL 1301
createApplication 1326
createBroadcastMessage 858
createConnectMessage 858
createContextDeserialize 128
createContextSerialize 128
createDefaultMessage 858
createDvrAudioChunk 461
createDvrDataChunk 462
createDvrOnMetadataChunk 462
createDvrVideoChunk 461
createEnhancedSeekMessage 859
createIndexLive 582
createInstance 1223, 1224, 1240, 1241
createPlayStatusMessage 858
createSeekMessage 858
createSOMessage 859
createStream 829, 948, 1225
CTRL_playlist_node 643

D

- data 393
- DATA_TYPE_AMF3 122
- DATA_TYPE_AMF3_ARRAY 124
- DATA_TYPE_AMF3_BOOLEAN_FALSE 123
- DATA_TYPE_AMF3_BOOLEAN_TRUE 123
- DATA_TYPE_AMF3_BYTEARRAY 124
- DATA_TYPE_AMF3_DATE 124
- DATA_TYPE_AMF3_INTEGER 123
- DATA_TYPE_AMF3_NULL 123
- DATA_TYPE_AMF3_NUMBER 123
- DATA_TYPE_AMF3_OBJECT 124
- DATA_TYPE_AMF3_STRING 123
- DATA_TYPE_AMF3_UNDEFINED 123
- DATA_TYPE_AMF3_XML_LEGACY 124
- DATA_TYPE_AMF3_XML_TOP 124
- DATA_TYPE_ARRAY 121
- DATA_TYPE_AS_OBJECT 122
- DATA_TYPE_BOOLEAN 120
- DATA_TYPE_BYTEARRAY 122
- DATA_TYPE_CUSTOM_CLASS 122
- DATA_TYPE_DATE 121
- DATA_TYPE_INTEGER 122
- DATA_TYPE_LONG_STRING 122
- DATA_TYPE_MIXED_ARRAY 121
- DATA_TYPE_MOVIE_CLIP 121
- DATA_TYPE_NULL 121
- DATA_TYPE_NUMBER 120
- DATA_TYPE_OBJECT 121
- DATA_TYPE_OBJECT_END 121
- DATA_TYPE_RECORDSET 122
- DATA_TYPE_REFERENCE_OBJECT 121
- DATA_TYPE_STRING 120
- DATA_TYPE_UNDEFINED 121
- DATA_TYPE_UNKNOWN 120
- DATA_TYPE_XML 122
- DATA_TYPE_XML_TOP 123
- DATE_AND_TIME_FORMAT 598
- DATE_AND_TIME_PATTERN 598
- DATEFORMAT 162
- dateValue 165
- debug 617, 618
- debugRTSPSession 907
- DebugUtils 39
- DECODE 15
- decode 18, 19
- DECODE_OBJ_REF 189
- DECODE_TRAITS 189
- DECODE_TRAITS_EXT 189
- DECODE_TRAITS_REF 189
- DECODE_UNDEFINED 189
- decodeFileToFile 21
- decodeFromFile 20
- decodeHexString 31
- decodeRangeHeader 1267
- decodeS 1001
- decodeSS 1001
- decodeStorageDir 277, 994
- decodeStreamInfo 1270
- decodeToFile 20
- decodeToObject 19
- decodeValue 103
- decompress 149
- decrement 321
- DEFAULT_APPINSTANCE_NAME 252
- DEFAULT_APPLICATION_NAME 231
- DEFAULT_CHUNK_MEMORY_CACHESIZE 428
- DEFAULT_CONFIGURATION_FILE 597
- DEFAULT_CONFIGURATION_KEY 597
- DEFAULT_CUPERTINO_PLAYLIST_GZIP_THRESHOLD 442
- DEFAULT_PORT 608
- DEFAULT_PROPERTY_ALLOWABLE_AV_PACKET_DELTA 424
- DEFAULT_PROPERTY_APPEND_DISCONTINUITY_DELTA 419
- DEFAULT_PROPERTY_AUDIO_ONLY_CHUNK_TARGET_DURATION 426
- DEFAULT_PROPERTY_BREAK_ONPTS 426
- DEFAULT_PROPERTY_CHUNK_CACHE_CLASS 422
- DEFAULT_PROPERTY_CHUNK_DURATION_MINIMUM 427
- DEFAULT_PROPERTY_CHUNK_GROUPING_SECONDS 418
- DEFAULT_PROPERTY_CHUNK_READER_CLASS 420
- DEFAULT_PROPERTY_CHUNK_WRITER_CLASS 421
- DEFAULT_PROPERTY_DEBUG_MAX_INVALID_CHUNKS_LOGGED 433
- DEFAULT_PROPERTY_DEBUG_MAX_RAW_PACKETS

- 435
- DEFAULT_PROPERTY_DEBUG_MAX_VALID_CHUNKS_LOGGED 434
- DEFAULT_PROPERTY_DEBUG_RAW_PACKETS 435
- DEFAULT_PROPERTY_DVR_MAX_CHUNK_LOG 436
- DEFAULT_PROPERTY_FILE_SYSTEM_CLASS 421
- DEFAULT_PROPERTY_MANIFEST_PERSISTENT_CLASS 422
- DEFAULT_PROPERTY_MAX_RECORDING_LENGTH 430
- DEFAULT_PROPERTY_MBR_MINIMUM_PACKET_TIME_GAP_SIZE 429
- DEFAULT_PROPERTY_MBR_MINIMUM_UTC_TIME_GAP_SIZE 429
- DEFAULT_PROPERTY_MEDIACACHE_READER_CLASS 430
- DEFAULT_PROPERTY_PACKET_DELTA_TO_NOTIFY 425
- DEFAULT_PROPERTY_PACKET_DELTA_TO_RESET_TIME 424
- DEFAULT_PROPERTY_PACKET_DURATION_MAXIMUM 429
- DEFAULT_PROPERTY_PACKET_SORT_TIME 426
- DEFAULT_PROPERTY_RECORDINGS_LOADER_CLASS 423
- DEFAULT_PROPERTY_REPEATER_HEARTBEAT_DURATION 427
- DEFAULT_PROPERTY_SANJOSE_ABST_DURATION_TOLERANCE 441
- DEFAULT_PROPERTY_SANJOSE_ABST_TIMESCALE 441
- DEFAULT_PROPERTY_SMOOTH_MANIFEST_LIVE_CAN_PAUSE 431
- DEFAULT_PROPERTY_SMOOTH_MANIFEST_LIVE_CAN_SEEK 430
- DEFAULT_PROPERTY_SMOOTH_MANIFEST_MAJOR_VERSION 432
- DEFAULT_PROPERTY_SMOOTH_MANIFEST_MINOR_VERSION 432
- DEFAULT_PROPERTY_SMOOTH_MANIFEST_RECORD_CAN_PAUSE 431
- DEFAULT_PROPERTY_SMOOTH_MANIFEST_RECORD_CAN_SEEK 431
- DEFAULT_PROPERTY_SMOOTH_MANIFEST_RECORD_SPECIFY_DURATION 432
- DEFAULT_PROPERTY_STORAGE_DIRECTORY 419
- DEFAULT_PROPERTY_TEXT_READER_CLASS 423
- DEFAULT_PROPERTY_TEXT_WRITER_CLASS 423
- DEFAULT_RANDOMACCESSREADER 1056
- DEFAULT_REPOSITORY_NAME 596
- DEFAULT_XML_CONFIGURATION_FILE 597
- DefaultDvrStreamVersionHandler 368
- defaultLocale 97
- defaultTimeZone 97
- deleteArchivedStore 485
- deleteDirectory 47
- deleteSlot 1012, 1036
- deleteStream 830
- describe 960, 961
- deserialize 130, 142, 148, 165, 177, 183, 194, 195, 385, 388, 390, 460
- deserializeBody 660, 680, 683, 686, 688, 690, 696, 701, 704, 709
- deserializeDate 113
- deserializeFrame 677
- deserializeInnerObject 128
- deserializeInt 113
- deserializeString 114, 677
- detachFromWMSSession 951
- determineChunkGroupIdentifier 402
- determineExistingStoreForPlaying 369, 500
- determineExistingStoreForRecording 369, 499
- determineStreamVersion 466
- determineUtcTime 405
- directOutput 1273
- disconnect 321, 1014, 1023, 1039
- doBreak 856
- doCRC32 36
- doesFileExist 531
- doesStackContainMethod 42
- doHTTPAuthentication 511
- doIdle 761, 925
- DONT_BREAK_LINES 16
- doSessionRedirect 569
- doSet 78
- doubleValue 164
- doWatchdog 718, 743, 750, 874, 995
- dummy 83
- DVR_DEFAULT_FILESTORE 418
- DVR_DEFAULT_RECORDER_ID 417
- DVR_REPEATER_PACKETIZER_ID 417

DVR_STREAMING_PACKETIZER_ID 417
DVR_TIME 443
DVR_WINDOW_DURATION_UNLIMITED 419
DvrManifestChunkEntry 371
DvrManifestCodecEntry 375
DvrManifestEntry 382
DvrManifestEntryFactory 385
DvrManifestEntryRangeGroup 389
DvrManifestOnMetadataEntry 393
DvrManifestTimeMapEntry 396
dvrRecorders 1168
dvrStart 381
dvrStop 381
dvrStreamStorageDeleted 477
dvrStreamStorageLoaded 477
dvrStreamStoreCreate 476
dvrStreamStoreDestroy 476
dvrStreamStoreInit 476
dvrToPt 505
dvrToUtc 505

E

elapsedTime 906
ElapsedTimer 44
ENCODE 15
encodeBytes 17, 18, 384
encodeFileToFile 20
encodeFromFile 20
encodeHexString 30
encodeObject 16
encodeToFile 19
encodeValue 103
ENCRYPTION_TYPE_CUPERTINO 445
ENCRYPTION_TYPE_PLAYREADY 445
encryptions 371
end 100
endChunk 464, 573
equals 95
error 616
EVT_ALL 649
EVT_announce 642
EVT_app_start 641
EVT_app_stop 641
EVT_comment 642

EVT_connect 639
EVT_connect_burst 639
EVT_connect_pending 639
EVT_create 640
EVT_decoderaudiostart 642
EVT_decoderaudiostop 642
EVT_decodervideostart 642
EVT_decodervideostop 642
EVT_describe 642
EVT_destroy 640
EVT_disconnect 639
EVT_encoderaudiostart 642
EVT_encoderaudiostop 642
EVT_encodervideostart 643
EVT_encodervideostop 643
EVT_pause 640
EVT_play 639
EVT_publish 639
EVT_record 641
EVT_recordstop 641
EVT_seek 640
EVT_server_start 641
EVT_server_stop 641
EVT_setbuffertime 640
EVT_setstreamtype 640
EVT_stop 640
EVT_unpause 640
EVT_unpublish 639
EVT_vhost_start 641
EVT_vhost_stop 641
execute 1339
exists 8, 1022
existsRTSPTunnelingSession 878
expandEndTime 401, 459
expandEnvironmentVariables 97, 98
expandStartTime 401, 459
expandToPortPair 877
extractCodecConfigFromTrackInfo 952
extractHTTPRequestInfo 567

F

fastDateFormat 162
FastPlaySettings 1044
fatal 617

FCPublish 835
FCSubscribe 836
fcSubscribe 357
FCUnPublish 835
FCUnpublish 836
FCUnSubscribe 837
FCUnsubscribe 837
fcUnSubscribe 357
fcUnSubscribeAll 357
FD_ALL 649
FD_c_client_id 634
FD_c_ip 634
FD_c_proto 634
FD_c_referrer 634
FD_c_user_agent 634
FD_cs_bytes 634
FD_cs_bytes_inc 604
FD_cs_stream_bytes 635
FD_cs_stream_bytes_inc 604
FD_cs_uri_query 636
FD_cs_uri_stem 636
FD_date 633
FD_s_ip 635
FD_s_port 637
FD_s_uri 634
FD_sc_bytes 635
FD_sc_bytes_inc 604
FD_sc_stream_bytes 635
FD_sc_stream_bytes_inc 604
FD_time 633
FD_tz 633
FD_x_app 633
FD_x_appinst 634
FD_x_category 633
FD_x_comment 637
FD_x_ctx 633
FD_x_ctx_override 633
FD_x_duration 636
FD_x_duration_inc 604
FD_x_event 632
FD_x_file_ext 636
FD_x_file_length 635
FD_x_file_name 636
FD_x_file_size 635
FD_x_severity 637
FD_x_sname 635
FD_x_sname_query 636
FD_x_spos 635
FD_x_status 636
FD_x_stream_id 637
FD_x_suri 637
FD_x_suri_query 636
FD_x_suri_stem 636
FD_x_vhost 633
FILEEXTENSION 1008
FILEFORMAT_FLV 1305
FILEFORMAT_MP4 1305
FILEFORMAT_UNKNOWN 1305
fileToByteArray 47
FileUtils 46
filters 509
FLAGS_DEFAULT 676
floatValue 164
flush 1012, 1023, 1040, 1111, 1228
flushBase64 27
FLV_CHUNKHEADER_BUFFERSIZE 54
FLV_CHUNKHEADER_FIRSTBYTE 53
FLV_CHUNKHEADER_HEADERSIZE 54
FLV_CHUNKHEADER_ISIZE 53
FLV_CHUNKHEADER_ITIMECODE 53
FLV_CHUNKHEADER_ITYPE 53
FLV_CHUNKHEADER_SECONDBYTE 53
FLV_CHUNKHEADER_VALUESIZE 54
FLV_DFRAME 54
FLV_KFRAME 54
FLV_PFRAME 54
FLV_TCINDEXAUDIO 54
FLV_TCINDEXDATA 55
FLV_TCINDEXVIDEO 55
FLV_UFRAME 54
FLVUtils 55
forceNewLoggerInstance 621
forceReset 718
formatBytes 40
formatBytesShort 39
formatBytesStruct 41
formatDeleteCookie 71
formatH264CodecConfig 1270
formatH264CodecConfigPacket 1270
formatMilliseconds 42

formatRTPInfo 962
formatSeconds 42
formatSetCookie 72
formatUtcTime 42
FORWARD 6
frameTypeToString 56
fS 1002

G

generateAuth 85, 86, 87
generateAuthHTTP 86
generateHash 88
generateHashBytes 87, 88
get 139, 175, 192, 223, 224, 1021, 1284
getAbsTimecode 206, 216
getAccess 1041, 1109
getActiveCount 1339
getAdapterName 540
getAddress 1279
getAddressCount 892
getAddressRawStr 1279
getAddressStr 1279
getAdminAgent 993
getAdminGUID 999
getAdminInterfaceObjectList 980, 997
getAllAsStrings 302
getAllowDomains 267
getAndClearPendingFrameGrabs 1256
getAndClearPendingOverlays 1259
getAppInstance 232, 334, 492, 543, 557, 716, 818, 912, 949, 1173, 1182, 1203, 1225
getAppInstanceName 953, 1173
getAppInstanceNames 234
getAppInstanceProperty 852
getApplication 252, 334, 609, 818, 1310
getApplicationFolderNames 1320
getApplicationInstance 750, 1195
getApplicationInstanceTouchTimeout 285
getApplicationLock 1317
getApplicationNames 1320
getApplicationPath 232
getApplicationProperty 851
getApplicationTimeout 264, 1315
getAppName 952, 1173
getArchiveStrategy 369, 483, 500
getArtifactsTextRepresentation 373, 394
getAudioCodec 57, 65, 375
getAudioCodecConfigPacket 1110
getAudioCodecId 780
getAudioHost 883
getAudioMissing 1094
getAudioMP3Layer 64
getAudioPacketizerItem 874, 922
getAudioPort 882
getAudioSize 1079
getAudioTC 1082
getAudioTrack 946
getAuthenticatePlayHandler 911
getAuthenticatePublishHandler 911
getAuthenticationList 1321
getAuthenticationMethod 511, 517
getAutoAllocateInterleavePorts 964
getAVSyncMethod 956
getBaseClass 733, 1333
getBasePath 8
getBitrateAudio 779
getBitrateTotal 779
getBitrateVideo 779
getBodySize 660, 681, 684, 686, 688, 690, 696, 702, 704, 709
getBoolean 141, 177, 193, 194, 226, 228
getBufferSize 691
getBufferTime 344, 1079
getBurstStartStop 1106
getByte 140, 177, 193, 194, 226, 227
getByteAllocation 73
getByteContainerLevel 207
getCacheName 1104
getCallbackParamCount 814
getChecksum 1048
getChunkCounter 209
getChunker 493
getChunkGroupDuration 402
getChunkIndex 396
getChunks 204
getClassName 196, 199
getClient 255, 307, 314, 1085, 1309
getClientById 255
getClientCount 255, 1312
getClientCountTotal 254

getClientID 832
getClientId 333, 1085
getClientIdGenerator 982, 998
getClientIdGeneratorRecycleDelaySize 1001
getClientIdGeneratorRecycleSize 1001
getClientIdGeneratorTimeout 1001
getClientIdleFrequency 267, 1318
getClientProperty 852
getClients 255, 1010, 1035
getClientsLockObj 275
getClientTimeout 1308
getClientUpdates 1038
getClosestStartTime 400, 458, 498
getCodecEntryForTime 453
getCodecId 780
getCommandInterface 997
getCommandInterfaceCommandHandler 984, 1001
getCommandInterfaceHostPort 979, 996
getCommittedVirtualMemory 1000
getCommonInitialTextRepString 384
getConfigDir 1341
getConfigPath 232
getConfiguration 1280
getConnectionCounter 233, 260, 979, 995, 996, 1314
getConnectionHolder 557, 914
getConnectionLimit 1315, 1342
getConnectionTimeout 733
getConnectionValidator 996
getConnectLastAttempt 720
getConnectLastForceReset 720
getConnectLastSuccess 720
getConnectTime 342
getContentDescriptors 697
getContentLength 520
getContentType 521, 696
getContextStr 284, 480, 492, 1115, 1204
getCookieStr 562, 921
getCoreHandlerPoolSize 981, 998, 1311
getCoreTransportPoolSize 980, 997, 1310
getCount 1172
getCpuTime 98
getCryptoPoolActiveCount 1000
getCryptoPoolMaxSize 999
getCurrent 321
getCurrentHeapSize 1000
getCurrentItem 1242
getCurrentTime 468
getData 218, 393, 684, 1026
getDataBuffer 217
getDataMissing 1094
getDataSize 1080
getDataTC 1083
getDataType 1084
getDate 44, 141, 177, 193, 194, 226, 228
getDateStarted 234, 260, 343, 979, 996, 1314
getDateString 45
getDefaultStreamingStore 482
getDePacketizerList 877
getDePacketizerName 878
getDescription 661, 681, 684, 733, 1335
getDirection 1045
getDirecton 7
getDouble 141, 176, 193, 194, 225, 227
getDuration 383, 949, 1061, 1157, 1233, 1236
getDvrApplicationContext 292
getDvrArtifact 372, 393
getDvrChunkAtTime 497
getDvrChunkByIndex 497
getDvrChunkNearTime 497
getDvrFileSystemClass 482
getDvrManager 492, 1183
getDvrPlaylistRequest 466
getDvrProperties 292
getDvrRecorder 1112, 1116, 1176
getDvrRecorderId 1183
getDvrRecorderList 293, 1112, 1327
getDvrRecorders 1177
getDvrRepeater 1116
getDvrSessionInfo 567
getDvrStorageDir 482
getDvrStorageWindowSeconds 482, 491
getDvrStoreList 1327
getDvrTime 457
getDynamicLogProperties 982, 998, 1322
getElapsedTime 356, 565, 923, 1113
getEmbeddedFlag 691
getEncoding 609
getEncryptionDelegate 483
getEncryptionRepeaterSharedSecret 483
getEncryptions 372

getEncryptionsTextRepresentation 373
getEncryptionType 446
getEndIndex 387
getEntriesToPurge 453
getExecutor 1339
getExt 1107
getExtent 1047
getFastPlaySettings 341, 1098
getFileExtension 1224
getFileInBytes 81
getFileInBytesRate 83
getFileInfo 564
getFileIOPoolSize 1311
getFileOutBytes 82
getFileOutBytesRate 83
getFilePointer 7
getFileSystem 493
getFirstByte 218
getFirstEntry 401, 455
getFirstIndex 401
getFirstRendition 785
getFirstSegment 772
getFlags 678, 709
getFlashVer 334
getFloat 141, 176, 193, 194, 225, 228
getFps 1044
getFrameCount 1263
getFrameIdentifier 684
getFrameMapIds 654
getFrames 654
getFrameType 57
getGlobalLogValue 623
getGUID 999
getHandler 889
getHandlerThreadPool 981, 998, 1312
getHeader 519, 526
getHeaderBytes 523
getHeaderMap 519
getHeaderNames 520
getHeaders 526
getHeaderSize 527, 1109
getHeight 781
getHighestVersionedStore 481
getHomePath 1310
getHost 882, 947
getHostPortsList 1311
getHostType 884
getHTTPHeader 567
getHTTPHeaderMap 567
getHTTPHeaderNames 567
getHTTPIntHeader 567
getHttpProviders 1281
getHTTPStreamerAdapter 557, 1325
getHTTPStreamerAdapterIDs 1281, 1325
getHTTPStreamerApplicationContext 281
getHTTPStreamerContext 1321
getHTTPStreamerItem 539
getHTTPStreamerList 277, 1325
getHTTPStreamerProperties 276
getHTTPStreamerSession 1053, 1113
getHTTPStreamerSessionCount 280, 281
getHTTPStreamerSessionCountsByName 279
getHTTPStreamerSessions 279, 280
getID 540
getId 204
getIdleCheckFrequency 1318
getIdleFrequency 347, 540, 919
getIdleHandler 915
getIdleMinimumWaitTime 1322
getIdleTimeout 717
getIdleWorkerCount 1319
getIdleWorkers 1322
getIdStr 677, 708
getIdString 734
getIndex 383, 532, 583, 588, 1217
getIndexLive 583
getIndexMap 399
getInIpAddress 888
getInitialRepeaterItems 488
getInnerObj 199
getInPort 889
getInputStream 522
getInstance 622, 993, 1348
getInt 140, 176, 193, 194, 224, 227
getIntData 151
getIntHeader 520, 526
getIOPerformanceCounter 262, 564, 921
getIoPerformanceCounter 234, 978, 995, 1314
getIOScheduler 1313
getIoSession 924

getIp 341, 912
getIpAddress 558
getItems 1213
getJmxRemoteConfig 997
getKeepAliveTime 734
getKeepAliveTimeout 1319
getKey 192, 223
getKeys 192, 223
getLanguage 696
getLastAccessed 1048
getLastAudioTimecode 1228
getLastAuthenticateMethod 911
getLastClientId 1028
getLastConnectAcceptedStamp 322
getLastConnectAcceptedStampString 322
getLastConnectAcceptedTime 322
getLastConnectAcceptedTimeString 322
getLastConnectRejectedByReasonStamp 323
getLastConnectRejectedByReasonStampString 323
getLastConnectRejectedByReasonTime 323
getLastConnectRejectedByReasonTimeString 323
getLastConnectRejectedStamp 322
getLastConnectRejectedStampString 322
getLastConnectRejectedTime 322
getLastConnectRejectedTimeString 323
getLastDataTimecode 1229
getLastDisconnectStamp 324
getLastDisconnectStampString 324
getLastDisconnectTime 324
getLastDisconnectTimeString 324
getLastKeyFrame 1095
getLastLiveEntry 401, 455
getLastOperation 1028
getLastPacket 1096
getLastPacketsByType 960
getLastPurgeTime 468
getLastRecordedEntry 401, 452
getLastRecordedIndex 398, 457
getLastRequest 571
getLastStreamId 830
getLastTC 64
getLastTouchTime 285
getLastValidateTime 352
getLastVideoTimecode 1228
getLength 1061, 1217
getLicenseCounter 1001
getLicenseHolder 1186, 1203
getLicenseTracker 1001
getLicenseValidator 996
getLiveDuration 400, 457
getLiveEntries 400, 455
getLiveEntriesWithLimit 400, 456
getLivePlaylist 589
getLiveRangeEndingBeforeTime 401
getLiveRepeaterCapabilities 356
getLiveStreamDvrRecorderControl 292
getLiveStreamingPacketizer 558
getLiveStreamPacketizer 744, 833, 1112, 1113, 1175
getLiveStreamPacketizerControl 282
getLiveStreamPacketizerId 1193
getLiveStreamPacketizerList 277, 355, 1111, 1325
getLiveStreamPacketizerLock 1175
getLiveStreamPacketizerProperties 275
getLiveStreamRepeater 745, 1114
getLiveStreamTranscoder 1117
getLiveStreamTranscoderControl 292
getLiveStreamTranscoderItem 1203
getLiveStreamTranscoderList 291, 356, 1112, 1325
getLiveStreamTranscoders 1117
getLiveTailEntries 400
getLiveThreads 1000
getLocale 523
getLock 555, 653, 741, 750, 773, 778, 786, 876, 1241
getLockCount 743, 757
getLogger 613, 623, 820
getLoggerObj 621, 623
getLong 140, 176, 193, 194, 225, 227
getMACAddress 98
getManifest 492
getManifestChannel 458
getManifestRepresentation 372, 375, 383, 394, 396
getMapIdStr 660, 678, 708
getMaxHeapSize 1000
getMaximumPendingReadBytes 266, 1317
getMaximumPendingWriteBytes 265, 355, 1316
getMaximumSetBufferTime 266, 355, 1317
getMaxRTCPWaitTime 956
getMaxStorageDirDepth 276
getMaxTimecode 1095, 1228
getMediaCaster 742, 751, 1128

getMediaCasterCount 750
getMediaCasterDef 716, 737
getMediaCasterDefs 736
getMediaCasterHostPortConfig 738
getMediaCasterId 716, 742
getMediaCasterList 1320
getMediaCasterNames 736, 750
getMediaCasterProcessorCount 739
getMediaCasterProperties 275
getMediaCasterRTPTSPRTPTransportMode 295
getMediaCasterSettings 1320
getMediaCasterStreamItem 718
getMediaCasterStreams 263
getMediaCasterType 721, 948
getMediaCasterValidator 285
getMediaExtension 8, 1053, 1062
getMediaIOPerformance 1099
getMediaIOPerformanceCounter 334
getMediaList 786, 1053
getMediaListProvider 295
getMediaListSegment 778
getMediaName 8
getMediaReader 964
getMediaReaderContentType 295, 1327
getMediaReaderProperties 275
getMediaReaders 1315
getMediaWriterProperties 275
getMediaWriters 1315
getMember 199
getMemberCount 198
getMembers 198
getMessagesInBytes 81
getMessagesInBytesRate 82
getMessagesInCount 81
getMessagesInCountRate 82
getMessagesLossBytes 82
getMessagesLossBytesRate 83
getMessagesLossCount 81
getMessagesLossCountRate 82
getMessagesOutBytes 81
getMessagesOutBytesRate 83
getMessagesOutCount 81
getMessagesOutCountRate 82
getMetadata 1059
getMetadataEntryForTime 453
getMetadataPacket 866, 951
getMetaDataProvider 1109
getMethod 520
getMimeType 661
getMinimumAvailableChunks 481
getMissing 214
getMode 961
getModFunctions 263
getModuleInstance 264
getModuleList 264
getMp3TagMap 1315
getMPEGTSAudioBitrate 971
getMPEGTSAudioLanguage 957
getMPEGTSAudioPID 958
getMPEGTSProgramID 957
getMPEGTSVideoBitrate 971
getMPEGTSVideoPID 957
getMsgBytes 524
getMultiplier 1044
getName 232, 253, 732, 773, 780, 885, 1014, 1027, 1039, 1081,
1215, 1217, 1243, 1312, 1333, 1341
getNameGroupByGroupName 1176
getNameGroups 1176
getNameGroupStreamNames 1176
getNetConnection 723, 1085
getNetConnectionHostPortConfig 1320
getNetConnectionIdleFrequency 1318
getNetConnectionProcessorCount 1312
getNextChunkIndex 457
getNextCodecIndex 458
getNextMetadataIndex 458
getNextNameGroupId 1176
getNextNetConnectionId 1308
getNextSessionId 927
getNextStreamIndex 1171, 1172
getNextTimeMapIndex 458
getNodeByTagName 106
getNodeValue 107
getNodeValueByTagName 106
getNormalizedNTPTimecode 961
getNumberLiveEntries 400, 454
getNumberRecordedEntries 400, 454
getObject 141, 153, 177, 193, 194, 226, 228
getObjectEncoding 152, 156, 208, 354
getObjectNames 1021

getObjectReference 157
getOffsetToNextTag 691
getOnMetadataData 67
getOrCreate 1021
getOutHost 959
getOutIpAddress 889
getOutPort 889
getOutputStream 525
getPacketizerName 481
getPacketStartTime 382
getPageUrl 346, 836
getParam 814
getParamBoolean 817, 818
getParamCount 814
getParamDate 816
getParamDouble 816, 817
getParameter 521
getParameterMap 522
getParameterNames 521
getParameterValues 522
getParamInt 816
getParamLong 817
getParamMixedArray 815
getParamObj 815
getParamString 815
getParamType 814
getParamValue 104
getParent 1017, 1041, 1143
getPassword 313
getPath 9, 510, 523, 1052, 1061
getPeakThreads 1000
getPictureData 661
getPictureType 661
getPingRoundTripTime 343
getPingTimeout 265, 352, 1316
getPlayClass 1334
getPlayDuration 568, 781
getPlayer 1094
getPlayerCount 743, 757
getPlaylist 1241
getPlaylistAlignment 466
getPlayPackets 1095
getPlayStart 568, 781
getPlayStreamCount 263
getPlayStreamCountsByName 263
getPlayStreams 335
getPlayStreamsByName 263
getPollingInterval 1244
getPort 610, 1279
getProcessorCount 1280
getProperties 233, 254, 336, 491, 539, 543, 566, 733, 773, 779, 786, 913, 982, 998, 1095, 1183, 1192, 1203, 1310, 1334, 1342
getProperty 301, 732, 1013, 1036, 1308, 1334
getPropertyBoolean 301
getPropertyDouble 302
getPropertyInt 302
getPropertyLong 302
getPropertyStr 301
getProtocol 347, 522
getProtocolUsage 236, 296, 1002, 1327
getProtocolUsageTracker 1002
getPublishAudioCodecId 1114
getPublisher 1245
getPublisherCount 279
getPublishers 278
getPublishStreamNames 284, 1169
getPublishStreams 335
getPublishVideoCodecId 1114
getPurgeController 493
getQueryStr 345, 561, 916, 1107
getQueryString 520
getQueueSize 1339
getRange 387
getReceiveVideoFPS 1099
getReconnectWaitTime 720
getRecordedDuration 399, 457
getRecordedEntries 398, 399, 451, 456
getRecordedEntriesInRange 399, 452
getRecordedEntriesMap 452
getRecordedEntriesWithLimit 399, 456, 497
getRecordedEntryByIndex 399, 452, 497
getRecordedEntryByTimeKey 399, 452
getRecordedEntryStartingBeforeTime 399
getRecordingName 1186
getRecordingStreamStore 482
getRedirectSessionBody 570
getRedirectSessionCode 569, 924
getRedirectSessionContentType 570
getRedirectSessionMessage 924
getRedirectSessionURL 569, 924

getRefCount 1011, 1035
getReference 127
getReferrer 346, 560, 836, 916
getRemoteAddr 523
getRemoteHost 523, 609
getRendition 785
getRenditions 785
getRepeat 1214, 1244
getRepeaterItemsDvr 488
getRepeaterLastSequence 1193
getRepeaterMediaCasterDef 543
getRepeaterOriginUrl 266, 352, 835
getRepeaterQueryString 267
getRequestFilters 510, 517
getRequestURI 520
getRequestURL 521
getRespAMFAudioObj 345, 1106
getRespAMFDataObj 345, 1106
getRespAMFVideoObj 345, 1106
getRespFunctions 342
getResponseAMFObj 345
getRsoStorageDir 269
getRsoStoragePath 269
getRTPAVSyncMethod 272
getRTPContext 943, 1321
getRTPDatagramConfigIncoming 1321
getRTPDatagramConfigOutgoing 1321
getRTPDestination 969
getRTPIIdleFrequency 268, 1318
getRTPIInfo 962
getRTIPMaxRTCPWaitTime 273
getRTIPPlayAuthenticationMethod 272
getRTIPProperties 275
getRTIPublishAuthenticationMethod 272
getRTPSession 306, 314, 894
getRTPSessionCount 274
getRTPSessionCountsByName 274
getRTPSessions 273
getRTPStream 354, 1110
getRTIPWriteListener 914
getRTSPBindIpAddress 287, 966
getRTSPConnectionAddressType 288, 966
getRTSPConnectionIpAddress 288, 966
getRTSPMaximumPendingWriteBytes 286, 965
getRTSPOriginAddressType 289, 967
getRTSPOriginIpAddress 288, 966
getRTSPPlayRangeStart 920
getRTSPPlayRangeStop 920
getRTSPSessionDescription 963
getRTSPSessionExtraLines 969
getRTSPSessionName 963
getRTSPSessionTimeout 286, 965
getRTSPStream 909
getRTSPTunnelingLock 878
getRTSPTunnelingSession 878
getRTSPTunnelingSessionId 920
getScheme 522
getSDPData 893
getSDPLang 969
getSecondByte 218
getSegment 772, 773
getSeq 217
getSerializeSize 445
getServerGUID 999
getServerHostPort 347
getServerIp 559, 915
getServerName 522
getServerPort 523, 560, 915
getSession 927, 946
getSessionDescriptionData 760
getSessionGUID 999
getSessionId 554, 908
getSessionIds 927
getSessionProtocol 559
getSessions 875
getSessionTimeout 556
getSessionTracker 593
getSessionType 558
getSharedObjectReadAccess 271, 348
getSharedObjects 255, 256
getSharedObjectWriteAccess 271, 348
getShort 140, 176, 193, 194, 225, 227
getSize 204, 214, 654, 678, 709, 781
getSlot 1010, 1034
getSlotNames 1010, 1034
getSlots 1016, 1034
getSlotVersion 1027
getSoVersion 1027
getSrc 205, 215, 1080
getSSLConfig 1281

getSslFactoryClass 1279
getStart 1217
getStartIndex 387
getStartLiveOnPreviousBufferTime 1246
getStartOnPreviousBufferTime 1234
getStartStream 1195
getStartTC 1045
getStartTCOffset 1045
getStartTimecode 382
getStartupStreams 1326
getStopTimecode 382
getStorageDir 1015, 1023, 1040
getStream 555, 716, 820, 946, 1169, 1170, 1182, 1224
getStreamArray 746
getStreamAttributes 957
getStreamAudioSampleAccess 270, 349
getStreamBaseName 481
getStreamBitrate 834, 1273
getStreamClientless 1169
getStreamCount 263
getStreamDef 1337
getStreamDefs 1336
getStreamExt 562, 744, 953
getStreamFile 343, 344
getStreamFileForRead 1105
getStreamFileForWrite 1105
getStreamFileMapper 276
getStreamId 946
getStreamInfo 947, 956
getStreamIsRunningLock 720
getStreamKeyDir 268
getStreamKeyPath 269
getStreamLastSeq 719
getStreamLength 834, 1273
getStreamList 747
getStreamListLock 1168
getStreamLock 943
getStreamManager 750
getStreamMissingTime 718
getStreamName 492, 562, 953, 1202
getStreamNameAliasProvider 278
getStreamNameLock 1169
getStreamNameLogging 954
getStreamNameParts 563
getStreamNames 757
getStreamPacketizerItem 875, 922
getStreamPort 883
getStreamPosition 563, 1062
getStreamProperties 275
getStreamProperty 853
getStreamQueryStr 959
getStreamReadAccess 270, 350
getStreams 253, 1086, 1169
getStreamSessionId 954
getStreamSessionIp 954
getStreamSessionVersion 954
getStreamStorageDir 268
getStreamStoragePath 269
getStreamStore 481
getStreamStores 481
getStreamTimeout 718
getStreamTimeoutLastReset 719
getStreamTimeoutLastTime 719
getStreamTimeoutReason 719
getStreamTrack 946
getStreamType 254, 336, 731, 833, 952, 1096, 1224
getStreamTypeNames 1337
getStreamTypes 1308
getStreamTypeStr 543
getStreamVersionHandler 483
getStreamVideoSampleAccess 269, 349
getStreamWriteAccess 271, 351
getString 140, 153, 175, 192, 193, 224, 226
getStringReference 157
getSystemTime 99
getTargetEncoding 156
getTextEncoding 660, 681, 696, 702, 705
getThreadPool 981, 998, 1311
getTime 45
getTimecode 205, 216, 698, 859
getTimeMap 460, 504
getTimeMapEntries 504
getTimeMapper 492
getTimeMapping 396
getTimeOffsetBetweenItems 1247
getTimeRunning 234, 261, 343, 566, 923, 979, 996, 1314
getTimeRunningSeconds 234, 261, 343, 566, 923, 979, 996, 1315
getTimeSeconds 45
getTimeStampFormat 696
getTimeString 45

getTotal 324
getTotalAccepted 324
getTotalIOPerformanceCounter 334
getTotalRejected 325
getTrack 888, 945
getTrackId 945
getTrackNames 945
getTrait 153, 196
getTraitReference 157
getTranscoderApplicationContext 292
getTranscoderName 1202
getTranscoderProperties 276
getTransportMode 961
getThreadPool 981, 998, 1312
getTTL 884, 891
getType 125, 175, 205, 215, 383, 386, 398, 780, 860
getTypeAsString 778
getUDPMangedDeliveryCount 970
getUDPMangedDeliveryDelay 970
getUDPPortManager 983, 999
getUDPPortSharingManager 983, 999
getUDPTransport 888, 955
getUDPTransportManager 876, 1324
getUniqueId 742
getUniqueStreamIdStr 1113
getUnsignedShort 36
getUri 346, 560, 916
getURL 684, 704
getUserAgent 560, 910
getUserAgents 980, 997
getUserHTTPHeader 568
getUserQueryStr 566
getUserTime 99
getUtcStartTime 382
getValidationFrequency 265, 1316
getValue 130, 142, 147, 165, 179, 195, 681, 699, 702
getVersion 832, 977, 994, 1012, 1036
getVHost 233, 253, 306, 314, 339, 540, 543, 554, 715, 819, 876, 910, 928, 947, 1173
getVHostItems 1346
getVHostList 982, 999
getVHostMap 1345
getVHostNames 1346, 1349
getVideoCodec 57, 65, 375
getVideoCodecConfigPacket 1110
getVideoCodecId 779
getVideoFrameType 65
getVideoHost 883
getVideoMissing 1094
getVideoPacketizerItem 875, 922
getVideoPort 882
getVideoSize 1080
getVideoTC 1083
getVideoTimecodeOffset 65
getVideoTrack 946
getVODLastTimeTC 967
getVODPlayLen 968
getVODStartTimeTC 967
getWidth 781
getWriteListener 351
getXMLPropertyBool 109
getXMLPropertyDouble 109
getXMLPropertyExists 108
getXMLPropertyInt 108
getXMLPropertyIntSize 108
getXMLPropertyLong 108
getXMLPropertyLongSize 109
getXMLPropertyStr 107
gmtTimeZone 97
grabFrame 1256
GZIP 16

H

handleArchivedStream 369, 500
handleCallback 1100
handleMessage 865, 891
handleOnMetadata 1201
handlePacket 1182, 1193, 1201
hasAudio 459, 494
hasCodecData 459
hasData 459, 494
hasEncryption 494
hasOnMetadata 459, 494
hasTimeMapData 460
hasVideo 459, 494
HEADERSIZE 668
height 777
hexadecimal 30
HOSTNAME_KEY 597

- HostPort 1278
- HostPortList 1283
- HttpRequestToByteArray 69, 70
- HttpRequestToFile 69
- HTTPProvider2Base 509
- HTTPStreamerSessionCupertino 531
- HTTPStreamerSessionMPEGDash 578
- HTTPStreamerSessionSanJose 582
- HTTPStreamerSessionSmoothStreamer 588
- HTTPStreamerSessionWebM 593
- HTTPUtils 69
- I
- ID3FOOTER_SIZE 652
- ID3Frames 653
- ID3HEADER_SIZE 652
- ID3HEADER_VERSION 652
- ID3HEADERFLAGS_DEFAULT 652
- ID3HEADERFLAGS_EXPERIMENTAL 653
- ID3HEADERFLAGS_EXTENDED 653
- ID3HEADERFLAGS_FOOTERPRESENT 653
- ID3HEADERFLAGS_UNSYNC 653
- ID3V2FrameAttachedPicture 660
- ID3V2FrameBase 676
- ID3V2FrameComment 680
- ID3V2FrameLinkedInformation 683
- ID3V2FramePrivate 686
- ID3V2FrameRawBytes 688
- ID3V2FrameRecommendedBufferSize 690
- ID3V2FrameSynchronizedText 695
- ID3V2FrameSynchronizedTextDescriptor 698
- ID3V2FrameTextInformation 701
- ID3V2FrameURLLink 704
- ID3V2Utils 706
- idle 960, 1085
- idleFrequency 907
- idleHandler 906
- idStringToName 734
- importManifest 458
- incAbsTimecode 206
- incByteContainerLevel 207
- incClientCountTotal 255
- incrementAccept 320
- incrementBytesIn 78
- incrementBytesLoss 80
- incrementBytesOut 80
- incrementFileIn 78
- incrementFileOut 79
- incrementMediaInBytes 944, 1100
- incrementMediaLossBytes 1100
- incrementMediaOutBytes 1099
- incrementMessagesIn 78, 79
- incrementMessagesLoss 79, 80, 81
- incrementMessagesOut 79, 80
- incrementReject 320
- incSlotVersion 1027
- index 381
- indexOf 37
- indexOfDifferent 37
- info 615, 616
- init 6, 308, 309, 403, 405, 467, 472, 491, 502, 503, 510, 517, 540, 542, 715, 742, 760, 874, 1026, 1051, 1058, 1078, 1139, 1162, 1181, 1192, 1201, 1308, 1339, 1348
- initContextLogging 622
- initialize 451
- initializeLogging 622
- initialManifest 488
- initialManifestEnd 488
- initLiveStreamRepeating 833, 1114, 1143
- initStream 830
- InputStream 23
- interleavePackets 60, 61
- interruptPlay 1142
- intToByteArray 34, 35
- intToByteArraySafeSync 707
- intToHexStr 94
- intToStr 94
- intValue 164
- invoke 973
- invokePrevious 820, 821
- IOPerformanceCounter 77
- ioPerformanceCounter 905
- ioSession 908
- ip 904
- isAcceptConnection 254, 337
- isAcceptSession 562
- isActive 555, 609, 1183, 1192
- isAddressMulticast 91
- isAMF0 152, 157

isAMF3 152, 157
isAMF3Start 126
isAnnounce 905, 913
isAnnounceOrDescribe 914
isAppend 1093
isAppInstanceLoaded 232
isApplicationLoaded 1310
isArrayStart 126
isAudio 217
isAudioCodecConfig 64
isAVSyncNonSR 956
isBlockUDPOut 892, 970
isByteArrayStart 127
isByteContainerEmpty 207
isByteContainerFull 207
isCheckIpAddr 955
isCheckSSRC 955
isClient 1010, 1035
isClustered 1104
isConnected 336, 906, 914
isDebugEnabled 614
isDebugLog 83
isDebugRTSPSession 919
isDescribe 905, 913
isDynamic 199
isDynamicLogContextLoaded 982, 999
isEmpty 94, 388, 390, 398, 653
isEnabledFor 614
isEncoderInUse 1263
isEncrypted 347
isErrorEnabled 614
isFileInfo 564
isFirstABSTRequest 583
isFlashMediaLiveEncoder 353
isFlashVersion10 353
isFlashVersion90115 353
isFlashVersionH264Capable 353
isForceAMF0 860
isForceMPEGTSOut 970
isForceRTSPInterleaved 962
isGlobalLogValueSet 623
isHitEnd 1049
isInfoEnabled 614
isInMulticast 891
isInRange 387, 390
isIntData 151
isLastSentAbsTimecode 208
isLive 496, 964
isLiveRepeater 352
isLiveStreamPacketize 1198
isLiveStreamTranscode 1205
isLoaded 493
isLoggedConnect 912
isLongTimecode 208
isMediaCasterPlay 1115
isMember 198
isMergeOnMetadata 1115
isMixedArrayStart 127
isModePlay 959
isModePublish 959
isModeUnknown 959
isMPEGTSOut 963
isMulticast 881, 891
isNew 206
ISO8601_FORMAT 599
ISO8601_PATTERN 599
isObjectEncodingAMF0 208, 354
isObjectEncodingAMF3 208, 353
isObjEnd 127
isObjStart 126
isOnMetadataPacket 66
isOpen 7, 1052, 1059, 1097
isOutMulticast 891
isOverlayAvailable 1258
isPacketizeAudio 574, 1194
isPacketizeData 574, 1195
isPacketizeVideo 574, 1194
isPaused 949
isPending 863
isPersistent 1011, 1022, 1035
isPlay 1084
isPlaying 1079
isPlaylistReady 465, 466
isPlayLogged 556
isPublishStreamReady 948, 1115
isPurgingEnabled 467
isReceiveAudio 1098
isReceiveVideo 1099
isRecord 1084
isRecording 484, 496, 1185

isRecordingPaused 484, 496, 1185
isRedirectSession 569, 923
isRepeaterEdge 1194
isResetPlayStream 949
isResyncAudioVideoOnSR 951
isRTPIgnoreProfileLevelId 970
isRTPIncomingDatagramPortValid 282
isRTPWrapped 884
isRTSP 958
isRTSPAlwaysUseSDPPorts 968
isRTSPPull 958
isRunning 1263
isSecure 347, 523
isSendOnMetadata 1211, 1231, 1246
isSendPlayStopLogEvent 1108
isSendPublishStopLogEvent 1109
isSendRecordStopLogEvent 1108
isSendResult 818
isSendSDESEvents 963
isSession 719
isSessionValid 904, 911
isShutdownClient 921
isShutdownOnRelease 744
isShuttingDown 1322
isSSL 347
isStartLiveOnPreviousKeyFrame 1246
isStartOnPreviousKeyFrame 1234
isStartStartupStreams 1326
isStream 719, 882
isStreamDomainProtectionActive 544
isStreamIsRunning 720
isStreamStarted 968
isSuspended 983, 994, 1281, 1323
isSwitchLog 1245
isTemplateLoaded 1204
isTimeout 554, 965
isTimeoutSession 555
isTimesInMilliseconds 1246
isTraceEnabled 614
isTranscoderActive 1204
isTranscodeResult 1117
isValid 742, 1065
isValidated 556
isValidateFMLEConnections 290, 357
isValidAudioCodec 574

isValidStreamDomainStr 564
isValidVideoCodec 574
isVersionFile 1156
isVHostRunning 993
isVideo 217
isVideoCodecConfig 64
isVideoH264SEIListenerEmpty 1118
isVideoKeyFrame 64, 66
isVisited 1342
isWaitForVideoKeyFrame 1156
isWarnEnabled 614
isWowzaAudioOnly 780

J

JNDI_CONTEXT_NAME 598

K

KEYDATA_MODE_ENCRYPT 795
KEYDATA_MODE_INIT 794
KEYDATA_MODE_PLAYLIST 795
killClient 1309
killRTSPSession 1309

L

lastAuthenticateMethod 904
lastModified 8
len 210
length 9, 94, 1107, 1142
LICENSECOUNTER_DRM_BUYDRM_LIVE 1299
LICENSECOUNTER_DRM_BUYDRM_VOD 1299
LICENSECOUNTER_DRM_EZDRM_LIVE 1298
LICENSECOUNTER_DRM_EZDRM_VOD 1298
LICENSECOUNTER_DRM_VERIMATRIX_LIVE 1299
LICENSECOUNTER_DRM_VERIMATRIX_VOD 1299
LICENSECOUNTER_NDVR 1298
LICENSECOUNTER_PUBLISHER 1298
LICENSECOUNTER_PUBLISHERTRANSCODER 1299
LICENSECOUNTER_TOTAL 1300
LICENSECOUNTER_TRANSCODE_DECODE 1298
LICENSECOUNTER_TRANSCODE_DECODECOUNTAU
DIO 1299
LICENSECOUNTER_TRANSCODE_DECODECOUNTAU

- DIOVIDEO 1300
- LICENSECOUNTER_TRANSCODE_DECODECOUNTVID
EO 1300
- LICENSECOUNTER_TRANSCODE_DECODEPOLLING
1299
- LICENSECOUNTER_TRANSCODE_ENCODE 1298
- LICENSECOUNTER_TRANSCODE_ENCODECOUNTAU
DIO 1300
- LICENSECOUNTER_TRANSCODE_ENCODECOUNTAU
DIOVIDEO 1300
- LICENSECOUNTER_TRANSCODE_ENCODECOUNTVID
EO 1300
- LICENSECOUNTER_TRANSCODE_ENCODEPOLLING
1299
- LICENSECOUNTER_TRANSCODE_STREAMNAMES
1299
- liveStreamPacketizers 1168
- load 1024, 1040
- loadArchivedRecordings 472
- loadArchivedStore 485
- loadConfig 1345
- loadConfigFile 1270, 1275
- loadConfigProperties 107
- loadFile 662
- lock 771, 776, 784, 908, 1017, 1040
- lockRepeaterStreams 561, 948
- log 614, 615
- LOG4J_ID_KEY 597
- LOG4J_PACKAGE_NAME 596
- logDebug 824
- logError 824
- loggedConnect 905
- LOGGERNAME_SERVER 621
- logInfo 824
- logLiveChunk 532, 578, 583
- logLiveFragment 589
- logNotifier 992
- LogNotifyCalculateIncremental 604
- logVODChunk 532, 583
- logVODFragment 589
- logWarn 824
- longToByteArray 35, 36
- longToHexStr 94
- longToStr 94
- longValue 164
- M
- main 16, 993
- makeNewLoggerInstance 622
- MANIFEST_TAGNAME_CAN_PLAY 450
- MANIFEST_TAGNAME_CAN_RECORD 450
- MANIFEST_TAGNAME_CHUNK_GROUPING 451
- MANIFEST_TAGNAME_CURRENT_TIME 451
- MANIFEST_TAGNAME_HAS_ENCRYPTION 450
- MANIFEST_TAGNAME_PURGE_TIME 451
- MANIFESTFILE_KEY_ARTIFACT 380
- MANIFESTFILE_KEY_AUDIO_CODEC 380
- MANIFESTFILE_KEY_CHUNKINDEX 380
- MANIFESTFILE_KEY_DVRTIME 380
- MANIFESTFILE_KEY_ENCRYPTIONS 381
- MANIFESTFILE_KEY_INDEX 379
- MANIFESTFILE_KEY_METADATA 381
- MANIFESTFILE_KEY_NAME 380
- MANIFESTFILE_KEY_PACKETTIME 380
- MANIFESTFILE_KEY_SIZE 380
- MANIFESTFILE_KEY_START 379
- MANIFESTFILE_KEY_STOP 379
- MANIFESTFILE_KEY_TYPE 379
- MANIFESTFILE_KEY_UTCTIME 380
- MANIFESTFILE_KEY_VIDEO_CODEC 380
- MAXSTREAMINDEX 1166
- md5Digest 85
- MD5DigestUtils 85
- md5Lock 85
- MEDIACACHE_PREFIX 418
- MediaCasterItem 731
- MediaCasterList 736
- MediaCasterSettings 738
- MediaCasterStreamItem 741
- MediaCasterStreamManager 746
- MediaCasterStreamMap 749
- MEDIACASTERTYPE_LIVEREPEATER 715
- MEDIACASTERTYPE_RTPLIVE 715
- MEDIACASTERTYPE_SHOUTCAST 715
- MEDIACASTERTYPE_UNKNOWN 715
- MediaList 771
- mediaList 784
- MediaListRendition 778
- MediaListSegment 784

mediaListSegment 776
MediaStreamBase 1162
mediaStreamListeners 1168
MediaStreamMap 1168
MediaUtils 89
members 190
MILLS_PER_HOUR 124
MIMETYPE_VIDEO_MP4 440
MIMETYPES_JPEG 659
MIMETYPES_PNG 659
MIMETYPES_URL 659
MISSING 1047
MODE_PLAY 941
MODE_PUBLISH 941
MODE_UNKNOWN 941
ModuleBase 814
ModuleClientLogging 824
ModuleCore 829
ModuleFastPlay 838
ModuleMediaCaster 756
ModuleMediaCasterStreamMonitorAdvanced 848
ModuleProperties 851
monitors 847
msb0baseTime 97
msb1baseTime 97
MSG_STARTSTREAM 545
MSG_SWITCHSTREAM 545

N

name 771, 777
nameGroupId 1167
nameGroups 1167
NetworkUtils 91
newValue 100
newXPathFactory 109
next 1244
nextStreamId 1168
NO_OPTIONS 15
notifyActionOnCodecInfoAudio 1104
notifyActionOnCodecInfoVideo 1104
notifyActionOnMetaData 1103
notifyActionPause 1103
notifyActionPauseRaw 1103
notifyActionPlay 1102
notifyActionPublish 1103
notifyActionSeek 1103
notifyActionStop 1104
notifyActionUnPublish 1103
notifyDvrStreamManagerCreate 294
notifyDvrStreamManagerDestroy 295
notifyDvrStreamManagerInit 294
notifyDvrStreamStorageDeleted 487
notifyDvrStreamStoreCreate 487
notifyDvrStreamStoreDestroy 487
notifyDvrStreamStoreInit 487
notifyDvrStreamStoreLoaded 487
notifyLiveStreamDvrRecorderCreate 293
notifyLiveStreamDvrRecorderDestroy 294
notifyLiveStreamDvrRecorderInit 294
notifyLiveStreamPacketizerCreate 289
notifyLiveStreamPacketizerDestroy 290
notifyLiveStreamPacketizerInit 290
notifyLiveStreamTranscoderCreate 291
notifyLiveStreamTranscoderDestroy 291
notifyLiveStreamTranscoderInit 291
notifyMediaReaderClose 287
notifyMediaReaderCreate 287
notifyMediaReaderExtractMetaData 287
notifyMediaReaderInit 287
notifyMediaReaderOpen 287
notifyMediaStreamCreate 1174
notifyMediaStreamDestroy 1175
notifyMediaWriterOnFLVAddMetadata 284
notifyMediaWriterOnWriteComplete 284
notifyPlayPublish 1174
notifyPlayUnpublish 1173
notifySessionCreate 929
notifySessionDestroy 929
notifySlotDelete 1040
notifySlotSetValue 1040
notifyTimeReset 487
notifyVHostClientConnect 1349
notifyVHostCreate 1350
notifyVHostInit 1350
notifyVHostItemCreate 1345
notifyVHostItemDestroy 1346
notifyVHostItemUpdate 1346
notifyVHostShutdownComplete 1350
notifyVHostShutdownStart 1350

notifyVideoH264Packet 1118

O

offset 210

ON_METADATA_TYPE 450

onAACEncodeInfo 767

onAcceptConnection 361

onAcceptLicense 364

onAcceptorCreate 1285

onAcceptorDestroy 1285

onAfterDecodeFrame 1260

onAfterEncodeFrame 1262

onAfterScaleFrame 1260

onAnnounce 869, 917

onApplicationCreate 298

onApplicationDestroy 298

onApplicationInstanceCreate 297

onApplicationInstanceDestroy 297

onAppStart 756, 790, 848

onAppStop 756, 790, 848

onBeforeDecodeFrame 1260

onBeforeEncodeFrame 1262

onBeforeScaleFrame 1260

onBind 511, 513

onCalculateSourceAudioBitrate 1251

onCalculateSourceVideoBitrate 1251

onCall 791

onCallback 1124

onClientAccept 359

onClientConnect 359

onClientDisconnect 359

onClientReject 360

onCodecConfigAAC 764

onCodecInfoAudio 1123

onCodecInfoVideo 1123

onConnect 792

onConnectAccept 792

onConnectFailure 727

onConnectReject 793

onConnectStart 726

onConnectSuccess 726

onCreate 533

onData 722

onDescribe 869, 917

onDestroy 535

onDisconnect 362, 365, 762, 792

onDvrStreamManagerCreate 1179

onDvrStreamManagerDestroy 1179

onDvrStreamManagerInit 1179

onFile 74

onFillChunkDataPacket 536

onFillChunkEnd 535

onFillChunkStart 535

onFLVAddMetadata 1158

onFrameAAC 764

onFrameMP3 764

onGetParameter 869, 917

onGrabFrame 1257

onHeaderData 765, 767

onHTTPCupertinoEncryptionKeyCreateLive 796

onHTTPCupertinoEncryptionKeyCreateVOD 795

onHTTPCupertinoEncryptionKeyData 795

onHTTPCupertinoEncryptionKeyLiveChunk 796

onHTTPCupertinoEncryptionKeyRequest 795

onHTTPCupertinoEncryptionKeyVODChunk 796

onHTTPCupertinoStreamingSessionCreate 797

onHTTPCupertinoStreamingSessionDestroy 797

onHTTPRequest 513

onHTTPSsanJoseStreamingSessionCreate 798

onHTTPSsanJoseStreamingSessionDestroy 798

onHTTPSessionCreate 799

onHTTPSessionDestroy 799

onHTTPSmoothStreamingPlayReadyCreateLive 800

onHTTPSmoothStreamingPlayReadyCreateVOD 800

onHTTPSmoothStreamingSessionCreate 801

onHTTPSmoothStreamingSessionDestroy 801

onHTTPStreamerSessionCreate 572

onHTTPStreamerSessionDestroy 572

onIndex 534

onInit 534

onInitAfterLoadTemplate 1251

onInitBeforeLoadTemplate 1251

onInitStart 1250

onInitStop 1251

onLiveStreamDvrRecorderCreate 1187

onLiveStreamDvrRecorderDestroy 1187

onLiveStreamDvrRecorderInit 1187

onLiveStreamPacketizerCreate 1196

onLiveStreamPacketizerDestroy 1196

onLiveStreamPacketizerInit 1196
onLiveStreamTranscoderCreate 1206
onLiveStreamTranscoderDestroy 1206
onLiveStreamTranscoderInit 1206
onLog 601, 605
onMediaCasterCreate 724
onMediaCasterDestroy 724
onMediaReaderClose 1064
onMediaReaderCreate 1063
onMediaReaderExtractMetaData 1064
onMediaReaderInit 1063
onMediaReaderOpen 1063
onMediaStreamCreate 1134
onMediaStreamDestroy 1134
onMetaData 765, 767, 1122
onModuleLoad 789
onModuleUnload 789
onMP3EncodeInfo 767
onNewVHost 996
onOpen 534
onOptions 869, 918
onPause 870, 918, 1120
onPauseRaw 1122
onPlay 870, 918, 1119
onPlaylistItemStart 1212
onPlaylistItemStop 1212
onPublish 1120
onRecord 870, 918
onRedirect 870, 918
onRegisterPlayer 724
onRegisterStreamNameGroup 1254
onRejectConnection 361
onRejectLicense 364
onResetMediaCaster 729, 849
onResetStream 1255
onResponseWriteStart 974
onResponseWriteStop 974
onResult 788, 804
onRTPSessionCreate 802, 867
onRTPSessionDestroy 802, 867
onSeek 1120
onServerConfigLoaded 987
onServerCreate 985
onServerInit 985
onServerShutdownComplete 986
onServerShutdownStart 985
onSessionAudioDecodeCodecInfo 1254
onSessionAudioEncodeCodecInfo 1253
onSessionAudioEncodeCreate 1252
onSessionAudioEncodeInit 1252
onSessionAudioEncodeSetup 1253
onSessionDataEncodeCreate 1252
onSessionDataEncodeInit 1253
onSessionDestinationCreate 1251
onSessionVideoDecodeCodecInfo 1254
onSessionVideoEncodeCodecInfo 1253
onSessionVideoEncodeCreate 1252
onSessionVideoEncodeInit 1252
onSessionVideoEncodeSetup 1253
onSetParameter 869, 918
onSetSourceStream 725
onSetup 871, 919
onSharedObjectConnect 1018
onSharedObjectCreate 1018
onSharedObjectDestroy 1018
onSharedObjectDisconnect 1019
onShutdownStart 1254
onShutdownStop 1255
onSlotDelete 1029
onSlotSetValue 1029
onStop 1120
onStreamCreate 803, 848
onStreamDestroy 803, 848
onStreamStart 727, 1129
onStreamStop 727
onTeardown 871, 919
onTrim 765
onUnbind 511, 514
onUnPublish 1120
onUnRegisterPlayer 725
onUnregisterStreamNameGroup 1254
onValidateMediaCaster 728, 848
onValidateMediaCastersStart 728, 848
onValidateMediaCastersStop 728, 849
onVHostClientConnect 1330
onVHostCreate 1329
onVHostInit 1329
onVHostItemCreate 1328
onVHostItemDestroy 1328
onVHostItemUpdate 1328

onVHostShutdownComplete 1330
onVHostShutdownStart 1329
onVideoH264Packet 1127
onWriteComplete 1158
open 6, 1052, 1058, 1214
order 190
ORDERED 16
output 863
OutputStream 26

P

p 1162
PACKET_TIME 444
packetComplete 1086
packetizerLicenses 1167
packetOutput 1274
packetTime 381
PARAM1 809
PARAM10 811
PARAM2 810
PARAM3 810
PARAM4 810
PARAM5 810
PARAM6 810
PARAM7 810
PARAM8 810
PARAM9 810
PARAMMETHODNAME 809
parseAllowDomains 267
parseBodyForParams 521
parseIdString 734
parseQueryStr 104
PASSWORDFILEFORMAT_CLEAR 308
PASSWORDFILEFORMAT_UNKNOWN 308
pathToFileURL 102
pause 832, 951, 1140
PAUSE_PAUSE 1137
PAUSE_PLAY 1137
PAUSE_TOGGLE 1137
pauseRaw 836, 1140
pauseRecording 485, 495, 1185
peekByte 125
PICTUREMAXFILESIZE 659
PICTURETYPE_ARTISTLOGO 659

PICTURETYPE_COVERBACK 659
PICTURETYPE_COVERFRONT 658
PICTURETYPE_FILEICON 658
PICTURETYPE_ILLUSTRATION 659
PICTURETYPE_MOVIESCREENCAPTURE 659
PICTURETYPE_OTHER 658
PICTURETYPE_OTHERFILEICON 658
PICTURETYPE_PUBLISHERLOGO 659
ping 340
play 831, 950, 1139, 1140, 1209, 1233, 1236, 1242, 1243, 1244
play2 831
playDuration 778
PLAYEVENT_AFTERBUFFERFILL 1057
PLAYEVENT_AFTERMETADATA 1057
PLAYEVENT_BEFOREBUFFERFILL 1057
PLAYEVENT_BEFOREMETADATA 1057
PLAYEVENT_STARTPLAYBACK 1057
Playlist 1213
PlaylistItem 1216
PLAYSIZES_AUDIO_BYTES 1138
PLAYSIZES_AUDIO_COUNT 1138
PLAYSIZES_DATA_BYTES 1138
PLAYSIZES_DATA_COUNT 1138
PLAYSIZES_LOSS_BYTES 1138
PLAYSIZES_LOSS_COUNT 1139
PLAYSIZES_SIZE 1139
PLAYSIZES_VIDEO_BYTES 1138
PLAYSIZES_VIDEO_COUNT 1138
playStart 778
PLAYSTATUSTYPE_COMPLETE 1138
PLAYSTATUSTYPE_STOP 1138
PLAYSTATUSTYPE_SWITCH 1137
PLAYTRANSITION_APPEND 813
PLAYTRANSITION_APPEND_IMMEDIATE 813
PLAYTRANSITION_RESET 813
PLAYTRANSITION_RESET_IMMEDIATE 813
PLAYTRANSITION_STOP 813
PLAYTRANSITION_SWAP 813
PLAYTRANSITION_SWITCH 813
PLAYTRANSITION_UNKNOWN 813
PLAYTRANSITIONSTR_APPEND 812
PLAYTRANSITIONSTR_RESET 812
PLAYTRANSITIONSTR_STOP 812
PLAYTRANSITIONSTR_SWAP 812
PLAYTRANSITIONSTR_SWITCH 812

- PLAYTRANSITIONSTR_UNKNOWN 812
- postChunkAdded 474
- postChunksPurged 475
- preChunkAdded 474
- preChunksPurged 474
- previous 1244
- properties 509, 771, 776, 784, 905
- PROPERTY_ALLOWABLE_AV_PACKET_DELTA 424
- PROPERTY_APPEND_DISCONTINUITY_DELTA 418
- PROPERTY_ARCHIVE_STRATEGY 419
- PROPERTY_AUDIO_ONLY_CHUNK_TARGET_DURATION 425
- PROPERTY_BREAK_ON_PTS 426
- PROPERTY_CHUNK_CACHE_CLASS 422
- PROPERTY_CHUNK_DURATION_MINIMUM 427
- PROPERTY_CHUNK_GROUPING_SECONDS 418
- PROPERTY_CHUNK_MEMORY_CACHESIZE 428
- PROPERTY_CHUNK_READER_CLASS 420
- PROPERTY_CHUNK_WRITER_CLASS 421
- PROPERTY_CUPERTINO_ON_CHUNK_START_RESET_COUNTER 442
- PROPERTY_CUPERTINO_PLAYLIST_FORCE_LIVE 441
- PROPERTY_CUPERTINO_PLAYLIST_FORCE_NONLIVE 441
- PROPERTY_CUPERTINO_PLAYLIST_GZIP_THRESHOLD 442
- PROPERTY_CUPERTINO_PLAYLIST_USE_GZIP 442
- PROPERTY_DEBUG_CHUNK_RETRIEVALS 436
- PROPERTY_DEBUG_CUPERTINO_PLAYER_ADAPTER 437
- PROPERTY_DEBUG_FAILED_CHUNK_RETRIEVALS 436
- PROPERTY_DEBUG_LOG_INVALID_CHUNK_DETAILS 433
- PROPERTY_DEBUG_LOG_INVALID_CHUNK_MATCHER 433
- PROPERTY_DEBUG_LOG_VALID_CHUNK_DETAILS 434
- PROPERTY_DEBUG_LOG_VALID_CHUNK_MATCHER 434
- PROPERTY_DEBUG_MAX_INVALID_CHUNKS_LOGGED 433
- PROPERTY_DEBUG_MAX_VALID_CHUNKS_LOGGED 434
- PROPERTY_DEBUG_MAXIMUM_RAW_PACKETS 435
- PROPERTY_DEBUG_MBR_ALIGNMENT 438
- PROPERTY_DEBUG_MBR_ALIGNMENT_RESOLUTION 438
- PROPERTY_DEBUG_MBR_PLAYER_ADAPTER 437
- PROPERTY_DEBUG_METHODS 436
- PROPERTY_DEBUG_PLAYER_ADAPTER 437
- PROPERTY_DEBUG_PLAYLIST_REQUEST 438
- PROPERTY_DEBUG_RAW_PACKETS 435
- PROPERTY_DEBUG_RAW_PACKETS_MATCHER 435
- PROPERTY_DEBUG_REPEATER 437
- PROPERTY_DEBUG_SANJOSE_PLAYER_ADAPTER 437
- PROPERTY_DEBUG_SMOOTH_PLAYER_ADAPTER 437
- PROPERTY_DEBUG_STATE_CHANGE 436
- PROPERTY_DEBUG_TOSSED_HOLDERS 434
- PROPERTY_ENCRYPTION_INFO_DELEGATE 438
- PROPERTY_FILE_SYSTEM_CLASS 421
- PROPERTY_MANIFEST_PERSISTENT_CLASS 421
- PROPERTY_MAX_CHUNK_LOG 436
- PROPERTY_MAX_RECORDING_LENGTH 429
- PROPERTY_MBR_MINIMUM_PACKET_TIME_GAP_SIZE 428
- PROPERTY_MBR_MINIMUM_UTCTIME_GAP_SIZE 429
- PROPERTY_MBR_USE_UTC_FOR_ALIGNMENT 428
- PROPERTY_MEDIACACHE_ENABLED 430
- PROPERTY_MEDIACACHE_READER_CLASS 430
- PROPERTY_PACKET_DELTA_TO_NOTIFY 425
- PROPERTY_PACKET_DELTA_TO_RESET_TIME 424
- PROPERTY_PACKET_DURATION_MAXIMUM 428
- PROPERTY_PACKET_SORT_TIME 426
- PROPERTY_PLAYLIST_REQUEST_DELEGATE 438
- PROPERTY_RECORD_AUDIO 425
- PROPERTY_RECORD_DATA 425
- PROPERTY_RECORD_VIDEO 425
- PROPERTY_RECORDINGS_LOADER_CLASS 422
- PROPERTY_REPEATER_HEARTBEAT_DURATION 427
- PROPERTY_REPEATER_SHARED_SECRET 427
- PROPERTY_SANJOSE_ABST_DURATION_TOLERANCE 441
- PROPERTY_SANJOSE_ABST_TIMESCALE 441
- PROPERTY_SANJOSE_PLAYLIST_DELIVERYTYPE 439
- PROPERTY_SANJOSE_PLAYLIST_LIVE_STREAMTYPE 439
- PROPERTY_SANJOSE_PLAYLIST_MIMETYPE 438
- PROPERTY_SANJOSE_PLAYLIST_RECORDED_STREAM

-
- TYPE 439
 PROPERTY_SANJOSE_PLAYLIST_VERSION 439
 PROPERTY_SMOOTH_MANIFEST_H264_CODEC 433
 PROPERTY_SMOOTH_MANIFEST_LIVE_CAN_PAUSE 431
 PROPERTY_SMOOTH_MANIFEST_LIVE_CAN_SEEK 430
 PROPERTY_SMOOTH_MANIFEST_MAJOR_VERSION 432
 PROPERTY_SMOOTH_MANIFEST_MINOR_VERSION 432
 PROPERTY_SMOOTH_MANIFEST_RECORDED_CAN_PAUSE 431
 PROPERTY_SMOOTH_MANIFEST_RECORDED_CAN_SEEK 431
 PROPERTY_SMOOTH_MANIFEST_RECORDED_SPECIFY_DURATION 431
 PROPERTY_SMOOTH_MANIFEST_VERBOSE_DURATION 432
 PROPERTY_START_RECORDING_ON_STARTUP 425
 PROPERTY_STORAGE_DIRECTORY 419
 PROPERTY_TEXT_READER_CLASS 423
 PROPERTY_TEXT_WRITER_CLASS 423
 PROPERTY_WINDOW_DURATION 419
 PROTOCOLUSAGE_CUPERTINO 251
 PROTOCOLUSAGE_RTMP 250
 PROTOCOLUSAGE_RTMPE 251
 PROTOCOLUSAGE_RTMPS 251
 PROTOCOLUSAGE_RTMPT 251
 PROTOCOLUSAGE_RTMPTE 251
 PROTOCOLUSAGE_RTMPTS 251
 PROTOCOLUSAGE_RTP 251
 PROTOCOLUSAGE_SANJOSE 252
 PROTOCOLUSAGE_SMOOTH 251
 PROTOCOLUSAGE_TOTAL 252
 PROTOCOLUSAGE_WEBM 252
 PROTO_HTTPCUPERTINO 648
 PROTO_HTTPDVRCHUNK 648
 PROTO_HTTPSANJOSE 648
 PROTO_HTTPSCUPERTINO 648
 PROTO_HTTPSDVRCHUNK 648
 PROTO_HTTPSMOOTH 648
 PROTO_HTTPSSANJOSE 648
 PROTO_HTTPSSMOOTH 648
 PROTO_HTTPSSTREAMER 648
 PROTO_HTTPSTREAMER 647
 PROTO_RTMP 647
 PROTO_RTMPE 647
 PROTO_RTMPS 647
 PROTO_RTMPT 647
 PROTO_RTMPTE 647
 PROTO_RTMPTS 647
 PROTO_RTSP 647
 ptToDvr 505
 publish 830, 1100, 1162, 1224, 1225
 PublishingProviderBase 1230
 PublishingProviderLive 1233
 PublishingProviderMediaReader 1236
 purge 1016, 1034
 purgeEntries 453, 496
 purgeManifestEntries 486
 put 191, 192, 222, 223, 1021
 putAll 300
 putAudioPacketizerItem 874, 922
 putDvrRecorder 1116
 putFileInfo 564
 putFrame 654
 putGlobalLogValue 623
 putHTTPStreamerAdapter 1325
 putLiveStreamTranscoder 1117
 putMetaData 1157
 putRTSPSessionExtraLine 969
 putRTSPTunnelingSession 879
 putSlot 1010, 1035
 putStreamNameParts 563
 putStreamPacketizerItem 875, 923
 putVideoPacketizerItem 875, 923
- ## Q
- QUERY_PARAM_DVR 465
 queryStr 906
- ## R
- ranges 389
 read 7, 23
 READ_ACCESS_ALL 333
 READ_ACCESS_NONE 333
 READACCESS 1033, 1078
 readAppConfig 235
-

readAppInstConfig 282
readChunk 58
readChunkHeader 58
readConfig 983, 1000
readHeader 58, 59
readPrevChunkHeader 58
readVHostConfig 1327
readXMLConfig 1000
receiveAudio 834
RECEIVER_NAME_KEY 597
receiveVideo 834
record 950
recordingPaused 470
recordingReset 471
recordingResumed 470
recordingStarted 470
recordingStopped 471
redirectConnection 338, 339
redirectSession 570, 907, 924
redirectSessionCode 907
redirectSessionMessage 908
redirectSessionURL 908
referrer 906
refreshManifest 451
registerCallback 1101
registerLiveStreamTranscoder 1002
registerOnPlayStatus 1101
registerOnStatus 1101
registerPlayer 717, 743, 754
registerPlayRTPSession 274
rejectConnection 320, 337, 338
rejectSession 561, 908
release 744, 753, 1011, 1035
releaseMediaCaster 758
releaseSessionId 927
releaseSocketAddress 876
releaseStream 830
releaseUDPPortPair 877
reloadConfig 1345
reloadVHostConfig 978, 994
remove 135, 171, 192, 224, 753, 1022, 1349
removeAcceptorListener 1313
removeActionListener 917
removeAppInstance 235
removeApplication 1327
removeApplicationInstanceListener 233
removeApplicationListener 1313, 1349
removeClient 1011, 1023, 1035, 1309
removeClientListener 257, 1102
removeConnectionListener 319
removeDvrChunkListener 497
removeDvrRecorder 1116, 1177
removeDvrRecorderListener 293
removeDvrRecordingListener 487
removeDvrStoreListener 486
removeDvrStreamManagerListener 294
removeFrame 654
removeFromPlaylist 1243
removeGlobalLogValue 623
removeGlobalLogValues 622
removeHTTPStreamerSession 281
removeIdleWorkerListener 1320
removeItem 1214
removeListener 1245
removeLiveStreamPacketizer 1175
removeLiveStreamPacketizerListener 289
removeLiveStreamTranscoder 1117
removeLiveStreamTranscoderListener 290
removeMediaCasterListener 260
removeMediaReaderListener 286
removeMediaStreamListener 258, 1174
removeMediaWriterListener 284
removeMetaDataListener 766
removeModuleListener 264
removeNameGroup 1176
removePlayStreamByName 262
removePublisher 279
removeRendition 785
removeRTPSession 274
removeRTSPStream 909
removeRTSPTunnelingSession 878
removeSegment 772
removeServerListener 980, 997
removeSession 928
removeSessionListener 929
removeSharedObjectListener 259, 1023
removeSlotListener 1015, 1040
removeStream 1172
removeStreamDomainStr 565
removeTrack 944

- removeTrackInternal 944
- removeVHostItemListener 1345
- removeVideoH264SEIListener 1118
- renditions 784
- reparentClient 355, 1322
- ReplaceItem 100
- requestFilters 509
- requiresLayout 609
- reset 491, 743, 772, 1139, 1331, 1341
- resetMDC 622
- resetMediaCasterStream 283
- resetNoLookup 1139
- resetSentMetadataFlag 952
- resetStream 464, 483, 574, 757, 1182, 1193, 1202
- resetTimecodes 1144
- resolveMediaList 1050
- resolvePlayAlias 1130, 1131, 1132
- resolveStreamAlias 1130, 1132
- ResponseFunction 856, 857
- ResponseFunctions 862
- resumeEncoding 27
- resumeRecording 485, 495, 1185
- retrieveRawChunk 403, 469
- REVERSE 6
- rewind 1059
- RTPContext 874
- RTPDestination 881
- RTPPort 888
- RTPPushPublishSession 893
- RTPSession 908
- rtpSession 306
- RTPSessions 927
- RTPStream 943
- RTPUtils 1267
- rtpWriteListener 905
- rtspPlayRangeStart 907
- rtspPlayRangeStop 907
- rtspTunnelingSessionId 903
- run 1241
- S
- SANJOSE_F4M_STREAMINGTYPE_STREAMING 440
- SANJOSE_F4M_STREAMTYPE_DVR 440
- SANJOSE_F4M_STREAMTYPE_LIVE 440
- SANJOSE_F4M_STREAMTYPE_LIVEORRECORDED 440
- SANJOSE_F4M_STREAMTYPE_RECORDED 440
- SANJOSE_F4M_VERSION_1_0 440
- SANJOSE_F4M_VERSION_2_0 440
- SDPLOCATION_AUDIO 940
- SDPLOCATION_STREAM 940
- SDPLOCATION_VIDEO 940
- seek 7, 831, 1059, 1140, 1210, 1233, 1236
- SEEK_EXACT 1058
- SEEK_KEYCLOSE 1058
- SEEK_KEYDOWN 1057
- SEEK_KEYUP 1057
- SEEKTARGET_AUDIO 1057
- SEEKTARGET_ENHANCED 1057
- SEEKTARGET_VIDEOKEYFRAME 1056
- segments 771
- send 1015, 1039, 1097
- sendAMF3 1097
- sendClientOnStatusError 821
- sendControlBytes 1106
- sendDirect 1096, 1097
- sendDirectAMF3 1096
- sendInternal 1039
- sendLivePlaySeek 1087
- sendLivePlayStart 1086
- sendLivePlaySwitch 1087
- sendMessage 890
- sendOnMetadata 463, 1230
- sendPauseNotify 1092
- sendPlayReset 1088
- sendPlaySeek 1089
- sendPlayStart 1089, 1090, 1091
- sendPlayStatus 1091, 1092
- sendPlayStop 1088
- sendPlaySwitch 1090
- sendResponse 890
- sendResult 819, 820
- sendStreamNotFound 1086
- sendStreamOnStatusError 821
- sendUnpauseNotify 1092, 1093
- sendVODPlaySwitch 1087
- sendZeroLengthPacket 1061
- serialize 129, 142, 148, 165, 166, 178, 183, 195, 372, 375, 383, 387, 390, 394, 396, 446, 460, 655, 677, 709
- SERIALIZE_CURRENT_VERSION 381

serializeBody 660, 680, 683, 686, 688, 690, 695, 701, 704, 709
serializeDate 113
serializeFooter 654
serializeHeader 654
serializeInt 113
serializeString 115, 677
serializeStringLen 677
serializeStringNoLength 115
serializeTags 654
serializeZeroLengthString 114
Server 992
serverIp 906
serverPort 906
service 539
sessionClosed 718, 891
sessionId 903
sessionOpened 717, 891
SESSIONPROTOCOL_COUNT 553
SESSIONPROTOCOL_CUPERTINOSTREAMING 553
SESSIONPROTOCOL_DVRCHUNKSTREAMING 553
SESSIONPROTOCOL_MPEGDASHSTREAMING 553
SESSIONPROTOCOL_SANJOSESTREAMING 553
SESSIONPROTOCOL_SMOOTHSTREAMING 553
SESSIONPROTOCOL_UNKNOWN 553
SESSIONPROTOCOL_WEBMSTREAMING 553
sessionStart 760
sessionStop 760
SESSIONTYPE_LIVE 552
SESSIONTYPE_LIVEDVR 552
SESSIONTYPE_UNKNOWN 552
SESSIONTYPE_VOD 552
set 138, 139, 174, 175
setAbsTimecode 217
setAbsTimecodeLong 206
setAbsTimecodeShort 206
setAcceptConnection 254, 355
setAcceptConnectionDescription 354
setAcceptConnectionExObj 354
setAcceptConnectionObj 354
setAcceptSession 562
setActive 555
setAddressCount 892
setAdminInterfaceHostPort 1312
setAllowDomains 267
setAnnounce 913
setAppend 1093
setAppInstance 558, 716, 912, 1203
setAppInstanceName 953
setAppInstanceProperty 852
setApplication 609
setApplicationInstanceTouchTimeout 285
setApplicationProperty 851
setApplicationTimeout 264, 1316
setAppName 952
setAudioCodecId 780
setAudioHost 884
setAudioPort 883
setAudioSize 1080
setAudioTC 1082, 1083
setAuthenticatePlayHandler 911
setAuthenticatePublishHandler 911
setAuthenticationMethod 511, 517
setAVSyncMethod 956
setBandwidthLimit 831
setBaseClass 733, 1333
setBitrateAudio 779
setBitrateTotal 779
setBitrateVideo 779
setBlockUDPOut 892, 971
setBody 859
setBufferSize 691
setBufferTime 344, 832, 1079, 1141
setByteContainerLevel 207
setCanPlay 493
setCanRecord 493
setCheckIpAddr 955
setCheckSSRC 955
setChecksum 1048
setChunkCounter 209
setChunkGroupDuration 402
setClassName 196, 199
setClient 307, 314, 1086
setClientIdleFrequency 268, 1318
setClientProperty 852
setClientTimeout 1311
setClustered 1104
setCodecInfoAudio 464, 574
setCodecInfoVideo 464, 574
setCommandInterface 997
setCommandInterfaceHostPort 980, 996

setConfigDir 1341
setConnected 914
setConnectionLimit 1342
setConnectionTimeout 733
setConnectionValidator 319
setContentType 697
setCookieStr 562, 921
setCoreHandlerPoolSize 981, 998, 1311
setCoreTransportPoolSize 981, 998, 1310
setCurrentDvrTime 467
setData 684, 1026
setDataBuffer 215
setDataSize 1080
setDataTC 1083, 1084
setDataType 1084
setDebugLog 83
setDebugRTSPSession 920
setDefaultStreamingStore 482
setDescribe 913
setDescription 661, 681, 684, 733, 1335
setDirection 1045
setDirecton 8
setDomainName 1278
setDuration 1210, 1234, 1237
setDvrRecorder 1116
setDvrRecorderId 1183
setDvrRecorderList 293, 1112
setDvrSessionInfo 566
setDynamic 199
setDynamicLogProperties 982, 999, 1323
setEmbeddedFlag 691
setEncoding 609
setEncryptions 372
setExt 1108
setFastPlay 839
setFastPlaySettings 341, 1098
setFileExtension 1224
setFileIOPoolSize 1311
setFlags 678, 709
setFlashVer 334
setForceAMF0 860
setForceMPEGTSOut 970
setForceRTSPInterleaved 962
setFps 1044
setFrameCount 1263
setFrameIdentifier 684
setHandler 890
setHasEncryption 494
setHeader 526
setHeaderSize 1110
setHeight 781
setHost 882, 947
setHostType 884
setHTTPStreamerAdapter 557
setHTTPStreamerItem 539
setHTTPStreamerList 277
setHTTPStreamerSession 1053, 1113
setID 541
setId 204
setIdleCheckFrequency 1319
setIdleFrequency 348, 919
setIdleHandler 915
setIdleMinimumWaitTime 1322
setIdleWorkerCount 1319
setIdStr 677, 708
setIndex 1217
setInnerObj 200
setIntData 151
setIntHeader 526
setIOPerformanceCounter 921
setIoPerformanceCounter 995
setIoSession 925
setIp 913
setIpAddress 559, 1279
setIsPlaying 1079
setKeepAliveTime 734
setKeepAliveTimeout 1319
setLanguage 696
setLastAccessed 1048
setLastAuthenticateMethod 912
setLastClientId 1028
setLastOperation 1028
setLastSentAbsTimecode 208
setLastValidateTime 352
setLive 964
setLivePlaylist 589
setLiveRepeaterCapabilities 356
setLiveStreamDvrRecorderControl 293
setLiveStreamingPacketizer 558
setLiveStreamPacketizer 745, 832, 1113

setLiveStreamPacketizerControl 282
setLiveStreamPacketizerId 1193
setLiveStreamPacketizerList 277, 356, 1111
setLiveStreamRepeater 745, 1114
setLiveStreamTranscoderControl 292
setLiveStreamTranscoderItem 1203
setLiveStreamTranscoderList 291, 356, 1112
setLock 778, 786
setLoggedConnect 912
setLongTimecode 208
setMaximumPendingReadBytes 266, 1317
setMaximumPendingWriteBytes 265, 1317
setMaximumSetBufferTime 266, 1317
setMaxRTCPWaitTime 956
setMaxStorageDirDepth 276
setMediaCasterDef 717
setMediaCasterHostPortConfig 739
setMediaCasterId 716
setMediaCasterItem 1128
setMediaCasterPlay 1115
setMediaCasterProcessorCount 739
setMediacasterRTPRTSPRTPTransportMode 295
setMediaCasterType 721
setMediaCasterValidator 285
setMediaList 786
setMediaListProvider 295
setMediaListSegment 779
setMediaReaderItem 1052, 1058
setMediaWriterItem 1156
setMergeOnMetadata 1116
setMessageBytes 858
setMetaDataProvider 1109
setMimeType 661
setMode 961
setMPEGTSAudioBitrate 971
setMPEGTSAudioLanguage 957
setMPEGTSAudioPID 958
setMPEGTSOut 964
setMPEGTSProgramID 957
setMPEGTSVideoBitrate 971
setMPEGTSVideoPID 957
setMultiplier 1044
setName 232, 253, 732, 773, 780, 885, 1014, 1027, 1039, 1081, 1141, 1333, 1341
setNetConnection 1085
setNetConnectionIdleFrequency 1318
setNetConnectionProcessorCount 1313
setNew 206
setObjectEncoding 152, 156, 208, 354
setOffsetToNextTag 691
setOpen 1098
setOutHost 959
setPacketizeAudio 1194
setPacketizeData 1195
setPacketizeVideo 1194
setParent 1041, 1156
setPersistent 1012, 1022, 1036
setPictureData 661
setPictureDataAsURL 661
setPictureType 661
setPingTimeout 265, 1316
setPlay 1085
setPlayClass 1334
setPlayDuration 568, 782
setPlayer 1094
setPlayLogged 557
setPlayStart 568, 781
setPollingInterval 1245
setPort 610, 1279
setProcessorCount 1280
setProperties 511, 514, 539, 1052, 1062, 1342
setProperty 301, 732, 1013, 1014, 1036, 1037, 1334
setPublishAudioCodecId 1114
setPublishVideoCodecId 1114
setQueryStr 561, 916, 1107
setRange 387
setRealTimeStartTime 1210, 1234, 1237
setReceiveAudio 1098
setReceiveVideo 1099
setReceiveVideoFPS 1099
setReconnectWaitTime 721
setRecord 1084
setRecordAudio 1184
setRecordData 1184
setRecordingName 1185
setRecordingStreamStore 482
setRecordVideo 1184
setRedirectSession 569, 923
setRedirectSessionBody 570
setRedirectSessionCode 569, 924

setRedirectSessionContentType 571
setRedirectSessionMessage 924
setRedirectSessionURL 570, 924
setReferrer 561, 916
setRemoteHost 609
setRepeat 1214, 1244
setRepeaterEdge 1194
setRepeaterMediaCasterDef 543
setRepeaterOriginUrl 266, 352, 835
setRepeaterQueryString 267
setRequestFilters 510, 517
setResetPlayStream 949
setResponseCode 527
setResyncAudioVideoOnSR 951
setRetAMFNumber 859
setRsoStorageDir 269
setRTPAVSyncMethod 273
setRTPDestination 969
setRTPIIdleFrequency 268, 1318
setRTPIgnoreProfileLevelId 970
setRTPMaxRTCPWaitTime 273
setRTPPlayAuthenticationMethod 272
setRTPPublishAuthenticationMethod 272
setRTPSession 307, 314, 894
setRTPStream 1111
setRTPWrapped 884
setRTSP 958
setRTSPAlwaysUseSDPPorts 968
setRTSPBindIpAddress 288, 966
setRTSPConnectionAddressType 288, 966
setRTSPConnectionIpAddress 288, 966
setRTSPMaximumPendingWriteBytes 286, 965
setRTSPOriginAddressType 289, 967
setRTSPOriginIpAddress 289, 967
setRTSPPlayRangeStart 920
setRTSPPlayRangeStop 920
setRTSPPull 958
setRTSPSessionDescription 963
setRTSPSessionName 963
setRTSPSessionTimeout 286, 965
setRTSPTunnelingSessionId 920
setSDPData 894
setSDPLang 969
setSendOnMetadata 1210, 1231, 1246
setSendPlayStopLogEvent 1108
setSendPublishStopLogEvent 1109
setSendRecordStopLogEvent 1108
setSendSDESEvents 963
setSeq 217
setServer 1349
setServerIp 559, 915
setServerPort 560, 915
setSession 946
setSessionId 554, 909
setSessionProtocol 559
setSessionTimeout 556
setSessionTracker 593
setSessionType 558
setSessionValid 911
setSharedObjectReadAccess 271, 348
setSharedObjectWriteAccess 272, 348
setShutdownClient 355, 921
setShutdownOnRelease 744
setShuttingDown 1322
setSize 204, 214, 781
setSlotValue 1037, 1038
setSlotVersion 1027
setSoVersion 1027
setSrc 205, 216, 857, 1080
setSSLConfig 1282
setSslFactoryClass 1280
setStartLiveOnPreviousBufferTime 1247
setStartLiveOnPreviousKeyFrame 1246
setStartOnPreviousBufferTime 1234
setStartOnPreviousKeyFrame 1234
setStartRecordingOnStartup 1186
setStartStartupStreams 1326
setStartTC 1045
setStartTCOffset 1045
setStorageDir 1015, 1023, 1040
setStream 555, 716
setStreamAudioSampleAccess 270, 350
setStreamDomainProtectionActive 544
setStreamExt 563, 744, 953
setStreamFileMapper 276
setStreamKeyDir 268
setStreamName 562, 953, 1171, 1202
setStreamNameAliasProvider 278
setStreamNameLogging 954
setStreamPort 883

setStreamPosition 564, 1062
setStreamProperty 852
setStreamQueryStr 959
setStreamReadAccess 270, 350
setStreamSessionId 954
setStreamSessionIp 955
setStreamSessionVersion 954
setStreamStorageDir 268
setStreamTimeout 718
setStreamType 254, 337, 731, 833, 952, 1096, 1224
setStreamTypeStr 543
setStreamVersionHandler 483
setStreamVideoSampleAccess 270, 349
setStreamWriteAccess 271, 351
setSuspended 1281
setSwitchLog 1245
setTargetEncoding 156
setTextEncoding 660, 681, 696, 702, 705
setThreadContext 357
setTimecode 205, 216, 699, 860
setTimecodes 216
setTimeOffsetBetweenItems 1247
setTimeoutSession 556
setTimesInMilliseconds 1246
setTimeStampFormat 696
setTrack 888
setTranscodeResult 1117
setTranscoderName 1202
setTransportMode 961
setTTL 884, 891
setType 125, 205, 215, 781, 860
setUDPManagedDeliveryCount 970
setUDPManagedDeliveryDelay 970
setUri 560, 916
setURL 684, 705
setUserAgent 560, 910
setUserAgents 980, 997
setUserHTTPHeader 568
setUserQueryStr 566
setValid 742
setValidateFMLEConnections 290, 357
setValidationFrequency 265, 1316
setValue 681, 699, 702
setVersion 1012, 1036
setVersionFile 1156
setVHost 306, 314, 540, 554, 910
setVideoCodecId 780
setVideoHost 883
setVideoPort 882
setVideoSize 1080
setVideoTC 1083
setVisited 1342
setVODLastTimeTC 968
setVODPlayLen 968
setVODStartTimeTC 967
setWaitForVideoKeyFrame 1156
setWidth 781
setWowzaAudioOnly 780
SharedObject 1033, 1034
SHARED_OBJECT_CMD_CONNECT 1008
SHARED_OBJECT_CMD_CONNECTSUCCESS 1009
SHARED_OBJECT_CMD_DELETE 1009
SHARED_OBJECT_CMD_DISCONNECT 1008
SHARED_OBJECT_CMD_ERROR 1009
SHARED_OBJECT_CMD_SEND 1009
SHARED_OBJECT_CMD_SETVALUE 1008
SHARED_OBJECT_STATUS_CHANGE 1009
SHARED_OBJECT_STATUS_CLEAR 1009
SHARED_OBJECT_STATUS_DELETE 1009
SHARED_OBJECT_STATUS_SUCCESS 1009
shortValue 164
shouldDeleteArchivedStream 369, 501
shouldDvrRecord 1189
shouldLoadArchivedStream 369, 501
shouldLoadStream 472
shouldLoadStreamVersion 473
shouldStartRecordingOnStartup 1186
shutdown 231, 252, 491, 531, 555, 582, 588, 593, 717, 743, 750, 874, 889, 910, 949, 1082, 1139, 1182, 1192, 1202, 1308, 1349
shutdownAppInstance 235
shutdownApplication 1323
shutdownClient 252, 336, 907
shutdownRTPSession 879
shutdownSession 540
shutdownStream 757
sinfo 1162
size 136, 146, 171, 191, 1011, 1021, 1034, 1107, 1142, 1284
skipByte 126
sloppyGetTrack 945
splitCookie 71

splitPragmas 70
splitQueryStr 70
stackTraceToString 42
stampToString 95
start 100, 993
startApplicationInstance 1324
startAudioData 1226
startAudioPacket 1111
startChunk 463, 573
startCommandInterface 977, 993
startDataData 1228
startDataPacket 1111
startMediaCasterStream 283
startPlay 1142
startPlayback 1061
startPublishing 1105
startRecording 484, 495, 1184
startRTPPull 1268, 1269
startServer 994
startStartupStreams 1326
startStream 747, 1182, 1193
startsWith 37
startVHost 978, 994
startVHosts 978, 995
startVideoData 1226
startVideoPacket 1111
STAT_connect_application_not_available 643
STAT_connect_application_not_found 644
STAT_connect_bad_gateway 644
STAT_connect_internal_error 644
STAT_connect_license_limit 644
STAT_connect_pending_wating 643
STAT_connect_redirect 644
STAT_connect_rejected_by_application 643
STAT_connect_rejected_by_module 644
STAT_connect_resource_limit 644
STAT_connect_service_unavailable 644
STAT_connect_successful 643
STAT_connect_unknown_protocol 643
STAT_general_internal_error 646
STAT_general_successful 646
STAT_play_bad_request 645
STAT_play_internal_error 645
STAT_play_rejected_by_application 645
STAT_play_rejected_by_module 645
STAT_play_stream_not_found 645
STAT_play_successful 645
STAT_play_unsupported_media_type 645
STAT_publish_bad_request 645
STAT_publish_in_use 646
STAT_publish_internal_error 646
STAT_publish_rejected_by_application 646
STAT_publish_successful 645
STAT_publish_unsupported_media_type 646
STAT_stop_client_disconnect 646
STAT_stop_successful 646
stateChange 488
statusCodeToStr 72
stopAdminAgent 983, 993
stopCommandInterface 977, 994
stopMediaCasterStream 283
stopName 1082, 1141
stopPublishing 1105
stopRecording 484, 495, 1185
stopRTPPull 1267
stopServer 993
stopStartupStreams 1326
stopStream 747
stopVHost 978, 994
stopVHosts 978, 994
storeChunks 486, 496
storeOnMetadata 486, 496
Stream 1240
streamCodecToString 55
streamExists 747, 948
STREAMINFO_SESSIONATTRIBUTES 943
STREAMINFO_SESSIONBANDWIDTH 942
STREAMINFO_SESSIONCONNECTIONDATA 942
STREAMINFO_SESSIONEMAILADDRESS 942
STREAMINFO_SESSIONINFORMATION 942
STREAMINFO_SESSIONNAME 942
STREAMINFO_SESSIONPHONENUMBER 942
STREAMINFO_SESSIONPORIGIN 942
STREAMINFO_SESSIONPROTOCOLVERSION 941
STREAMINFO_SESSIONREPEATTIMES 943
STREAMINFO_SESSIONTIMEZONES 943
STREAMINFO_SESSIONTIMING 943
STREAMINFO_SESSIONURI 942
StreamItem 1333
streamLicenses 1167

StreamList 1336
streamLock 1167
streamMonitorAudioStartTimeout 847
streamMonitorAudioTCNegTolerance 846
streamMonitorAudioTCPosTolerance 846
streamMonitorAudioTCToleranceEnable 846
streamMonitorAudioTimeout 847
streamMonitorAVSyncTolerance 847
streamMonitorAVSyncToleranceEnable 847
streamMonitorDataTCNegTolerance 846
streamMonitorDataTCPosTolerance 846
streamMonitorDataTCToleranceEnable 846
streamMonitorDebug 847
streamMonitorResetNameGroups 847
streamMonitorStreamStartTimeout 845
streamMonitorStreamTimeout 845
streamMonitorVideoStartTimeout 847
streamMonitorVideoTCNegTolerance 846
streamMonitorVideoTCPosTolerance 846
streamMonitorVideoTCToleranceEnable 846
streamMonitorVideoTimeout 847
streamNames 1167
streamNamesLock 1167
streamNameToGroup 1167
streamNameToValidFilename 47
streamPacketizers 907
streams 903, 1167
streamsOrder 903
streamSrcToMediaCaster 753
STREAMTIMEOUTREASON_GOOD 714
STREAMTIMEOUTREASON_MISSING 714
STREAMTIMEOUTREASON_NORTSPSESSION 714
STREAMTIMEOUTREASON_NOSESSION 714
STREAMTIMEOUTREASON_NOSTREAM 714
STREAMTIMEOUTREASON_NOTIMEOUT 714
STREAMTIMEOUTREASON_NOURL 714
STREAMTIMEOUTREASON_RECONNECTRUNNING 714
STREAMTIMEOUTREASON_UNKNOWN 713
streamToFileForRead 1125
streamToFileForWrite 1126
streamToIndex 1171
StreamUtils 1272
StringUtils 93
suspendAllHostPorts 1323
suspendAllVHosts 982, 995
suspendCommandInterface 993
suspendEncoding 27
suspendServer 983, 995
switchName 1081, 1141
switchPlay 1142
switchSetupToMPEGTS 960
SystemUtils 97

T

TAG_APIC 669
TAG_COMM 676
TAG_LINK 669
TAG_PRIV 669
TAG_RBUF 669
TAG_SYLT 669
TAG_TALB 669
TAG_TBPM 670
TAG_TCOM 670
TAG_TCON 670
TAG_TCOP 670
TAG_TDEN 670
TAG_TDLY 670
TAG_TDOR 670
TAG_TDRC 670
TAG_TDRL 671
TAG_TDTG 671
TAG_TENC 671
TAG_TEXT 671
TAG_TFLT 671
TAG_TIPL 671
TAG_TTT1 671
TAG_TTT2 671
TAG_TTT3 671
TAG_TKEY 672
TAG_TLAN 672
TAG_TLEN 672
TAG_TMCL 672
TAG_TMED 672
TAG_TMOO 672
TAG_TOAL 672
TAG_TOFN 672
TAG_TOLY 673
TAG_TOPE 673

- TAG_TOWN 673
- TAG_TPE1 673
- TAG_TPE2 673
- TAG_TPE3 673
- TAG_TPE4 673
- TAG_TPOS 673
- TAG_TPRO 674
- TAG_TPUB 674
- TAG_TRCK 674
- TAG_TRSN 674
- TAG_TRSO 674
- TAG_TSOA 674
- TAG_TSOP 674
- TAG_TSOT 674
- TAG_TSRC 674
- TAG_TSSE 675
- TAG_TSST 675
- TAG_TXXX 676
- TAG_UNKN 669
- TAG_WCOM 675
- TAG_WCOP 675
- TAG_WOAF 675
- TAG_WOAR 675
- TAG_WOAS 675
- TAG_WORS 676
- TAG_WPAY 676
- TAG_WPUB 676
- TAG_WXXX 676
- TAGS_TEXTINFORMATION 675
- TEMP_CONSOLE_APPENDER_NAME 598
- TEMP_LIST_APPENDER_NAME 598
- terminate 1339
- testFlashVersion 353
- testNextByte 125
- TEXTENCODING_ISO_8859_1 668
- TEXTENCODING_UTF16 668
- TEXTENCODING_UTF16BE 668
- TEXTENCODING_UTF8 669
- ThreadPool 1338
- TIME_MAP_TYPE 450
- timeCreated 905
- timeReset 471
- TIMESTAMP_RULE_FORMAT 597
- TIMESTAMPFORMAT_MILLISECONDS 695
- TIMESTAMPFORMAT_MPEG 695
- toArray 147
- toByteBuffer 147
- toDvr 505
- toHex 41
- toLong 41
- toNTPTIME 98
- toSMILString 772, 778, 785
- toString 143, 149, 165, 179, 183, 195, 205, 216, 302, 373, 375, 388, 391, 394, 396, 734, 771, 778, 784, 881, 1217, 1280, 1335, 1342
- totalIOPerformance2Last 905
- totalIOPerformanceLast 905
- touch 285, 336, 554, 910, 965, 1049, 1182, 1194
- touchApplicationInstance 1323, 1324
- toValidFilename 47
- trait 190
- transportFindBestMatch 964
- traverseDirectory 48
- triggerAMF3Switch 130
- trim 1100, 1162
- trimTrailingZero 677
- truncatePacket 214
- type 125, 381, 777
- U
- UDPAAppender 608
- unbind 889
- unbindAllHostPorts 1323
- unbindAllVHosts 983, 995
- uncacheRTTPStream 877
- unlock 1017, 1041
- unlockRepeaterStreams 948
- unpublish 1225
- unregisterCallback 1100
- unregisterLiveStreamTranscoder 1002
- unregisterOnPlayStatus 1101
- unregisterOnStatus 1101
- unregisterPlayer 717, 743, 754
- updateIOPerformance 995
- updateLoggingDuration 997, 1107, 1315
- updateLoggingValues 532, 559, 583, 589, 916, 1107, 1142
- updateOnCuePointTimecode 66
- updateSDPDestination 1267
- uri 906

URL_SAFE 16

urlToId 103

URLUtils 102

userAgent 904

userExists 313

UTC_TIME 444

utcTime 381

utcToDvr 505

V

validateConnection 363

validateLicense 366

validateNewConnection 319

validStreamDomainToString 565

valueOf 95, 444

values 444

versionFile 48

vhost 306, 903

VHOST_DEFAULT 1298

VHostItem 1341

VHostList 1345

VHostSingleton 1348

videoCodecId 777

videoCodecStringToId 56

videoCodecToMetaDataString 55

videoCodecToString 56

videoCodecTypeToString 90

videoPacketizers 907

VIDEOSAMPLE_ACCESS_ALL 332

VIDEOSAMPLE_ACCESS_NONE 333

VIDEOSAMPLEACCESS 1078

W

warn 618

width 777

WMSLogger 613

WMSLoggerFactory 621

WMSLoggerIDs 649

WMSProperties 300

wowzaAudioOnly 777

wrap 147

write 27, 860, 861

WRITE_ACCESS_ALL 333

WRITE_ACCESS_NONE 333

WRITEACCESS 1033, 1078

writeAppConfig 236

writeAppInstConfig 282

writeChunk 60

writeCodecConfig 1269, 1270

writeConfig 983, 1000

writeDeleteError 1016, 1038

writeDuration 63

writeError 1038

writeGeneratedKeyFrame 1060

writeHeader 59

writePackets 62, 63, 1059, 1060, 1155

writeSetValueError 1015, 1038

writeShortHeader 59

writeString 157

writeVHostConfig 1327

writeXMLConfig 1001

X

XMLUtils 106