



---

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	<a href="#"><u>FORWARD</u></a> Value: <b>1</b>
public static final	<a href="#"><u>REVERSE</u></a> Value: <b>-1</b>

### Method Summary

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

void	<a href="#"><code>open()</code></a> Open the media asset
int	<a href="#"><code>read(byte[] buf, int off, int size)</code></a> Read bytes from the media asset
void	<a href="#"><code>seek(long pos)</code></a> Seek to a position in the media asset
void	<a href="#"><code>setDirecton(int directon)</code></a> 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	<a href="#">AMFUtils()</a>
--------	----------------------------

### Method Summary

static <a href="#">AMFData[]</a>	<a href="#">convertParams</a> (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/rfc-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	<a href="#">Base64.InputStream</a> Base64.InputStream
class	<a href="#">Base64.OutputStream</a> Base64.OutputStream

## Field Summary

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

## Method Summary

<code>static byte[]</code>	<a href="#"><u>decode</u></a> (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>	<a href="#"><u>decode</u></a> (String s) Decodes data from Base64 notation, automatically detecting gzip-compressed data and decompressing it.
<code>static byte[]</code>	<a href="#"><u>decode</u></a> (String s, int options) Decodes data from Base64 notation, automatically detecting gzip-compressed data and decompressing it.
<code>static void</code>	<a href="#"><u>decodeFileToFile</u></a> (String infile, String outfile) Reads infile and decodes it to outfile.
<code>static byte[]</code>	<a href="#"><u>decodeFromFile</u></a> (String filename) Convenience method for reading a base64-encoded file and decoding it.
<code>static boolean</code>	<a href="#"><u>decodeToFile</u></a> (String dataToDecode, String filename) Convenience method for decoding data to a file.
<code>static Object</code>	<a href="#"><u>decodeToObject</u></a> (String encodedObject) Attempts to decode Base64 data and deserialize a Java Object within.

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

#### 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/rfc-1940.html>.  
Constant value: **32**

---

## Methods

### main

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

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

---

### encodeObject

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

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

#### Parameters:

`serializableObject` - The object to encode

#### Returns:

The Base64-encoded object

---

### encodeObject

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



(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	<a href="#">Base64.InputStream</a> (java.io.InputStream in) Constructs a <a href="#">Base64.InputStream</a> in DECODE mode.
public	<a href="#">Base64.InputStream</a> (java.io.InputStream in, int options) Constructs a <a href="#">Base64.InputStream</a> in either ENCODE or DECODE mode.

### Method Summary

int	<a href="#">read</a> () Reads enough of the input stream to convert to/from Base64 and returns the next byte.
int	<a href="#">read</a> (byte[] dest, int off, int len) Calls <a href="#">read()</a> 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	<a href="#">Base64.OutputStream</a> ( java.io.OutputStream out) Constructs a <a href="#">Base64.OutputStream</a> in ENCODE mode.
public	<a href="#">Base64.OutputStream</a> ( java.io.OutputStream out, int options) Constructs a <a href="#">Base64.OutputStream</a> in either ENCODE or DECODE mode.

## Method Summary

void	<a href="#">close</a> () Flushes and closes (I think, in the superclass) the stream.
void	<a href="#">flushBase64</a> () Method added by PHIL.
void	<a href="#">resumeEncoding</a> () Resumes encoding of the stream.
void	<a href="#">suspendEncoding</a> () Suspends encoding of the stream.
void	<a href="#">write</a> (byte[] theBytes, int off, int len) Calls <a href="#">write(int)</a> repeatedly until len bytes are written.
void	<a href="#">write</a> (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	<a href="#">alphas</a>
public static final	<a href="#">hexadecimal</a>

### Constructor Summary

public	<a href="#">BufferUtils()</a>
--------	-------------------------------

### Method Summary

static int	<a href="#">byteArrayToInt</a> (byte[] b) Convert byte array to int
static int	<a href="#">byteArrayToInt</a> (byte[] b, int offset) Conver byte array to int with offset
static int	<a href="#">byteArrayToInt</a> (byte[] b, int offset, int count) Convert byte array to int with offset.
static int	<a href="#">byteArrayToInt</a> (byte[] b, int offset, int count, boolean isReverse) Convert byte array to int with offset.
static long	<a href="#">byteArrayToLong</a> (byte[] b) Convert byte array to long
static long	<a href="#">byteArrayToLong</a> (byte[] b, int offset) Conver byte array to long with offset
static long	<a href="#">byteArrayToLong</a> (byte[] b, int offset, int count) Convert byte array to long with offset.
static long	<a href="#">byteArrayToLong</a> (byte[] b, int offset, int count, boolean isReverse) Convert byte array to long with offset.
static int	<a href="#">byteArrayToShort</a> (byte[] b) Convert byte array to int

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

### Method Summary

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

static String	<a href="#">formatUtcTime</a> (long utcTime)
static String	<a href="#">stackTraceToString</a> (Throwable e)
static String	<a href="#">toHex</a> (byte value) Format a byte value to a 0xff format
static String	<a href="#">toHex</a> (int value) Format a byte value to a 0xffffffff format
static String	<a href="#">toLong</a> (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	<a href="#">ElapsedTimer()</a> Construct a new ElapsedTimer and start the clock
--------	--

### Method Summary

java.util.Date	<a href="#">getDate()</a> Get the date the object was created
String	<a href="#">getDateString()</a> Get the date object was created as formatted String
long	<a href="#">getTime()</a> Get elapsed time object in existence (milliseconds)
double	<a href="#">getTimeSeconds()</a> Get elapsed time object in seconds
String	<a href="#">getTimeString()</a> 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	<a href="#">FileUtils()</a>
--------	-----------------------------

### Method Summary

static void	<a href="#">copyFile</a> (java.io.File fromFile, java.io.File toFile) Simple file copy routine
static void	<a href="#">copyFile2</a> (java.io.File in, java.io.File out)
static boolean	<a href="#">deleteDirectory</a> (java.io.File path)
static byte[]	<a href="#">fileToByteArray</a> (java.io.File file)
static String	<a href="#">streamNameToValidFilename</a> (String name) Encode a stream name (deal with path elements) to a valid filename.
static String	<a href="#">toValidFilename</a> (String name) Encode a name to a valid filename.
static void	<a href="#">traverseDirectory</a> (java.io.File dir, <a href="#">IFileProcess</a> fileNotify) Traverse a directory recursively calling fileNotify for each file and folder encountered
static java.io.File	<a href="#">versionFile</a> (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	<a href="#"><u>FLV_CHUNKHEADER_BUFFERSIZE</u></a> Size of temporary buffer needed for flv reading (byte[]) Value: <b>13</b>
public static final	<a href="#"><u>FLV_CHUNKHEADER_FIRSTBYTE</u></a> Header values: first byte of packet data Value: <b>3</b>
public static final	<a href="#"><u>FLV_CHUNKHEADER_HEADERSIZE</u></a> Size of packet header (byte[]) Value: <b>11</b>
public static final	<a href="#"><u>FLV_CHUNKHEADER_ISIZE</u></a> Header values: packet size Value: <b>1</b>
public static final	<a href="#"><u>FLV_CHUNKHEADER_IMECODE</u></a> Header values: timecode (milliseconds) Value: <b>2</b>
public static final	<a href="#"><u>FLV_CHUNKHEADER_ITYPE</u></a> Header values: packet type Value: <b>0</b>
public static final	<a href="#"><u>FLV_CHUNKHEADER_SECONDBYTE</u></a> Header values: second byte of packet data Value: <b>4</b>
public static final	<a href="#"><u>FLV_CHUNKHEADER_VALUESIZE</u></a> Size of header values array (long[]) Value: <b>5</b>
public static final	<a href="#"><u>FLV_DFRAME</u></a> D video frame type (partial frame based on key frame) Value: <b>3</b>
public static final	<a href="#"><u>FLV_KFRAME</u></a> Key video frame type Value: <b>1</b>

public static final	<a href="#">FLV_PFRAME</a> P video frame type (partial frame based on previous frame) Value: <b>2</b>
public static final	<a href="#">FLV_TCINDEXAUDIO</a> Value: <b>0</b>
public static final	<a href="#">FLV_TCINDEXDATA</a> Value: <b>2</b>
public static final	<a href="#">FLV_TCINDEXVIDEO</a> Value: <b>1</b>
public static final	<a href="#">FLV_UFRAME</a> Unknown video frame type Value: <b>0</b>

## Constructor Summary

public	<a href="#">FLVUtils()</a>
--------	----------------------------

## Method Summary

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

static int	<a href="#"><u>getFrameType</u></a> (int[] values) Given the headers values (including first byte of the packet), determine the type of video frame (FLV_*FRAME)
static long	<a href="#"><u>getLastTC</u></a> (java.io.File file) Get the duration of an .flv file.
static OnMetadataBasic	<a href="#"><u>getOnMetadataData</u></a> ( <a href="#"><u>AMFPacket</u></a> metaDataPacket)
static int	<a href="#"><u>getVideoCodec</u></a> ( <a href="#"><u>AMFPacket</u></a> packet) Get the codec id for this video packet.
static int	<a href="#"><u>getVideoCodec</u></a> (int value) Return the codec portion of the first byte of an video packet.
static int	<a href="#"><u>getVideoFrameType</u></a> ( <a href="#"><u>AMFPacket</u></a> packet)
static int	<a href="#"><u>getVideoTimecodeOffset</u></a> ( <a href="#"><u>AMFPacket</u></a> packet) Get the timecode offset in milliseconds between the PTS and DTS for this frame.
static int	<a href="#"><u>getVideoTimecodeOffset</u></a> (byte[] buffer) Get the timecode offset in milliseconds between the PTS and DTS for this frame.
static java.util.List	<a href="#"><u>interleavePackets</u></a> (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	<a href="#"><u>interleavePackets</u></a> (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	<a href="#"><u>isAudioCodecConfig</u></a> ( <a href="#"><u>AMFPacket</u></a> packet) Returns true if the packet is a video codec config packet
static boolean	<a href="#"><u>isOnMetadataPacket</u></a> ( <a href="#"><u>AMFPacket</u></a> packet) Returns true if packet is onMetaData or [@setDataFrame, onMetaData] data packet.
static boolean	<a href="#"><u>isVideoCodecConfig</u></a> ( <a href="#"><u>AMFPacket</u></a> packet) Returns true if the packet is a video codec config packet
static boolean	<a href="#"><u>isVideoKeyFrame</u></a> ( <a href="#"><u>AMFPacket</u></a> packet) Returns true if the packet is a video key frame
static boolean	<a href="#"><u>isVideoKeyFrame</u></a> (byte[] buffer) Returns true if the packet is a video key frame
static boolean	<a href="#"><u>isVideoKeyFrame</u></a> (java.nio.ByteBuffer buffer) Returns true if the packet is a video key frame
static boolean	<a href="#"><u>isVideoKeyFrame</u></a> (int[] chunkHeaderValues) Returns true if the packet is a video key frame
static <a href="#"><u>AMFPacket</u></a>	<a href="#"><u>readChunk</u></a> (java.io.DataInput is) Read a packets worth of .flv data from an InputStream and return as an AMFPacket
static <a href="#"><u>AMFPacket</u></a>	<a href="#"><u>readChunk</u></a> (java.io.InputStream is) Read a packets worth of .flv data from an InputStream and return as an AMFPacket

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

### Constructor Summary

public	<a href="#">HTTPUtils()</a>
--------	-----------------------------

### Method Summary

static String	<a href="#">assembleQueryStr</a> (java.util.Map queryMap) Assemble a map of name value pairs into a single query string.
static String	<a href="#">formatDeleteCookie</a> (String name, String path, String domain) Formats a cookie header value that is in the past to delete a cookie
static String	<a href="#">formatSetCookie</a> (String name, String value, int timeoffset, String path, String domain, boolean isSecure) Format a HTTP header Set-Cookie value
static byte[]	<a href="#">HTTPRequestToByteArray</a> (String inUrl, String method, String data, java.util.Map headers) Make a HTTP request and return the result as a byte array
static byte[]	<a href="#">HTTPRequestToByteArray</a> (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	<a href="#">HTTPRequestToFile</a> (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	<a href="#">splitCookie</a> (String str) Breaks Cookies header value into a list of name/value pairs.
static String[]	<a href="#">splitPragmas</a> (String str) Split HTTP Pragma values at commas that separate values.
static java.util.Map	<a href="#">splitQueryStr</a> (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	<a href="#"><code>getBytesAllocation</code></a> (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	<a href="#">onFile</a> (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	<a href="#"><u>IOPerformanceCounter</u></a> ( ) Create an empty performance counter.
--------	---

### Method Summary

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

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

long	<a href="#"><code>incrementMessagesLoss</code></a> (long bytes, long count) Increment bytes-loss by bytes and message count by count.
long	<a href="#"><code>incrementMessagesOut</code></a> () Increment byte-out message count by 1.
long	<a href="#"><code>incrementMessagesOut</code></a> (long bytes) Increment bytes-out by bytes and increment message count by 1.
long	<a href="#"><code>incrementMessagesOut</code></a> (long bytes, long count) Increment bytes-out by bytes and message count by count.
boolean	<a href="#"><code>isDebugLog</code></a> ()
void	<a href="#"><code>setDebugLog</code></a> (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	<a href="#">md5Digest</a>
protected static	<a href="#">md5Lock</a>

### Constructor Summary

public	<a href="#">MD5DigestUtils()</a>
--------	----------------------------------

### Method Summary

static String	<a href="#">generateAuth</a> (String method, String uri, String username, String password, String realm, String nonce) Generate an HTTP authorization response
static String	<a href="#">generateAuth</a> (String a2Hash, String alHash, String realm, String nonce, String qop, String nonceCount, String cnonce) Generate an HTTP authorization response
static String	<a href="#">generateAuth</a> (String method, String uri, String alHash, String realm, String nonce, String qop, String nonceCount, String cnonce) Generate an HTTP authorization response
static String	<a href="#">generateAuth</a> (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	<a href="#">generateAuthHTTP</a> (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	<a href="#">generateHash</a> (String value) Generate MD5 hash
static byte[]	<a href="#">generateHashBytes</a> (byte[] value) Generate MD5 hash
static byte[]	<a href="#">generateHashBytes</a> (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	<a href="#">MediaUtils()</a>
--------	------------------------------

### Method Summary

static String	<a href="#">audioCodecTypeToString</a> (int codecType) Audio codec ID to string.
static String	<a href="#">videoCodecTypeToString</a> (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	<a href="#">NetworkUtils()</a>
--------	--------------------------------

### Method Summary

static boolean	<a href="#">isAddressMulticast(String IpAddress)</a> 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	<a href="#">StringUtils()</a>
--------	-------------------------------

### Method Summary

static boolean	<a href="#">equals</a> (String s1, String s2) Checks if 2 strings are equals, accounting for null cases.
static String	<a href="#">intToHexStr</a> (int value, int strLen)
static String	<a href="#">intToStr</a> (int value, int strLen)
static boolean	<a href="#">isEmpty</a> (String s) Checks if string is empty, handling null String case.
static int	<a href="#">length</a> (String s) Returns string length, handling null String case as length of 0.
static String	<a href="#">longToHexStr</a> (long value, int strLen)
static String	<a href="#">longToStr</a> (long value, int strLen)
static String	<a href="#">stampToString</a> (long stamp) Convert a duration (milliseconds) to a formatted string.
static String	<a href="#">valueOf</a> (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	<a href="#">SystemUtils.ReplaceItem</a> SystemUtils.ReplaceItem
-------	--

### Field Summary

public static final	<a href="#">defaultLocale</a>
public static final	<a href="#">defaultTimeZone</a>
public static final	<a href="#">gmtTimeZone</a>
protected static final	<a href="#">msb0baseTime</a> Value: <b>2085978496000</b>
protected static final	<a href="#">msblbaseTime</a> Value: <b>-2208988800000</b>

### Constructor Summary

public	<a href="#">SystemUtils()</a>
--------	-------------------------------

### Method Summary

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



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

#### Methods inherited from class `java.lang.Object`

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

## Fields

### `defaultTimeZone`

`public static final java.util.TimeZone defaultTimeZone`

### `gmtTimeZone`

`public static final java.util.TimeZone gmtTimeZone`

### `defaultLocale`

`public static final java.util.Locale defaultLocale`

### `msb0baseTime`

`protected static final long msb0baseTime`

Constant value: **2085978496000**

### `msb1baseTime`

`protected static final long msb1baseTime`

Constant value: **-2208988800000**

## Constructors

### `SystemUtils`

`public SystemUtils()`

(continued from last page)

## Methods

### expandEnvironmentVariables

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

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

**Parameters:**

inValue - string with properties

**Returns:**

expanded string

### expandEnvironmentVariables

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

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

**Parameters:**

inValue - string with properties

valueMap - name/value pair map

**Returns:**

expanded string

### getMACAddress

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

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

**Returns:**

MAC address of localhost interface

### addBouncyCastleSecurityProvider

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

Load bouncy castle providers, Internal use.

### toNTPTime

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

Convert a timecode value in milliseconds to NTP time

**Parameters:**

t - timecode value in milliseconds

**Returns:**

NTP time

(continued from last page)

## **getCpuTime**

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

Get CPU time in nanoseconds.

---

## **getUserTime**

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

Get user time in nanoseconds.

---

## **getSystemTime**

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

Get system time in nanoseconds.

## com.wowza.util Class SystemUtils.ReplaceItem

java.lang.Object

└─com.wowza.util.SystemUtils.ReplaceItem

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

### Field Summary

public	<a href="#">end</a>
public	<a href="#">newValue</a>
public	<a href="#">start</a>

### Constructor Summary

public	<a href="#">SystemUtils.ReplaceItem</a> (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	<a href="#">URLUtils()</a>
--------	----------------------------

### Method Summary

static String	<a href="#">appendParamsToUrl</a> (String url, String params) Convenience call to add parameters to a url.
static String	<a href="#">decodeValue</a> (String val) Wrapper for URLDecoder.decode(val, "UTF-8");
static String	<a href="#">encodeValue</a> (String val) Wrapper for URLEncoder.encode(val, "UTF-8");
static String	<a href="#">getParamValue</a> (java.util.Map params, String key) Helper funtion to get single value from multiple value parameter Map
static java.util.Map	<a href="#">parseQueryStr</a> (String queryString, boolean doDecode) Parse query string part of url into Map of Lists (to support multiple values) of query parameters
static String	<a href="#">pathToFileURL</a> (String basePath) Convert a path to a url (file://[path])
static String	<a href="#">urlToId</a> (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&param2=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	<a href="#">XMLUtils()</a>
--------	----------------------------

### Method Summary

static org.w3c.dom.Node	<a href="#">getNodeByTagName</a> (org.w3c.dom.Element node, String name) Return a child Node by tag name.
static String	<a href="#">getNodeValue</a> (org.w3c.dom.Node node) Return the text value of a node.
static String	<a href="#">getNodeValueByTagName</a> (org.w3c.dom.Element node, String name) Get a child Node value by tag name.
static boolean	<a href="#">getXMLPropertyBool</a> (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	<a href="#">getXMLPropertyDouble</a> (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	<a href="#">getXMLPropertyExists</a> (javax.xml.xpath.XPath xpath, String xpathStr, org.w3c.dom.Element root)
static int	<a href="#">getXMLPropertyInt</a> (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	<a href="#">getXMLPropertyIntSize</a> (javax.xml.xpath.XPath xpath, String xpathStr, org.w3c.dom.Element root, int defaultVal)
static long	<a href="#">getXMLPropertyLong</a> (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	<a href="#">getXMLPropertyLongSize</a> (javax.xml.xpath.XPath xpath, String xpathStr, org.w3c.dom.Element root, long defaultVal)

static String	<a href="#">getXMLPropertyStr</a> ( 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	<a href="#">getXMLPropertyStr</a> ( 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	<a href="#">loadConfigProperties</a> (org.w3c.dom.Element root, String propertiesXPath, <a href="#">WMSProperties</a> properties) Loads <Properties> elemnt by xpath into properties object.
static void	<a href="#">loadConfigProperties</a> (org.w3c.dom.NodeList resultList, <a href="#">WMSProperties</a> properties) Given a nodeList load children as properties.
static javax.xml.xpath.XPath Factory	<a href="#">newXPathFactory</a> () 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	<a href="#">AMF3Utils()</a>
--------	-----------------------------

#### Method Summary

static java.util.Date	<a href="#">deserializeDate</a> (java.nio.ByteBuffer data) Deserialize date type
static int	<a href="#">deserializeInt</a> (java.nio.ByteBuffer data) Deserialize int
static String	<a href="#">deserializeString</a> (java.nio.ByteBuffer data) Deserialize string
static String	<a href="#">deserializeString</a> (java.nio.ByteBuffer data, <a href="#">AMFDataContextDeserialize</a> context) Deserialize string
static String	<a href="#">deserializeString</a> (java.nio.ByteBuffer data, int utflen) Deserialize string
static int	<a href="#">serializeDate</a> (java.io.DataOutputStream out, java.util.Date date) Serialize a date object
static int	<a href="#">serializeInt</a> (java.io.DataOutputStream out, int val) Serialize int value
static int	<a href="#">serializeString</a> (java.io.DataOutputStream out, String str) Serialize a string value
static int	<a href="#">serializeStringNoLength</a> (java.io.DataOutputStream out, String str) Serialize string but do not write the length
static void	<a href="#">serializeZeroLengthString</a> (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	<a href="#">AMF_LEVEL0</a> Value: <b>0</b>
public static final	<a href="#">AMF_LEVEL3</a> Value: <b>3</b>
public static final	<a href="#">DATA_TYPE_AMF3</a> Value: <b>17</b>
public static final	<a href="#">DATA_TYPE_AMF3_ARRAY</a> Value: <b>9</b>
public static final	<a href="#">DATA_TYPE_AMF3_BOOLEAN_FALSE</a> Value: <b>2</b>
public static final	<a href="#">DATA_TYPE_AMF3_BOOLEAN_TRUE</a> Value: <b>3</b>
public static final	<a href="#">DATA_TYPE_AMF3_BYTEARRAY</a> Value: <b>12</b>
public static final	<a href="#">DATA_TYPE_AMF3_DATE</a> Value: <b>8</b>
public static final	<a href="#">DATA_TYPE_AMF3_INTEGER</a> Value: <b>4</b>
public static final	<a href="#">DATA_TYPE_AMF3_NULL</a> Value: <b>1</b>

public static final	<a href="#"><u>DATA_TYPE_AMF3_NUMBER</u></a> Value: <b>5</b>
public static final	<a href="#"><u>DATA_TYPE_AMF3_OBJECT</u></a> Value: <b>10</b>
public static final	<a href="#"><u>DATA_TYPE_AMF3_STRING</u></a> Value: <b>6</b>
public static final	<a href="#"><u>DATA_TYPE_AMF3_UNDEFINED</u></a> Value: <b>0</b>
public static final	<a href="#"><u>DATA_TYPE_AMF3_XML_LEGACY</u></a> Value: <b>7</b>
public static final	<a href="#"><u>DATA_TYPE_AMF3_XML_TOP</u></a> Value: <b>11</b>
public static final	<a href="#"><u>DATA_TYPE_ARRAY</u></a> Value: <b>10</b>
public static final	<a href="#"><u>DATA_TYPE_AS_OBJECT</u></a> Value: <b>13</b>
public static final	<a href="#"><u>DATA_TYPE_BOOLEAN</u></a> Value: <b>1</b>
public static final	<a href="#"><u>DATA_TYPE_BYTEARRAY</u></a> Value: <b>33</b>
public static final	<a href="#"><u>DATA_TYPE_CUSTOM_CLASS</u></a> Value: <b>16</b>
public static final	<a href="#"><u>DATA_TYPE_DATE</u></a> Value: <b>11</b>
public static final	<a href="#"><u>DATA_TYPE_INTEGER</u></a> Value: <b>32</b>
public static final	<a href="#"><u>DATA_TYPE_LONG_STRING</u></a> Value: <b>12</b>
public static final	<a href="#"><u>DATA_TYPE_MIXED_ARRAY</u></a> Value: <b>8</b>
public static final	<a href="#"><u>DATA_TYPE_MOVIE_CLIP</u></a> Value: <b>4</b>

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

## Constructor Summary

public	<a href="#">AMFData()</a>
--------	---------------------------

## Method Summary

static <a href="#">AMFDataContextDeserialize</a>	<a href="#">createContextDeserialize()</a> Create an AMF3 deserialization context
---	--

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

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

#### Methods inherited from class java.lang.Object

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

## Fields

### DATA\_TYPE\_UNKNOWN

```
public static final byte DATA_TYPE_UNKNOWN
```

Constant value: **-1**

### DATA\_TYPE\_NUMBER

```
public static final byte DATA_TYPE_NUMBER
```

Constant value: **0**

### DATA\_TYPE\_BOOLEAN

```
public static final byte DATA_TYPE_BOOLEAN
```

Constant value: **1**

### DATA\_TYPE\_STRING

```
public static final byte DATA_TYPE_STRING
```



(continued from last page)

Constant value: **2**

---

## DATA\_TYPE\_OBJECT

```
public static final byte DATA_TYPE_OBJECT
```

Constant value: **3**

---

## DATA\_TYPE\_MOVIE\_CLIP

```
public static final byte DATA_TYPE_MOVIE_CLIP
```

Constant value: **4**

---

## DATA\_TYPE\_NULL

```
public static final byte DATA_TYPE_NULL
```

Constant value: **5**

---

## DATA\_TYPE\_UNDEFINED

```
public static final byte DATA_TYPE_UNDEFINED
```

Constant value: **6**

---

## DATA\_TYPE\_REFERENCE\_OBJECT

```
public static final byte DATA_TYPE_REFERENCE_OBJECT
```

Constant value: **7**

---

## DATA\_TYPE\_MIXED\_ARRAY

```
public static final byte DATA_TYPE_MIXED_ARRAY
```

Constant value: **8**

---

## DATA\_TYPE\_OBJECT\_END

```
public static final byte DATA_TYPE_OBJECT_END
```

Constant value: **9**

---

## DATA\_TYPE\_ARRAY

```
public static final byte DATA_TYPE_ARRAY
```

Constant value: **10**

(continued from last page)

---

## DATA\_TYPE\_DATE

```
public static final byte DATA_TYPE_DATE
```

Constant value: **11**

---

## DATA\_TYPE\_LONG\_STRING

```
public static final byte DATA_TYPE_LONG_STRING
```

Constant value: **12**

---

## DATA\_TYPE\_AS\_OBJECT

```
public static final byte DATA_TYPE_AS_OBJECT
```

Constant value: **13**

---

## DATA\_TYPE\_RECORDSET

```
public static final byte DATA_TYPE_RECORDSET
```

Constant value: **14**

---

## DATA\_TYPE\_XML

```
public static final byte DATA_TYPE_XML
```

Constant value: **15**

---

## DATA\_TYPE\_CUSTOM\_CLASS

```
public static final byte DATA_TYPE_CUSTOM_CLASS
```

Constant value: **16**

---

## DATA\_TYPE\_AMF3

```
public static final byte DATA_TYPE_AMF3
```

Constant value: **17**

---

## DATA\_TYPE\_INTEGER

```
public static final byte DATA_TYPE_INTEGER
```

Constant value: **32**

---

## DATA\_TYPE\_BYTEARRAY

```
public static final byte DATA_TYPE_BYTEARRAY
```

---

(continued from last page)

Constant value: **33**

---

## DATA\_TYPE\_XML\_TOP

```
public static final byte DATA_TYPE_XML_TOP
```

Constant value: **34**

---

## DATA\_TYPE\_AMF3\_UNDEFINED

```
public static final byte DATA_TYPE_AMF3_UNDEFINED
```

Constant value: **0**

---

## DATA\_TYPE\_AMF3\_NULL

```
public static final byte DATA_TYPE_AMF3_NULL
```

Constant value: **1**

---

## DATA\_TYPE\_AMF3\_BOOLEAN\_FALSE

```
public static final byte DATA_TYPE_AMF3_BOOLEAN_FALSE
```

Constant value: **2**

---

## DATA\_TYPE\_AMF3\_BOOLEAN\_TRUE

```
public static final byte DATA_TYPE_AMF3_BOOLEAN_TRUE
```

Constant value: **3**

---

## DATA\_TYPE\_AMF3\_INTEGER

```
public static final byte DATA_TYPE_AMF3_INTEGER
```

Constant value: **4**

---

## DATA\_TYPE\_AMF3\_NUMBER

```
public static final byte DATA_TYPE_AMF3_NUMBER
```

Constant value: **5**

---

## DATA\_TYPE\_AMF3\_STRING

```
public static final byte DATA_TYPE_AMF3_STRING
```

Constant value: **6**

---

## DATA\_TYPE\_AMF3\_XML\_LEGACY

public static final byte **DATA\_TYPE\_AMF3\_XML\_LEGACY**

Constant value: **7**

---

## DATA\_TYPE\_AMF3\_DATE

public static final byte **DATA\_TYPE\_AMF3\_DATE**

Constant value: **8**

---

## DATA\_TYPE\_AMF3\_ARRAY

public static final byte **DATA\_TYPE\_AMF3\_ARRAY**

Constant value: **9**

---

## DATA\_TYPE\_AMF3\_OBJECT

public static final byte **DATA\_TYPE\_AMF3\_OBJECT**

Constant value: **10**

---

## DATA\_TYPE\_AMF3\_XML\_TOP

public static final byte **DATA\_TYPE\_AMF3\_XML\_TOP**

Constant value: **11**

---

## DATA\_TYPE\_AMF3\_BYTEARRAY

public static final byte **DATA\_TYPE\_AMF3\_BYTEARRAY**

Constant value: **12**

---

## AMF\_LEVEL0

public static final byte **AMF\_LEVEL0**

Constant value: **0**

---

## AMF\_LEVEL3

public static final byte **AMF\_LEVEL3**

Constant value: **3**

---

(continued from last page)

## MILLS\_PER\_HOUR

```
public static final int MILLS_PER_HOUR
```

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

## type

```
protected int type
```

## Constructors

### AMFData

```
public AMFData()
```

## Methods

### getType

```
public int getType()
```

Returns the data type for this object

**Returns:**

object type DATA\_TYPE\_\*

### setType

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

Sets the the data type for this object

**Parameters:**

type - type DATA\_TYPE\_\*

### testNextByte

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

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

**Parameters:**

data - binary data being deserialized  
test - value being tested

**Returns:**

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

### peekByte

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

---

(continued from last page)

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

**Parameters:**

data - binary data being deserialized

**Returns:**

next byte in buffer

---

## skipByte

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

Skip forward one byte in the byte buffer

**Parameters:**

data - binary data being deserialized

**Returns:**

next byte in buffer

---

## isObjStart

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

Returns true if next byte in data is object start

**Parameters:**

data - binary data being deserialized  
context - deserialization context

**Returns:**

Returns true if next byte in data is object start

---

## isAMF3Start

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

Returns true if next byte in data is AMF3 start

**Parameters:**

data - binary data being deserialized  
context - deserialization context

**Returns:**

Returns true if next byte in data is object start

---

## isArrayStart

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

Returns true if next byte in data is an array start

**Parameters:**

data - binary data being deserialized  
context - deserialization context

---

(continued from last page)

**Returns:**

Returns true if next byte in data is and array start

---

## isMixedArrayStart

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

Returns true if next byte in data is mixed array start

**Parameters:**

data - binary data being deserialized  
context - deserialization context

**Returns:**

Returns true if next byte in data is mixed array start

---

## isObjEnd

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

Returns true if next byte is object end

**Parameters:**

data - binary data being deserialized  
context - deserialization context

**Returns:**

Returns true if next byte in object end

---

## isByteArrayStart

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

Returns true if next byte is ByteArray start

**Parameters:**

data - binary data being deserialized  
context - deserialization context

**Returns:**

true if next byte in object end

---

## getReference

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

Get AMF3 object reference

**Parameters:**

data - binary data being deserialized  
context - deserialization context

**Returns:**

null if not a reference else referenced object

## deserializeInnerObject

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

Deserialize next item

**Parameters:**

data - binary data being deserialized  
context - deserialization context

**Returns:**

deserialized object

**Throws:**

IOException

---

## createContextSerialize

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

Create an AMF3 serialization context

**Returns:**

AMF3 serialization context

---

## createContextSerialize

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

Create an AMF3 serialization context

**Parameters:**

objectEncoding - object encoding level (see AMF\_LEVEL\*)

**Returns:**

AMF3 serialization context

---

## createContextDeserialize

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

Create an AMF3 deserialization context

**Returns:**

AMF3 deserialization context

---

## createContextDeserialize

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

Create an AMF3 deserialization context

**Parameters:**

objectEncoding - object encoding level (see AMF\_LEVEL\*)

**Returns:**



(continued from last page)

## AMF3 deserialization context

---

**serialize**

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

Serialize object to output stream

**Parameters:**

out - Output stream

---

**serialize**

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

Serialize object to output stream

**Parameters:**

out - Output stream

objectEncoding - object encoding level (see AMF\_LEVEL\*)

---

**serialize**

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

Serialize object to output stream

**Parameters:**

out - Output stream

context - serialization context used by AMF3

---

**serialize**

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

Serial object to byte array

**Returns:**

serialized byte array

---

**serialize**

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

Serial object to byte array

**Parameters:**

objectEncoding - object encoding level (see AMF\_LEVEL\*)

**Returns:**

serialized byte array

---

**serialize**

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

Serial object to byte array

---

---

(continued from last page)

**Parameters:**

context - serialization context used by AMF3

**Returns:**

serialized byte array

---

## deserialize

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

Deserialize data in byte buffer

**Parameters:**

data - binary data

---

## deserialize

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

Deserialize data in byte buffer

**Parameters:**

data - binary data

context - deserialization context used by AMF3

---

## getValue

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

Convert object to Java native class

**Returns:**

java native class

---

## triggerAMF3Switch

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

Return true if the object is serialized differently in AMF3

**Parameters:**

data - AMF object

**Returns:**

true if the object is serialized differently in AMF3

---

## com.wowza.wms.amf Class AMFDataArray

```
java.lang.Object
|
+-com.wowza.wms.amf.AMFData
|   |
|   +-com.wowza.wms.amf.AMFDataArray
```

```
public class AMFDataArray
extends AMFData
```

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

### Create Array of Strings

```
AMFDataArray amfDataArray = new AMFDataArray();

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

### Iterate Items In Array

```
AMFDataArray amfDataArray;

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

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

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

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

[AMF\\_LEVEL0](#), [AMF\\_LEVEL3](#), [DATA\\_TYPE\\_AMF3](#), [DATA\\_TYPE\\_AMF3\\_ARRAY](#), [DATA\\_TYPE\\_AMF3\\_BOOLEAN\\_FALSE](#),  
[DATA\\_TYPE\\_AMF3\\_BOOLEAN\\_TRUE](#), [DATA\\_TYPE\\_AMF3\\_BYTEARRAY](#), [DATA\\_TYPE\\_AMF3\\_DATE](#),  
[DATA\\_TYPE\\_AMF3\\_INTEGER](#), [DATA\\_TYPE\\_AMF3\\_NULL](#), [DATA\\_TYPE\\_AMF3\\_NUMBER](#), [DATA\\_TYPE\\_AMF3\\_OBJECT](#),  
[DATA\\_TYPE\\_AMF3\\_STRING](#), [DATA\\_TYPE\\_AMF3\\_UNDEFINED](#), [DATA\\_TYPE\\_AMF3\\_XML\\_LEGACY](#),  
[DATA\\_TYPE\\_AMF3\\_XML\\_TOP](#), [DATA\\_TYPE\\_ARRAY](#), [DATA\\_TYPE\\_AS\\_OBJECT](#), [DATA\\_TYPE\\_BOOLEAN](#),  
[DATA\\_TYPE\\_BYTEARRAY](#), [DATA\\_TYPE\\_CUSTOM\\_CLASS](#), [DATA\\_TYPE\\_DATE](#), [DATA\\_TYPE\\_INTEGER](#),  
[DATA\\_TYPE\\_LONG\\_STRING](#), [DATA\\_TYPE\\_MIXED\\_ARRAY](#), [DATA\\_TYPE\\_MOVIE\\_CLIP](#), [DATA\\_TYPE\\_NULL](#),  
[DATA\\_TYPE\\_NUMBER](#), [DATA\\_TYPE\\_OBJECT](#), [DATA\\_TYPE\\_OBJECT\\_END](#), [DATA\\_TYPE\\_RECORDSET](#),  
[DATA\\_TYPE\\_REFERENCE\\_OBJECT](#), [DATA\\_TYPE\\_STRING](#), [DATA\\_TYPE\\_UNDEFINED](#), [DATA\\_TYPE\\_UNKNOWN](#),  
[DATA\\_TYPE\\_XML](#), [DATA\\_TYPE\\_XML\\_TOP](#), [MILLS\\_PER\\_HOUR](#), [type](#)

## Constructor Summary

public	<a href="#">AMFDataArray</a> () Create empty AMFDataArray object
public	<a href="#">AMFDataArray</a> (byte[] data) Deserialize entire data array and create AMFDataArray object
public	<a href="#">AMFDataArray</a> (byte[] data, int offset, int size) Deserialize data array starting at offset for size bytes and create AMFDataArray object
public	<a href="#">AMFDataArray</a> (java.nio.ByteBuffer data) Deserialize entire data array and create AMFDataArray object
public	<a href="#">AMFDataArray</a> (java.nio.ByteBuffer data, <a href="#">AMFDataContextDeserialize</a> context)

## Method Summary

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

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

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

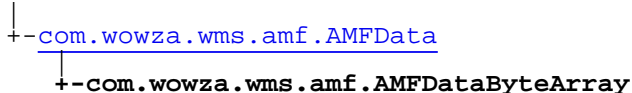
```
public String toString()
```

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	<a href="#">AMFDataByteArray()</a> Create empty AMFDataByteArray object
public	<a href="#">AMFDataByteArray(byte[] data)</a> Deserialize entire byte array and create AMFDataByteArray object.
public	<a href="#">AMFDataByteArray(byte[] data, int offset, int size)</a> Deserialize data array starting at offset for size bytes and create AMFDataByteArray object.
public	<a href="#">AMFDataByteArray(java.nio.ByteBuffer data)</a> Deserialize entire data array and create AMFDataByteArray object.
public	<a href="#">AMFDataByteArray(java.nio.ByteBuffer data, <a href="#">AMFDataContextDeserialize</a> context)</a> Deserialize entire data array and create AMFDataByteArray object.

#### Method Summary

int	<a href="#">compress()</a> Compress the internal buffer using the ZLIB compression library
int	<a href="#">decompress()</a> Decompress the internal buffer using the ZLIB compression library
void	<a href="#">deserialize(java.nio.ByteBuffer data)</a>



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

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

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

#### Methods inherited from class java.lang.Object

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

## Constructors

(continued from last page)

## AMFDataByteArray

```
public AMFDataByteArray( )
```

Create empty AMFDataByteArray object

---

## AMFDataByteArray

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

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

**Parameters:**

data - binary data

---

## AMFDataByteArray

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

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

**Parameters:**

data - binary data

offset - starting offset into data

size - size of data to deserialize

---

## AMFDataByteArray

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

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

**Parameters:**

data - binary data

---

## AMFDataByteArray

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

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

**Parameters:**

data - binary data

context - deserialization context (used for AMF3 decoding)

## Methods

(continued from last page)

## size

```
public int size()
```

Returns the number of bytes in the byte array

**Returns:**

number of bytes in the array

---

## toArray

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

Returns the underlying data buffer (not a copy)

**Returns:**

data buffer (not a copy)

---

## toByteBuffer

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

Wraps the underlying data buffer with a ByteBuffer object.

**Returns:**

byte[] wrapped as ByteBuffer

---

## wrap

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

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

**Parameters:**

data - byte[] data

**Returns:**

wrapped byte[]

---

## wrap

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

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

**Parameters:**

data - ByteBuffer data

**Returns:**

wrapped ByteBuffer

---

## getValue

```
public Object getValue()
```

Returns the underlying byte[] data buffer

## deserialize

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

Deserialize data in byte buffer

---

## deserialize

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

Deserialize data in byte buffer

---

## serialize

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

Serialize object to output stream

---

## serialize

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

Serialize object to output stream

---

## serialize

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

Serialize object to output stream

---

## serialize

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

Serial object to byte array

---

## serialize

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

Serial object to byte array

---

## serialize

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

Serial object to byte array

---

## compress

```
public int compress()
```

Compress the internal buffer using the ZLIB compression library

**Returns:**

---

(continued from last page)

size of compressed buffer

---

## decompress

```
public int decompress()
```

Decompress the internal buffer using the ZLIB compression library

**Returns:**

size of decompressed buffer

---

## toString

```
public String toString()
```

Return object as formatted string

## com.wowza.wms.amf

### Class AMFDataContextDeserialize

java.lang.Object

└─com.wowza.wms.amf.AMFDataContextDeserialize

public class **AMFDataContextDeserialize**  
extends Object

AMF context used for deserialization

#### Constructor Summary

public	<a href="#">AMFDataContextDeserialize()</a> Constructor
public	<a href="#">AMFDataContextDeserialize(int objectEncoding)</a> Constructor with encoding

#### Method Summary

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

boolean	<a href="#"><code>isIntData()</code></a> Internal use, get int data
void	<a href="#"><code>setIntData(int intData)</code></a> Internal use, set int data
void	<a href="#"><code>setObjectEncoding(int objectEncoding)</code></a> 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	<a href="#">AMFDataContextSerialize()</a> Constructor
public	<a href="#">AMFDataContextSerialize(int objectEncoding)</a> Constructor with object encoding, see AMFData.AMF_LEVEL*

#### Method Summary

int	<a href="#">getObjectEncoding()</a> Get object encoding, see AMFData.AMF_LEVEL*
int	<a href="#">getObjectReference(Object obj)</a> Get index of object in object cache
int	<a href="#">getStringReference(String str)</a> Get index of string item in string cache
int	<a href="#">getTargetEncoding()</a> Get target encoding, see AMFData.AMF_LEVEL*
int	<a href="#">getTraitReference(AMFDataTrait obj)</a> Get index of trait object in trait cache
boolean	<a href="#">isAMF0()</a> Is context AMF0
boolean	<a href="#">isAMF3()</a> Is context AMF3
void	<a href="#">setObjectEncoding(int objectEncoding)</a> Set object encoding, see AMFData.AMF_LEVEL*
void	<a href="#">setTargetEncoding(int targetEncoding)</a> Set target encoding, , see AMFData.AMF_LEVEL*
void	<a href="#">writeString(java.io.DataOutputStream out, String str)</a> 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	<a href="#">DATEFORMAT</a> Value: <b>EEE, dd MMM yyyy HH:mm:ss</b>
protected	<a href="#">fastDateFormat</a>

### 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	<a href="#">AMFDataItem</a> () Construct AMF type DATA_TYPE_NULL object
public	<a href="#">AMFDataItem</a> (String value) Construct AMF type DATA_TYPE_STRING object
public	<a href="#">AMFDataItem</a> (int value) Construct AMF type DATA_TYPE_NUMBER object
public	<a href="#">AMFDataItem</a> (long value) Construct AMF type DATA_TYPE_NUMBER object
public	<a href="#">AMFDataItem</a> (double value) Construct AMF type DATA_TYPE_NUMBER object
public	<a href="#">AMFDataItem</a> (boolean value) Construct AMF type DATA_TYPE_BOOLEAN
public	<a href="#">AMFDataItem</a> (java.util.Date value) Construct AMF type DATA_TYPE_DATE
public	<a href="#">AMFDataItem</a> (byte[] data) Deserialize entire data array and create AMFDataItem object
public	<a href="#">AMFDataItem</a> (byte[] data, int offset, int size) Deserialize data array starting at offset for size bytes and create AMFDataItem object
public	<a href="#">AMFDataItem</a> (java.nio.ByteBuffer data) Deserialize entire data array and create AMFDataItem object
public	<a href="#">AMFDataItem</a> (java.nio.ByteBuffer data, <a href="#">AMFDataContextDeserialize</a> context)



## Method Summary

boolean	<a href="#">booleanValue()</a> Return object as boolean.
byte	<a href="#">byteValue()</a> Return object as byte.
java.util.Date	<a href="#">dateValue()</a> Return object as Date.
void	<a href="#">deserialize()</a> (java.nio.ByteBuffer data)
void	<a href="#">deserialize()</a> (java.nio.ByteBuffer data, <a href="#">AMFDataContextDeserialize</a> context)
double	<a href="#">doubleValue()</a> Return object as double.
float	<a href="#">floatValue()</a> Return object as float.
Object	<a href="#">getValue()</a> Return value as Java class
int	<a href="#">intValue()</a> Return object as int.
long	<a href="#">longValue()</a> Return object as long.
byte[]	<a href="#">serialize()</a>
byte[]	<a href="#">serialize()</a> ( <a href="#">AMFDataContextSerialize</a> context)
void	<a href="#">serialize()</a> (java.io.DataOutputStream out)
void	<a href="#">serialize()</a> (java.io.DataOutputStream out, <a href="#">AMFDataContextSerialize</a> context)
void	<a href="#">serialize()</a> (java.io.DataOutputStream out, int objectEncoding)
byte[]	<a href="#">serialize()</a> (int objectEncoding)
short	<a href="#">shortValue()</a> Return object as short.
String	<a href="#">toString()</a> 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)
```

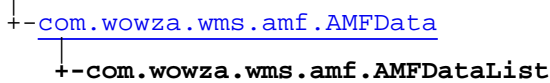
Serial object to byte array

---

## 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	<a href="#">AMFDataList</a> () Create empty AMFDataList object
public	<a href="#">AMFDataList</a> (byte[] data) Deserialize entire data array and create AMFDataList object
public	<a href="#">AMFDataList</a> (byte[] data, int offset, int size) Deserialize data array starting at offset for size bytes and create AMFDataList object
public	<a href="#">AMFDataList</a> (java.nio.ByteBuffer data) Deserialize entire data array and create AMFDataList object
public	<a href="#">AMFDataList</a> (java.nio.ByteBuffer data, <a href="#">AMFDataContextDeserialize</a> context)

### Method Summary

void	<a href="#">add</a> ( <a href="#">AMFData</a> data) Append a new item onto the array
------	---

void	<a href="#">add</a> (boolean data) Append a boolean (will be wrapped in an AMFDataItem object)
void	<a href="#">add</a> (java.util.Date data) Append a date (will be wrapped in an AMFDataItem object)
void	<a href="#">add</a> (double data) Append a double (will be wrapped in an AMFDataItem object)
void	<a href="#">add</a> (int data) Append a int (will be wrapped in an AMFDataItem object)
void	<a href="#">add</a> (int index, <a href="#">AMFData</a> data) Insert an item into the array
void	<a href="#">add</a> (int index, boolean data) Insert a boolean value (will be wrapped in an AMFDataItem object)
void	<a href="#">add</a> (int index, java.util.Date data) Insert a date value (will be wrapped in an AMFDataItem object)
void	<a href="#">add</a> (int index, double data) Insert a double value (will be wrapped in an AMFDataItem object)
void	<a href="#">add</a> (int index, int data) Insert a int value (will be wrapped in an AMFDataItem object)
void	<a href="#">add</a> (int index, long data) Insert a long value (will be wrapped in an AMFDataItem object)
void	<a href="#">add</a> (int index, String data) Insert a string value (will be wrapped in an AMFDataItem object)
void	<a href="#">add</a> (long data) Append a long (will be wrapped in an AMFDataItem object)
void	<a href="#">add</a> (String data) Append a string (will be wrapped in an AMFDataItem object)
void	<a href="#">deserialize</a> (java.nio.ByteBuffer data)
void	<a href="#">deserialize</a> (java.nio.ByteBuffer data, <a href="#">AMFDataContextDeserialize</a> context)
<a href="#">AMFData</a>	<a href="#">get</a> (int index) Get item at index
boolean	<a href="#">getBoolean</a> (int index) Get item at index return as boolean
byte	<a href="#">getBytes</a> (int index) Get item at index return as byte
java.util.Date	<a href="#">getDate</a> (int index) Get item at index return as Date
double	<a href="#">getDouble</a> (int index) Get item at index return as double



float	<a href="#"><code>getFloat(int index)</code></a> Get item at index return as float
int	<a href="#"><code>getInt(int index)</code></a> Get item at index return as int
long	<a href="#"><code>getLong(int index)</code></a> Get item at index return as long
<a href="#"><code>AMFDataObj</code></a>	<a href="#"><code>getObject(int index)</code></a> Get item at index return as AMFDataObj
short	<a href="#"><code>getShort(int index)</code></a> Get item at index return as short
String	<a href="#"><code>getString(int index)</code></a> Get item at index return as String
int	<a href="#"><code>getType(int index)</code></a> Get type of item at index.
Object	<a href="#"><code>getValue()</code></a>
<a href="#"><code>AMFData</code></a>	<a href="#"><code>remove(int index)</code></a> Remove an element from the AMFDataList object
byte[]	<a href="#"><code>serialize()</code></a>
byte[]	<a href="#"><code>serialize(AMFDataContextSerialize context)</code></a>
byte[]	<a href="#"><code>serialize(AMFDataContextSerialize context, byte[] prepend)</code></a>
void	<a href="#"><code>serialize(java.io.DataOutputStream out)</code></a>
void	<a href="#"><code>serialize(java.io.DataOutputStream out, AMFDataContextSerialize context)</code></a>
void	<a href="#"><code>serialize(java.io.DataOutputStream out, AMFDataContextSerialize context, byte[] prepend)</code></a>
void	<a href="#"><code>serialize(java.io.DataOutputStream out, int objectEncoding)</code></a>
byte[]	<a href="#"><code>serialize(int objectEncoding)</code></a>
void	<a href="#"><code>set(int index, AMFData data)</code></a> Set an array item
void	<a href="#"><code>set(int index, boolean data)</code></a> Set an boolean value (will be wrapped in an AMFDataItem object)
void	<a href="#"><code>set(int index, java.util.Date data)</code></a> Set an date value (will be wrapped in an AMFDataItem object)
void	<a href="#"><code>set(int index, double data)</code></a> Set an double value (will be wrapped in an AMFDataItem object)

void	<a href="#">set</a> (int index, int data) Set an int value (will be wrapped in an AMFDataItem object)
void	<a href="#">set</a> (int index, long data) Set an long value (will be wrapped in an AMFDataItem object)
void	<a href="#">set</a> (int index, String data) Set an string value (will be wrapped in an AMFDataItem object)
int	<a href="#">size</a> () Returns the number of items in array
String	<a href="#">toString</a> () 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	<a href="#">AMFDataMixedArray()</a> Create empty AMFDataMixedArray object
public	<a href="#">AMFDataMixedArray</a> (byte[] data) Deserialize entire data array and create AMFDataMixedArray object
public	<a href="#">AMFDataMixedArray</a> (byte[] data, int offset, int size) Deserialize data array starting at offset for size bytes and create AMFDataMixedArray object
public	<a href="#">AMFDataMixedArray</a> (java.nio.ByteBuffer data) Deserialize entire data array and create AMFDataMixedArray object
public	<a href="#">AMFDataMixedArray</a> (java.nio.ByteBuffer data, <a href="#">AMFDataContextDeserialize</a> context)

## Method Summary

void	<a href="#">deserialize</a> (java.nio.ByteBuffer data)
void	<a href="#">deserialize</a> (java.nio.ByteBuffer data, <a href="#">AMFDataContextDeserialize</a> context)
void	<a href="#">serialize</a> (java.io.DataOutputStream out)
void	<a href="#">serialize</a> (java.io.DataOutputStream out, <a href="#">AMFDataContextSerialize</a> context)
void	<a href="#">serialize</a> (java.io.DataOutputStream out, int objectEncoding)
String	<a href="#">toString</a> () 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	<a href="#">DECODE_OBJ_REF</a> Value: <b>1</b>
public static final	<a href="#">DECODE_TRAITS</a> Value: <b>4</b>
public static final	<a href="#">DECODE_TRAITS_EXT</a> Value: <b>3</b>
public static final	<a href="#">DECODE_TRAITS_REF</a> Value: <b>2</b>
public static final	<a href="#">DECODE_UNDEFINED</a> Value: <b>0</b>
protected	<a href="#">members</a>
protected	<a href="#">order</a>
protected	<a href="#">trait</a>

### 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	<a href="#">AMFDataObj</a> () Create empty AMFDataObj object
public	<a href="#">AMFDataObj</a> (byte[] data) Deserialize entire data array and create AMFDataObj object
public	<a href="#">AMFDataObj</a> (byte[] data, int offset, int size) Deserialize data array starting at offset for size bytes and create AMFDataObj object
public	<a href="#">AMFDataObj</a> (java.nio.ByteBuffer data) Deserialize entire data array and create AMFDataObj object
public	<a href="#">AMFDataObj</a> (java.nio.ByteBuffer data, <a href="#">AMFDataContextDeserialize</a> context)

## Method Summary

boolean	<a href="#"><u>containsKey</u></a> (String name)
void	<a href="#"><u>deserialize</u></a> (java.nio.ByteBuffer data)
void	<a href="#"><u>deserialize</u></a> (java.nio.ByteBuffer data, <a href="#"><u>AMFDataContextDeserialize</u></a> context)
<a href="#"><u>AMFData</u></a>	<a href="#"><u>get</u></a> (int index)
<a href="#"><u>AMFData</u></a>	<a href="#"><u>get</u></a> (String name)
boolean	<a href="#"><u>getBoolean</u></a> (int index)
boolean	<a href="#"><u>getBoolean</u></a> (String name)
byte	<a href="#"><u>getBytes</u></a> (int index)
byte	<a href="#"><u>getBytes</u></a> (String name)
String	<a href="#"><u>getClassName</u></a> ()
java.util.Date	<a href="#"><u>getDate</u></a> (int index)
java.util.Date	<a href="#"><u>getDate</u></a> (String name)
double	<a href="#"><u>getDouble</u></a> (int index)
double	<a href="#"><u>getDouble</u></a> (String name)
float	<a href="#"><u>getFloat</u></a> (int index)
float	<a href="#"><u>getFloat</u></a> (String name)
int	<a href="#"><u>getInt</u></a> (int index)
int	<a href="#"><u>getInt</u></a> (String name)
String	<a href="#"><u>getKey</u></a> (int index)
java.util.List	<a href="#"><u>getKeys</u></a> ()
long	<a href="#"><u>getLong</u></a> (int index)
long	<a href="#"><u>getLong</u></a> (String name)
<a href="#"><u>AMFDataObj</u></a>	<a href="#"><u>getObject</u></a> (int index)

<a href="#">AMFDataObj</a>	<a href="#">getObject</a> (String name)
short	<a href="#">getShort</a> (int index)
short	<a href="#">getShort</a> (String name)
String	<a href="#">getString</a> (int index)
String	<a href="#">getString</a> (String name)
<a href="#">AMFDataTrait</a>	<a href="#">getTrait</a> ()
Object	<a href="#">getValue</a> ()
void	<a href="#">put</a> (String name, <a href="#">AMFData</a> data)
void	<a href="#">put</a> (String name, boolean data)
void	<a href="#">put</a> (String name, java.util.Date data)
void	<a href="#">put</a> (String name, double data)
void	<a href="#">put</a> (String name, int data)
void	<a href="#">put</a> (String name, long data)
void	<a href="#">put</a> (String name, String data)
<a href="#">AMFData</a>	<a href="#">remove</a> (int index)
<a href="#">AMFData</a>	<a href="#">remove</a> (String name)
byte[]	<a href="#">serialize</a> ()
byte[]	<a href="#">serialize</a> ( <a href="#">AMFDataContextSerialize</a> context)
void	<a href="#">serialize</a> (java.io.DataOutputStream out)
void	<a href="#">serialize</a> (java.io.DataOutputStream out, <a href="#">AMFDataContextSerialize</a> context)
void	<a href="#">serialize</a> (java.io.DataOutputStream out, int objectEncoding)
byte[]	<a href="#">serialize</a> (int objectEncoding)
void	<a href="#">setClassName</a> (String className)
int	<a href="#">size</a> () Return the number of members of this object/array

String	<a href="#">toString()</a> Return object as formatted string
--------	---

Methods inherited from class <a href="#">com.wowza.wms.amf.AMFData</a>
<a href="#">createContextDeserialize</a> , <a href="#">createContextDeserialize</a> , <a href="#">createContextSerialize</a> , <a href="#">createContextSerialize</a> , <a href="#">deserialize</a> , <a href="#">deserialize</a> , <a href="#">deserializeInnerObject</a> , <a href="#">getReference</a> , <a href="#">getType</a> , <a href="#">getValue</a> , <a href="#">isAMF3Start</a> , <a href="#">isArrayStart</a> , <a href="#">isByteArrayStart</a> , <a href="#">isMixedArrayStart</a> , <a href="#">isObjEnd</a> , <a href="#">isObjStart</a> , <a href="#">peekByte</a> , <a href="#">serialize</a> , <a href="#">serialize</a> , <a href="#">serialize</a> , <a href="#">serialize</a> , <a href="#">serialize</a> , <a href="#">setType</a> , <a href="#">skipByte</a> , <a href="#">testNextByte</a> , <a href="#">triggerAMF3Switch</a>

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

Methods inherited from interface <a href="#">com.wowza.wms.amf.IAMFDataObj</a>
<a href="#">containsKey</a> , <a href="#">get</a> , <a href="#">get</a> , <a href="#">getBoolean</a> , <a href="#">getBoolean</a> , <a href="#">getBytes</a> , <a href="#">getBytes</a> , <a href="#">getDate</a> , <a href="#">getDate</a> , <a href="#">getDouble</a> , <a href="#">getDouble</a> , <a href="#">getFloat</a> , <a href="#">getFloat</a> , <a href="#">getInt</a> , <a href="#">getInt</a> , <a href="#">getKey</a> , <a href="#">getKeys</a> , <a href="#">getLong</a> , <a href="#">getLong</a> , <a href="#">getObject</a> , <a href="#">getObject</a> , <a href="#">getShort</a> , <a href="#">getShort</a> , <a href="#">getString</a> , <a href="#">getString</a> , <a href="#">put</a> , <a href="#">put</a> , <a href="#">put</a> , <a href="#">put</a> , <a href="#">put</a> , <a href="#">put</a> , <a href="#">remove</a> , <a href="#">remove</a>

## 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	<a href="#">AMFDataTrait</a> ( ) Constructor
--------	---

### Method Summary

void	<a href="#">addMember</a> (String member) Add a member
<a href="#">AMFDataTrait</a>	<a href="#">clone</a> ( ) clone the trait
String	<a href="#">getClassName</a> ( ) Get class name
<a href="#">AMFData</a>	<a href="#">getInnerObj</a> ( ) Get inner object
String	<a href="#">getMember</a> (int i) Get member by index
int	<a href="#">getMemberCount</a> ( ) Get the number of members
java.util.List	<a href="#">getMembers</a> ( ) Get a list of trait members
boolean	<a href="#">isDynamic</a> ( ) Is class dynamic
boolean	<a href="#">isMember</a> (String member) Return true if member of this trait
void	<a href="#">setClassName</a> (String className) Set class name
void	<a href="#">setDynamic</a> (boolean isDynamic) Set isDynamic
void	<a href="#">setInnerObj</a> ( <a href="#">AMFData</a> 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	<a href="#">AMFDEBUGHEADERSIZE</a>
public static final	<a href="#">AMFFORCETYPE1</a> Value: <b>true</b>

#### Constructor Summary

public	<a href="#">AMFObj</a> (int id) Create new AMFObj for a given channel (id)
public	<a href="#">AMFObj</a> (int id, int objectEncoding) Create new AMFObj for a given channel (id)

#### Method Summary

void	<a href="#">addChunk</a> (byte[] buffer, int offset, int len) Add a chunk to the chunk list
void	<a href="#">clearByteContainer</a> () Clear the byte container
long	<a href="#">getAbsTimecode</a> () Get the absolute time code
int	<a href="#">getByteContainerLevel</a> () Fake container for processing
long	<a href="#">getChunkCounter</a> ()
java.util.List	<a href="#">getChunks</a> () Get the chunks that make up this packet
int	<a href="#">getId</a> () Get channel id
int	<a href="#">getObjectEncoding</a> ()

int	<a href="#"><u>getSize()</u></a> Get packet size
int	<a href="#"><u>getSrc()</u></a> Get stream id (0 if not stream data)
long	<a href="#"><u>getTimecode()</u></a> Get timecode (milliseconds) sometimes relative
int	<a href="#"><u>getType()</u></a> Get content type IVHost.CONTENTTYPE_*
long	<a href="#"><u>incAbsTimecode(long absTimecode)</u></a> Increment the absolute timecode
void	<a href="#"><u>incByteContainerLevel(int byteContainerLevel)</u></a> Fake container for processing
boolean	<a href="#"><u>isByteContainerEmpty()</u></a> Fake container for processing
boolean	<a href="#"><u>isByteContainerFull()</u></a> Fake container for processing
boolean	<a href="#"><u>isLastSentAbsTimecode()</u></a>
boolean	<a href="#"><u>isLongTimecode()</u></a> Get is a 32 bit timecode
boolean	<a href="#"><u>isNew()</u></a> Is this a new packet.
boolean	<a href="#"><u>isObjectEncodingAMF0()</u></a>
boolean	<a href="#"><u>isObjectEncodingAMF3()</u></a>
long	<a href="#"><u>setAbsTimecodeLong(long absTimecode)</u></a> Set the absolute timecode
long	<a href="#"><u>setAbsTimecodeShort(long absTimecode)</u></a> Set the absolute timecode
void	<a href="#"><u>setByteContainerLevel(int byteContainerLevel)</u></a> Fake container for processing
void	<a href="#"><u>setChunkCounter(long chunkCounter)</u></a>
void	<a href="#"><u>setId(int id)</u></a> Set channel id
void	<a href="#"><u>setLastSentAbsTimecode(boolean isLastSentAbsTimecode)</u></a>
void	<a href="#"><u>setLongTimecode(boolean isLongTimecode)</u></a> Set is a 32 bit timecode
void	<a href="#"><u>setNew(boolean isNew)</u></a> Set is new packet

void	<a href="#"><u>setObjectEncoding</u></a> (int objectEncoding)
void	<a href="#"><u>setSize</u></a> (int size) Set packet size
void	<a href="#"><u>setSrc</u></a> (int src) Set stream id (0 if not stream data)
void	<a href="#"><u>setTimecode</u></a> (long timecode) Set timecode (milliseconds) sometimes relative
void	<a href="#"><u>setType</u></a> (int type) Set content type IVHost.CONTENT_TYPE_*
String	<a href="#"><u>toString</u></a> () 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	<a href="#">buffer</a>
public	<a href="#">len</a>
public	<a href="#">offset</a>

### Constructor Summary

public	<a href="#">AMFObjChunk</a> (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. AMFPacket is also used to store data read/written to/from an flv file.

### Constructor Summary

public	<a href="#">AMFPacket()</a> Create new empty packet
public	<a href="#">AMFPacket(int type, int src, int size)</a> Create new packet with given values

### Method Summary

int	<a href="#">addData(byte[] data, int offset, int size)</a> Add data to the packet
int	<a href="#">addDataEx(byte[] srcData, int srcOffset, int destOffset, int srcBytes)</a> Add data to the packet
static int	<a href="#">calcTotalPacketSize(int packetSize, int headerSize, int chunkSize, int amfNumber, boolean isLongTimecode)</a> Calculate the total packet size for given packet parameters
<a href="#">AMFPacket</a>	<a href="#">clone()</a>
long	<a href="#">getAbsTimecode()</a> Get absolute timecode (milliseconds)
byte[]	<a href="#">getData()</a> Get data as byte[]
java.nio.ByteBuffer	<a href="#">getDataBuffer()</a> Get data as ByteBuffer
int	<a href="#">getFirstByte()</a> Get first byte of data (used to peek into packet)
int	<a href="#">getMissing()</a> Get the number of bytes remaining unfilled in the packet
int	<a href="#">getSecondByte()</a> Get second byte of data (used to peek into packet)

long	<a href="#"><code>getSeq()</code></a> Get packet sequence number.
int	<a href="#"><code>getSize()</code></a> Get packet size
int	<a href="#"><code>getSrc()</code></a> Get stream id (0 if not stream data)
long	<a href="#"><code>getTimecode()</code></a> Get timecode (milliseconds) relative to the <i>previous</i> packet.
int	<a href="#"><code>getType()</code></a> Get content type IVHost.CONTENTTYPE_*
boolean	<a href="#"><code>isAudio()</code></a> Is this an audio packet IVHost.CONTENTTYPE_AUDIO
boolean	<a href="#"><code>isVideo()</code></a> Is this an audio packet IVHost.CONTENTTYPE_VIDEO
void	<a href="#"><code>setAbsTimecode(long absTimecode)</code></a> Set absolute timecode (milliseconds)
void	<a href="#"><code>setDataBuffer(byte[] data)</code></a> Set the data buffer to a byte array
void	<a href="#"><code>setDataBuffer(java.nio.ByteBuffer data)</code></a> Set the data for this packet
void	<a href="#"><code>setSeq(long seq)</code></a> Set packet sequence.
void	<a href="#"><code>setSize(int size)</code></a> Set packet size
void	<a href="#"><code>setSrc(int src)</code></a> Set stream id (0 if not stream data)
void	<a href="#"><code>setTimecode(long timecode)</code></a> Set timecode (milliseconds) relative,
void	<a href="#"><code>setTimecodes(long timecode, long absTimecode)</code></a> Set both relative and absolute timecode in one call (milliseconds)
void	<a href="#"><code>setType(int type)</code></a> Set content type IVHost.CONTENTTYPE_*
String	<a href="#"><code>toString()</code></a> Return object as formatted string
void	<a href="#"><code>truncatePacket(int newSize)</code></a>

#### 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	<a href="#"><code>containsKey</code></a> (String name) Return true if the object/array contains key
<a href="#">AMFData</a>	<a href="#"><code>get</code></a> (int index) Return the object at a particular index.
<a href="#">AMFData</a>	<a href="#"><code>get</code></a> (String name) Return the object at a particular key.
boolean	<a href="#"><code>getBoolean</code></a> (int index) Get item at index return as boolean
boolean	<a href="#"><code>getBoolean</code></a> (String name) Get item at key return as boolean
byte	<a href="#"><code>getByte</code></a> (int index) Get item at index return as byte
byte	<a href="#"><code>getByte</code></a> (String name) Get item at key return as byte
java.util.Date	<a href="#"><code>getDate</code></a> (int index) Get item at index return as Date
java.util.Date	<a href="#"><code>getDate</code></a> (String name) Get item at key return as Date
double	<a href="#"><code>getDouble</code></a> (int index) Get item at index return as double
double	<a href="#"><code>getDouble</code></a> (String name) Get item at key return as double
float	<a href="#"><code>getFloat</code></a> (int index) Get item at index return as float
float	<a href="#"><code>getFloat</code></a> (String name) Get item at key return as float
int	<a href="#"><code>getInt</code></a> (int index) Get item at index return as int
int	<a href="#"><code>getInt</code></a> (String name) Get item at key return as int

String	<a href="#"><u>getKey</u></a> (int index) Return the key at a particular index.
java.util.List	<a href="#"><u>getKeys</u></a> () Return a list of all the keys (the list is a copy)
long	<a href="#"><u>getLong</u></a> (int index) Get item at index return as long
long	<a href="#"><u>getLong</u></a> (String name) Get item at key return as long
<a href="#"><u>AMFDataObj</u></a>	<a href="#"><u>getObject</u></a> (int index) Get item at index return as AMFDataObj
<a href="#"><u>AMFDataObj</u></a>	<a href="#"><u>getObject</u></a> (String name) Get item at key return as AMFDataObj
short	<a href="#"><u>getShort</u></a> (int index) Get item at index return as short
short	<a href="#"><u>getShort</u></a> (String name) Get item at key return as short
String	<a href="#"><u>getString</u></a> (int index) Get item at index return as String
String	<a href="#"><u>getString</u></a> (String name) Get item at key return as String
void	<a href="#"><u>put</u></a> (String name, <a href="#"><u>AMFData</u></a> data) Put or replace object at key
void	<a href="#"><u>put</u></a> (String name, boolean data) Put or replace boolean value at key (data will be wrapped in an AMFDataItem object)
void	<a href="#"><u>put</u></a> (String name, java.util.Date data) Put or replace date value at key (data will be wrapped in an AMFDataItem object)
void	<a href="#"><u>put</u></a> (String name, double data) Put or replace double value at key (data will be wrapped in an AMFDataItem object)
void	<a href="#"><u>put</u></a> (String name, int data) Put or replace int value at key (data will be wrapped in an AMFDataItem object)
void	<a href="#"><u>put</u></a> (String name, long data) Put or replace long value at key (data will be wrapped in an AMFDataItem object)
void	<a href="#"><u>put</u></a> (String name, String data) Put or replace string value at key (data will be wrapped in an AMFDataItem object)
<a href="#"><u>AMFData</u></a>	<a href="#"><u>remove</u></a> (int index) Remove element by index
<a href="#"><u>AMFData</u></a>	<a href="#"><u>remove</u></a> (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	<a href="#"><u>DEFAULT_APPLICATION_NAME</u></a> Value: <b>_defapp_</b>
---------------------	---

### Method Summary

void	<a href="#"><u>addApplicationInstanceListener</u></a> ( <a href="#"><u>IApplicationInstanceNotify</u></a> applicationInstanceListener) Add applicationInstance listener.
<a href="#"><u>IApplicationInstance</u></a>	<a href="#"><u>getAppInstance</u></a> (String name) Get applicationInstance object by name
java.util.List	<a href="#"><u>getAppInstanceNames</u></a> ( ) Get a list of application instance names
String	<a href="#"><u>getApplicationPath</u></a> ( ) Get the root path for application
String	<a href="#"><u>getConfigPath</u></a> ( ) Get full path to Application.xml file
<a href="#"><u>ConnectionCounter</u></a>	<a href="#"><u>getConnectionCounter</u></a> ( ) Get the connectionCounter for application
ConnectionCounterSimple	<a href="#"><u>getConnectionCounter</u></a> (int counterIndex) Get the connectionCounter for application for application for a specific technology (see IVHost.COUNTER_*)
String	<a href="#"><u>getDateStarted</u></a> ( ) Get date application started
<a href="#"><u>IOPerformanceCounter</u></a>	<a href="#"><u>getIoPerformanceCounter</u></a> ( ) Get the performance counter for application
<a href="#"><u>IOPerformanceCounter</u></a>	<a href="#"><u>getIoPerformanceCounter</u></a> (int counterIndex) Get the performance counter for application for a specific technology (see IVHost.COUNTER_*)
String	<a href="#"><u>getName</u></a> ( ) Get the name of application
<a href="#"><u>WMSProperties</u></a>	<a href="#"><u>getProperties</u></a> ( ) Get application properties

void	<a href="#"><u>getProtocolUsage</u></a> (boolean[] protocolsInUse) Get the protocols in use by this application instance (see IApplicationInstance.PROTOCOLUSAGE_*)
String	<a href="#"><u>getTimeRunning</u></a> () Get time application running
double	<a href="#"><u>getTimeRunningSeconds</u></a> () Get time running in seconds
<a href="#"><u>IVHost</u></a>	<a href="#"><u>getVHost</u></a> () Get the parent vHost object
boolean	<a href="#"><u>isAppInstanceLoaded</u></a> (String name) Return true if application instance is loaded
String	<a href="#"><u>readAppConfig</u></a> (String sName) Method to read xml config file..
void	<a href="#"><u>removeAppInstance</u></a> ( <a href="#"><u>IApplicationInstance</u></a> appInstance) Disconnect all clients connected to an application instance and remove it from the IApplication application list.
void	<a href="#"><u>removeApplicationInstanceListener</u></a> ( <a href="#"><u>IApplicationInstanceNotify</u></a> applicationInstanceListener) Remove applicationInstance listener.
void	<a href="#"><u>setName</u></a> (String name) Set name of application
void	<a href="#"><u>shutdown</u></a> (boolean isServerShutdown) shutdown application
void	<a href="#"><u>shutdownAppInstance</u></a> (String appInstanceName) Shutdown an application instance by name.
boolean	<a href="#"><u>writeAppConfig</u></a> (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	<a href="#"><u>DEFAULT_APPINSTANCE_NAME</u></a> Value: <b>_definst_</b>
public static final	<a href="#"><u>PROTOCOLUSAGE_CUPERTINO</u></a> Value: <b>7</b>
public static final	<a href="#"><u>PROTOCOLUSAGE_RTMP</u></a> Value: <b>0</b>
public static final	<a href="#"><u>PROTOCOLUSAGE_RTMP_E</u></a> Value: <b>4</b>
public static final	<a href="#"><u>PROTOCOLUSAGE_RTMP_S</u></a> Value: <b>2</b>
public static final	<a href="#"><u>PROTOCOLUSAGE_RTMP_T</u></a> Value: <b>1</b>
public static final	<a href="#"><u>PROTOCOLUSAGE_RTMP_T_E</u></a> Value: <b>5</b>
public static final	<a href="#"><u>PROTOCOLUSAGE_RTMP_T_S</u></a> Value: <b>3</b>
public static final	<a href="#"><u>PROTOCOLUSAGE_RTP</u></a> Value: <b>6</b>
public static final	<a href="#"><u>PROTOCOLUSAGE_SANJOSE</u></a> Value: <b>9</b>
public static final	<a href="#"><u>PROTOCOLUSAGE_SMOOTH</u></a> Value: <b>8</b>
public static final	<a href="#"><u>PROTOCOLUSAGE_TOTAL</u></a> Value: <b>11</b>

public static final	<a href="#">PROTOCOLUSAGE_WEBM</a> Value: <b>10</b>
---------------------	--

## Method Summary

void	<a href="#">addClientListener</a> ( <a href="#">IClientNotify</a> clientListener) Add client listener.
void	<a href="#">addDvrRecorderListener</a> ( <a href="#">ILiveStreamDvrRecorderActionNotify</a> listener) Add a Dvr Recorder listener (see: <a href="#">ILiveStreamDvrRecorderActionNotify</a> )
void	<a href="#">addDvrStreamManagerListener</a> ( <a href="#">IDvrStreamManagerActionNotify</a> listener) Add a Dvr Application Store Manager listener (see: <a href="#">IDvrStoreActionNotify</a> )
void	<a href="#">addHTTPStreamerSession</a> ( <a href="#">IHTTPStreamerSession</a> httpStreamerSession) Add a HTTPStreamerSession to this application instance
void	<a href="#">addLiveStreamPacketizerListener</a> ( <a href="#">ILiveStreamPacketizerActionNotify</a> <a href="#">LiveStreamPacketizerListener</a> ) Add a Live Stream Packetizer listener (see: <a href="#">ILiveStreamPacketizerActionNotify</a> )
void	<a href="#">addLiveStreamTranscoderListener</a> ( <a href="#">ILiveStreamTranscoderNotify</a> <a href="#">LiveStreamTranscoderListener</a> ) Add a live stream transcoder listener
void	<a href="#">addMediaCasterListener</a> ( <a href="#">IMediaCasterNotify</a> mediaCasterListener) Add mediaCaster listener.
void	<a href="#">addMediaCasterListener</a> ( <a href="#">IMediaCasterNotify2</a> mediaCasterListener) Add mediaCaster listener.
void	<a href="#">addMediaReaderListener</a> ( <a href="#">IMediaReaderActionNotify</a> mediaReaderListener) Add media reader listener.
void	<a href="#">addMediaStreamListener</a> ( <a href="#">IMediaStreamNotify</a> mediaStreamListener) Add mediaStream listener.
void	<a href="#">addMediaWriterListener</a> ( <a href="#">IMediaWriterActionNotify</a> listener) Add a MediaWriter listener class.
void	<a href="#">addModuleListener</a> ( <a href="#">IModuleNotify</a> moduleListener) Add module listener.
void	<a href="#">addPlayStreamByName</a> ( <a href="#">IMediaStream</a> stream, String name) Add a media stream to the list of streams that are listening for a published stream
void	<a href="#">addPublisher</a> ( <a href="#">Publisher</a> publisher) Add a server side publisher to this application instance
void	<a href="#">addRTPIncomingDatagramPortAll</a> () Allow all incoming RTP UDP ports for this application instance
void	<a href="#">addRTPIncomingDatagramPortRange</a> (int startPort, int endPort) Add a port range to the list of valid incoming RTP UDP ports
void	<a href="#">addRTPSession</a> ( <a href="#">RTPSession</a> rtpSession) Add an RTP session to this application instance

void	<a href="#"><u>addSharedObjectListener</u></a> ( <a href="#"><u>ISharedObjectNotify</u></a> sharedObjectListener, boolean isPersistent) Add sharedObject listener.
void	<a href="#"><u>broadcastMsg</u></a> (java.util.List clientList, String handlerName) Broadcast a message to a specific list of clients connected to this application instance
void	<a href="#"><u>broadcastMsg</u></a> (java.util.List clientList, String handlerName, Object[] params) Broadcast a message to a specific list of clients connected to this application instance
void	<a href="#"><u>broadcastMsg</u></a> (String handlerName, Object[] params) Broadcast a message to all clients connected to this applicationInstance
boolean	<a href="#"><u>containsDvrRecorder</u></a> (String dvrRecorder) Does this application instance contain a references to this DVR recorder.
boolean	<a href="#"><u>containsHTTPStreamer</u></a> (String httpStreamer) Does this application instance allow streaming of a given HTTPStreamer
boolean	<a href="#"><u>containsLiveStreamPacketizer</u></a> (String liveStreamPacketizer) Does this application instance contain a references to this live stream packetizer.
boolean	<a href="#"><u>containsLiveStreamTranscoder</u></a> (String liveStreamTranscoder) Return true if this application instance contains the transcoder name
String	<a href="#"><u>decodeStorageDir</u></a> (String storageDir) This function will take a storage path that uses variables and expand the variables based on the context.
String[]	<a href="#"><u>getAllowDomains</u></a> () Get the list of domain names used to control access to this application.
<a href="#"><u>IApplication</u></a>	<a href="#"><u>getApplication</u></a> () Get parent application
int	<a href="#"><u>getApplicationInstanceTouchTimeout</u></a> () Get the application instance touch timeout (milliseconds).
int	<a href="#"><u>getApplicationTimeout</u></a> () Get application timeout (milliseconds)
<a href="#"><u>IClient</u></a>	<a href="#"><u>getClient</u></a> (int index) <b>Deprecated.</b> Get the client connection at index. This method is deprecated. It is best to use <a href="#"><u>getClient()</u></a> to return a List objects.
<a href="#"><u>IClient</u></a>	<a href="#"><u>getClientById</u></a> (int index) Get a client connection by the client Id
int	<a href="#"><u>getClientCount</u></a> () Get number of client connections currently connected to applicationInstance
int	<a href="#"><u>getClientCountTotal</u></a> () Get number of client connections in total that have connected to this applicationInstance
int	<a href="#"><u>getClientIdleFrequency</u></a> () Get default client idle frequency (milliseconds)
java.util.List	<a href="#"><u>getClients</u></a> () Get the set of clients currently connected to this application instance (replaces getClient(index))

edu.emory.mathcs.backport.java.util.concurrent.locks.WMSReadWriteLock	<a href="#"><u>getClientsLockObj()</u></a> Get the read/write lock for this application instance
<a href="#"><u>ConnectionCounter</u></a>	<a href="#"><u>getConnectionCounter()</u></a> Get the connectionCounter for applicationInstance
ConnectionCounterSimple	<a href="#"><u>getConnectionCounter(int counterIndex)</u></a> Get the connectionCounter for applicationInstance for a specific technology (see IVHost.COUNTER_*)
String	<a href="#"><u>getContextStr()</u></a> Returns the application context string in the form [application]/[appInstance].
String	<a href="#"><u>getDateStarted()</u></a> Get date applicationInstance started
DvrApplicationContext	<a href="#"><u>getDvrApplicationContext()</u></a> Get live stream dvr application context
<a href="#"><u>WMSProperties</u></a>	<a href="#"><u>getDvrProperties()</u></a> Get the property collection of DVR settings that are specific to this application instance.
String	<a href="#"><u>getDvrRecorderList()</u></a> Get the comma separated list of Dvr Recorder names being used by this application (see conf/Dvr.xml)
<a href="#"><u>IHTTPStreamerApplicationContext</u></a>	<a href="#"><u>getHTTPStreamerApplicationContext(String httpStreamName, boolean doCreate)</u></a> Get the HTTPStreamer application context for a given HTTPStreamer adapter
String	<a href="#"><u>getHTTPStreamerList()</u></a> Get the comma separated list of HTTPStreamers names being used by this application (see conf/HTTPStreamers.xml)
<a href="#"><u>WMSProperties</u></a>	<a href="#"><u>getHTTPStreamerProperties()</u></a> Get the property collection of HTTPStreamer settings that are specific to this application instance
int	<a href="#"><u>getHTTPStreamerSessionCount()</u></a> Get the current number of HTTPStreamerSessions associated with this application instance
int	<a href="#"><u>getHTTPStreamerSessionCount(int protocol)</u></a> Get the current number of HTTPStreamerSessions associated with this application instance by protocol.
int	<a href="#"><u>getHTTPStreamerSessionCount(int protocol, String streamName)</u></a> Get the current number of HTTPStreamerSessions associated with this application instance and stream name by protocol.
int	<a href="#"><u>getHTTPStreamerSessionCount(String streamName)</u></a> Get the current number of HTTPStreamerSessions associated with this application instance and stream name
java.util.Map	<a href="#"><u>getHTTPStreamerSessionCountsByName(int protocol)</u></a> Get a map of session counts by name for a given protocol
java.util.List	<a href="#"><u>getHTTPStreamerSessions()</u></a> Get the HTTPStreamerSessions associated with this application instance



java.util.List	<a href="#"><u>getHTTPStreamerSessions</u></a> (int protocol) Get the HTTPStreamerSessions associated with this application instance by protocol.
java.util.List	<a href="#"><u>getHTTPStreamerSessions</u></a> (int protocol, String streamName) Get the HTTPStreamerSessions associated with this application instance for a stream name by protocol.
java.util.List	<a href="#"><u>getHTTPStreamerSessions</u></a> (String streamName) Get the HTTPStreamerSessions associated with this application instance for a stream name
<a href="#"><u>IOPerformanceCounter</u></a>	<a href="#"><u>getIOPerformanceCounter</u></a> () Get the performance counter for applicationInstance
<a href="#"><u>IOPerformanceCounter</u></a>	<a href="#"><u>getIOPerformanceCounter</u></a> (int counterIndex) Get the performance counter for applicationInstance for a specific technology (see IVHost.COUNTER_*)
long	<a href="#"><u>getLastTouchTime</u></a> () Get the last time the instance was touched (milliseconds)
<a href="#"><u>ILiveStreamDvrRecorderControl</u></a>	<a href="#"><u>getLiveStreamDvrRecorderControl</u></a> () Get the Live Stream DVR Recorder Controller.
<a href="#"><u>ILiveStreamPacketizerControl</u></a>	<a href="#"><u>getLiveStreamPacketizerControl</u></a> () Get the Live Stream Packetizer Controller.
String	<a href="#"><u>getLiveStreamPacketizerList</u></a> () Get the comma separated list of LiveStreamPacketizers names being used by this application (see conf/LiveStreamPacketizers.xml)
<a href="#"><u>WMSProperties</u></a>	<a href="#"><u>getLiveStreamPacketizerProperties</u></a> () Get the property collection of LiveStreamPacketizer settings that are specific to this application instance
<a href="#"><u>ILiveStreamTranscoderControl</u></a>	<a href="#"><u>getLiveStreamTranscoderControl</u></a> () Get the Live Stream Transcoder Controller.
String	<a href="#"><u>getLiveStreamTranscoderList</u></a> () Get comma separated list of transcoders to use for this application instance
int	<a href="#"><u>getMaximumPendingReadBytes</u></a> () Set maximum number of bytes a client connection can have waiting to be written before the connection is terminated.
int	<a href="#"><u>getMaximumPendingWriteBytes</u></a> () Get maximum number a bytes a client connection can have waiting to be sent before the connection is terminated.
int	<a href="#"><u>getMaximumSetBufferTime</u></a> () Get maximum number of milliseconds allowed for the NetStream.setBufferTime(secs) call.
int	<a href="#"><u>getMaxStorageDirDepth</u></a> () Maximum folder depth allowed for the StreamStorageDir and SharedObjectStorageDir paths
<a href="#"><u>WMSProperties</u></a>	<a href="#"><u>getMediaCasterProperties</u></a> () Get the property collection of media caster settings that are specific to this application instance
int	<a href="#"><u>getMediacasterRTPRTSPRTPTransportMode</u></a> () RTP MediaCaster RTSP/RTP transport mode.

<a href="#">MediaCasterStreamMap</a>	<a href="#">getMediaCasterStreams()</a> Get the media caster streams attached to this application instance
<a href="#">IMediaCasterValidateMediaCaster</a>	<a href="#">getMediaCasterValidator()</a> Get the MediaCaster validator interface for this application instance
<a href="#">IMediaListProvider</a>	<a href="#">getMediaListProvider()</a> Get the current media list provider.
int	<a href="#">getMediaReaderContentType(String mediaType)</a> Get the content type of a media stream name prefix (see IMediaReader.CONTENTTYPE_*)
<a href="#">WMSProperties</a>	<a href="#">getMediaReaderProperties()</a> Get the property collection of media reader settings that are specific to this application instance
<a href="#">WMSProperties</a>	<a href="#">getMediaWriterProperties()</a> Get the property collection of media reader settings that are specific to this application instance
ModuleFunctions	<a href="#">getModFunctions()</a> Get list of application modules
Object	<a href="#">getModuleInstance(String name)</a> Get the instance of the module class for this application instance.
ModuleList	<a href="#">getModuleList()</a> Get the list of loaded modules.
String	<a href="#">getName()</a> Get applicationInstance name
int	<a href="#">getPingTimeout()</a> Get ping timeout (milliseconds)
int	<a href="#">getPlayStreamCount(String streamName)</a> Get the number of Flash players playing a given stream name
java.util.Map	<a href="#">getPlayStreamCountsByName()</a> Get a map of stream names to number of Flash players playing the stream name
java.util.List	<a href="#">getPlayStreamsByName(String name)</a> Get a list of media streams that are listening for published stream.
<a href="#">WMSProperties</a>	<a href="#">getProperties()</a> Get applicationInstance properties
boolean[]	<a href="#">getProtocolUsage()</a> Get the protocols in use by this application instance (see IApplicationInstance.PROTOCOLUSAGE_*)
void	<a href="#">getProtocolUsage(boolean[] protocolsInUse)</a> Get the protocols in use by this application instance (see IApplicationInstance.PROTOCOLUSAGE_*)
int	<a href="#">getPublisherCount()</a> Get the current number of server side publishers
java.util.List	<a href="#">getPublishers()</a> Get the set of server side publishers

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

int	<a href="#"><u>getRTSPMaximumPendingWriteBytes()</u></a> Get the maximum number of pending write bytes for an RTSP session
String	<a href="#"><u>getRTSPOriginAddressType()</u></a> Get the origin IP address type (IP4) to used in the Session Description Protocol data exchanged for an RTSP/RTP session
String	<a href="#"><u>getRTSPOriginIpAddress()</u></a> Get the origin IP address to used in the Session Description Protocol data exchanged for an RTSP/RTP session
int	<a href="#"><u>getRTSPSessionTimeout()</u></a> Get the RTSP session timeout (milliseconds)
String	<a href="#"><u>getSharedObjectReadAccess()</u></a> Get the default shared object read access
<a href="#"><u>ISharedObjects</u></a>	<a href="#"><u>getSharedObjects()</u></a> Get non-persistent shared object collection
<a href="#"><u>ISharedObjects</u></a>	<a href="#"><u>getSharedObjects(boolean isPersistent)</u></a> Get either persistent or non-persistent shared object collection
String	<a href="#"><u>getSharedObjectWriteAccess()</u></a> Get the default shared object write access
String	<a href="#"><u>getStreamAudioSampleAccess()</u></a> Get the default stream audio sample access
int	<a href="#"><u>getStreamCount()</u></a> Get the total number of open streams attached to this application instance
<a href="#"><u>IMediaStreamFileMapper</u></a>	<a href="#"><u>getStreamFileMapper()</u></a> Get the stream file mapper.
String	<a href="#"><u>getStreamKeyDir()</u></a> Get the stream key path
String	<a href="#"><u>getStreamKeyPath()</u></a> Get the resolved key path to the MediaStreams encryption keys
<a href="#"><u>IMediaStreamNameAliasProvider</u></a>	<a href="#"><u>getStreamNameAliasProvider()</u></a> Get the stream name alias provider
<a href="#"><u>WMSProperties</u></a>	<a href="#"><u>getStreamProperties()</u></a> Get the property collection of stream settings that are specific to this application instance
String	<a href="#"><u>getStreamReadAccess()</u></a> Get the default stream read access
<a href="#"><u>MediaStreamMap</u></a>	<a href="#"><u>getStreams()</u></a> Get all the mediaStream objects attached to this applicationInstance
String	<a href="#"><u>getStreamStorageDir()</u></a> Get stream storage path
String	<a href="#"><u>getStreamStoragePath()</u></a> Get the resolved storage path to the MediaStreams

String	<a href="#"><u>getStreamType()</u></a> Get default streamType for application.
String	<a href="#"><u>getStreamVideoSampleAccess()</u></a> Get the default stream video sample access
String	<a href="#"><u>getStreamWriteAccess()</u></a> Get the default stream write access
String	<a href="#"><u>getTimeRunning()</u></a> Get time applicationInstance running
double	<a href="#"><u>getTimeRunningSeconds()</u></a> Get time running in seconds
LiveStreamTranscoderApplicationContext	<a href="#"><u>getTranscoderApplicationContext()</u></a> Get live stream transcoder application context
<a href="#"><u>WMSProperties</u></a>	<a href="#"><u>getTranscoderProperties()</u></a> Get the property collection of Transcoder settings that are specific to this application instance
int	<a href="#"><u>getValidationFrequency()</u></a> Get time between validation pings (milliseconds)
<a href="#"><u>IVHost</u></a>	<a href="#"><u>getVHost()</u></a> Get parent vHost
void	<a href="#"><u>incClientCountTotal()</u></a> Increment the total number of connected client counter by one
boolean	<a href="#"><u>isAcceptConnection()</u></a> Is auto accept connection on/off
boolean	<a href="#"><u>isRTPIncomingDatagramPortValid(int port)</u></a> Check a port number to be sure it is a valid RTP UDP port for this application instance
boolean	<a href="#"><u>isValidateFMLEConnections()</u></a> Returns true if validating FMLE connection (default is false)
void	<a href="#"><u>notifyDvrStreamManagerCreate(IDvrStreamManager dvrStoreManager)</u></a> Notify listeners that Dvr Application Store Manager has been created.
void	<a href="#"><u>notifyDvrStreamManagerDestroy(IDvrStreamManager dvrManager)</u></a> Notify listeners that Dvr Application Store Manager has been destroyed.
void	<a href="#"><u>notifyDvrStreamManagerInit(IDvrStreamManager dvrStoreManager)</u></a> Notify listeners that Dvr Application Store Manager has been initialized.
void	<a href="#"><u>notifyLiveStreamDvrRecorderCreate(ILiveStreamDvrRecorder dvr, String streamName)</u></a> Notify Dvr Recorder Create
void	<a href="#"><u>notifyLiveStreamDvrRecorderDestroy(ILiveStreamDvrRecorder dvr)</u></a> Notify DVR Recorder has been destroyed.
void	<a href="#"><u>notifyLiveStreamDvrRecorderInit(ILiveStreamDvrRecorder dvr, String streamName)</u></a> Notify DVR Recorder has been initialized.

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

void	<a href="#"><u>removeDvrStreamManagerListener</u></a> ( <a href="#"><u>IDvrStreamManagerActionNotify</u></a> listener) Remove a Dvr Application Store Manager listener (see: IDvrStoreActionNotify)
void	<a href="#"><u>removeHTTPStreamerSession</u></a> ( <a href="#"><u>IHTTPStreamerSession</u></a> httpStreamerSession) Remove a HTTPStreamerSession from this application instance
void	<a href="#"><u>removeLiveStreamPacketizerListener</u></a> ( <a href="#"><u>ILiveStreamPacketizerActionNotify</u></a> liveStreamPacketizerListener) Remove a Live Stream Packetizer listener (see: ILiveStreamPacketizerActionNotify)
void	<a href="#"><u>removeLiveStreamTranscoderListener</u></a> ( <a href="#"><u>ILiveStreamTranscoderNotify</u></a> liveStreamTranscoderListener) Remove a live stream transcoder listener
void	<a href="#"><u>removeMediaCasterListener</u></a> ( <a href="#"><u>IMediaCasterNotify</u></a> mediaCasterListener) Remove mediaCaster listener.
void	<a href="#"><u>removeMediaReaderListener</u></a> ( <a href="#"><u>IMediaReaderActionNotify</u></a> mediaReaderListener) Remove media reader listener.
void	<a href="#"><u>removeMediaStreamListener</u></a> ( <a href="#"><u>IMediaStreamNotify</u></a> mediaStreamListener) Remove mediaStream listener.
void	<a href="#"><u>removeMediaWriterListener</u></a> ( <a href="#"><u>IMediaWriterActionNotify</u></a> listener) remove MediaWriter listener class.
void	<a href="#"><u>removeModuleListener</u></a> ( <a href="#"><u>IModuleNotify</u></a> moduleListener) Remove module listener
void	<a href="#"><u>removePlayStreamByName</u></a> ( <a href="#"><u>IMediaStream</u></a> stream) Remove media stream from the list of streams that are listening for a published stream
void	<a href="#"><u>removePublisher</u></a> ( <a href="#"><u>Publisher</u></a> publisher) Remove a server side publisher from this application instance
void	<a href="#"><u>removeRTPSession</u></a> ( <a href="#"><u>RTPSession</u></a> rtpSession) Remove an RTP session from this application instance
void	<a href="#"><u>removeSharedObjectListener</u></a> ( <a href="#"><u>ISharedObjectNotify</u></a> sharedObjectListener, boolean isPersistent) Remove sharedObject listener.
boolean	<a href="#"><u>resetMediaCasterStream</u></a> (String streamName) Reset a media caster stream
boolean	<a href="#"><u>resetMediaCasterStream</u></a> (String streamName, String streamExt) Reset a media caster stream
void	<a href="#"><u>setAcceptConnection</u></a> (boolean acceptConnection) Set is auto accept connection
void	<a href="#"><u>setAllowDomains</u></a> (String[] domainFilter) Set the list of domain names used to control access to this application.
void	<a href="#"><u>setApplicationInstanceTouchTimeout</u></a> (int applicationInstanceTouchTimeout) Set the application instance touch timeout (milliseconds).
void	<a href="#"><u>setApplicationTimeout</u></a> (int applicationTimeout) Set application timeout (milliseconds)

void	<a href="#"><u>setClientIdleFrequency</u></a> (int clientIdleFrequency) Set default client idle frequency (milliseconds)
void	<a href="#"><u>setDvrRecorderList</u></a> (String recorderList) Set the comma separated list of Dvr Recorder names being used by this application (see conf/Dvr.xml)
void	<a href="#"><u>setHTTPStreamerList</u></a> (String httpStreamerList) Set the comma separated list of HTTPStreamer names being used by this application (see conf/HTTPStreamers.xml)
void	<a href="#"><u>setLiveStreamDvrRecorderControl</u></a> ( <a href="#"><u>ILiveStreamDvrRecorderControl</u></a> controller) Set the Live Stream DVR Controller.
void	<a href="#"><u>setLiveStreamPacketizerControl</u></a> ( <a href="#"><u>ILiveStreamPacketizerControl</u></a> liveStreamPacketizerControl) Set the Live Stream Packetizer Contoller.
void	<a href="#"><u>setLiveStreamPacketizerList</u></a> (String liveStreamPacketizerList) Set the comma separated list of LiveStreamPacketizers names being used by this application (see conf/LiveStreamPacketizers.xml)
void	<a href="#"><u>setLiveStreamTranscoderControl</u></a> ( <a href="#"><u>ILiveStreamTranscoderControl</u></a> liveStreamTranscoderControl) Set the Live Stream Transcoder Contoller.
void	<a href="#"><u>setLiveStreamTranscoderList</u></a> (String liveStreamTranscoderList) Set comma separated list of transcoders to use for this application instance
void	<a href="#"><u>setMaximumPendingReadBytes</u></a> (int maximumPendingReaderBytes) Get maximum number of bytes a client connection can have waiting to be written before the connection is terminated.
void	<a href="#"><u>setMaximumPendingWriteBytes</u></a> (int maximumPendingWriteBytes) Set maximum number a bytes a client connection can have waiting to be sent before the connection is terminated.
void	<a href="#"><u>setMaximumSetBufferTime</u></a> (int maximumSetBufferTime) Set maximum number of milliseconds allowed for the NetStream.setBufferTime(secs) call.
void	<a href="#"><u>setMaxStorageDirDepth</u></a> (int maxStorageDirDepth) Maximum folder depth allowed for the StreamStorageDir and SharedObjectStorageDir paths
void	<a href="#"><u>setMediacasterRTPRTSPRTPTransportMode</u></a> (int mediacasterRTPRTSPRTPTransportMode) RTP MediaCaster RTSP/RTP transport mode.
void	<a href="#"><u>setMediaCasterValidator</u></a> ( <a href="#"><u>IMediaCasterValidateMediaCaster</u></a> mediaCasterValidator) Set the MediaCaster validator interface for this application instance
void	<a href="#"><u>setMediaListProvider</u></a> ( <a href="#"><u>IMediaListProvider</u></a> mediaListProvider) Set the current media list provider.
void	<a href="#"><u>setName</u></a> (String name) Set applicationInstance name
void	<a href="#"><u>setPingTimeout</u></a> (int pingTimeout) Set ping timeout (milliseconds)



void	<a href="#"><u>setRepeaterOriginUrl</u></a> (String repeaterOriginUrl) Set the Repeater Origin URL used by the Live Stream Repeater
void	<a href="#"><u>setRepeaterQueryString</u></a> (String repeaterQueryString) Set the Repeater query string that is used to connect to the origin.
void	<a href="#"><u>setRsoStorageDir</u></a> (String rsoStorageDir) Set remote shared object storage path
void	<a href="#"><u>setRTPAVSyncMethod</u></a> (int rtpAVSyncMethod) Set RTP audio/video sync method (RTPStream.AVSYNCMETHODS_SENDERREPORT, RTPStream.AVSYNCMETHODS_SYSTEMCLOCK, RTPStream.AVSYNCMETHODS_RTPTIMECODE)
void	<a href="#"><u>setRTPIIdleFrequency</u></a> (int rtspIdleFrequency) Get the default RTP idle frequency (milliseconds)
void	<a href="#"><u>setRTPMaxRTCPWaitTime</u></a> (int rtpMaxRTCPWaitTime) Set the maximum time to wait for RTCP packets (milliseconds)
void	<a href="#"><u>setRTPPlayAuthenticationMethod</u></a> (String rtpPlayAuthenticationMethod) Set the RTP play authentication method (as defined in conf/Authentication.xml)
void	<a href="#"><u>setRTPPublishAuthenticationMethod</u></a> (String rtpPublishAuthenticationMethod) Set the RTP publish authentication method (as defined in conf/Authentication.xml)
void	<a href="#"><u>setRTSPBindIpAddress</u></a> (String rtspBindIpAddress) Get the IP address to which UDP ports will be bound for RTSP/RTP sessions
void	<a href="#"><u>setRTSPConnectionAddressType</u></a> (String rtspConnectionAddressType) Set the connection IP address type (IP4) to used in the Session Description Protocol data exchanged for an RTSP/RTP session
void	<a href="#"><u>setRTSPConnectionIpAddress</u></a> (String rtspConnectionIpAddress) Set the connection IP address to used in the Session Description Protocol data exchanged for an RTSP/RTP session
void	<a href="#"><u>setRTSPMaximumPendingWriteBytes</u></a> (int rtspMaximumPendingWriteBytes) Set the maximum number of pending write bytes for an RTSP session
void	<a href="#"><u>setRTSPOriginAddressType</u></a> (String rtspOriginAddressType) Set the origin IP address type (IP4) to used in the Session Description Protocol data exchanged for an RTSP/RTP session
void	<a href="#"><u>setRTSPOriginIpAddress</u></a> (String rtspOriginIpAddress) Set the origin IP address to used in the Session Description Protocol data exchanged for an RTSP/RTP session
void	<a href="#"><u>setRTSPSessionTimeout</u></a> (int rtspSessionTimeout) Set the RTSP session timeout (milliseconds)
void	<a href="#"><u>setSharedObjectReadAccess</u></a> (String sharedObjectReadAccess) Set the default shared object read access
void	<a href="#"><u>setSharedObjectWriteAccess</u></a> (String sharedObjectWriteAccess) Set the default shared object write access
void	<a href="#"><u>setStreamAudioSampleAccess</u></a> (String streamAudioSampleAccess) Set the default stream audio sample access

void	<a href="#"><u>setStreamFileMapper</u></a> ( <a href="#"><u>IMediaStreamFileMapper</u></a> streamFileMapper) Set the stream file mapper.
void	<a href="#"><u>setStreamKeyDir</u></a> (String keyStorageDir) Set the stream key path
void	<a href="#"><u>setStreamNameAliasProvider</u></a> ( <a href="#"><u>IMediaStreamNameAliasProvider</u></a> streamNameAliasProvider) Set the stream name alias provider
void	<a href="#"><u>setStreamReadAccess</u></a> (String streamReadAccess) Set the default stream read access
void	<a href="#"><u>setStreamStorageDir</u></a> (String streamStorageDir) Set stream storage path
void	<a href="#"><u>setStreamType</u></a> (String streamType) Set default stream type for application.
void	<a href="#"><u>setStreamVideoSampleAccess</u></a> (String streamVideoSampleAccess) Set the default stream video sample access
void	<a href="#"><u>setStreamWriteAccess</u></a> (String streamWriteAccess) Set the default stream write access
void	<a href="#"><u>setValidateFMLEConnections</u></a> (boolean validateFMLEConnections) Returns true if validating FMLE connection (default is false)
void	<a href="#"><u>setValidationFrequency</u></a> (int validationFrequency) Set time between validation pings (milliseconds)
void	<a href="#"><u>shutdown</u></a> (boolean isServerShutdown, boolean isAppShutdown) shutdown applicationInstance
void	<a href="#"><u>shutdownClient</u></a> ( <a href="#"><u>IClient</u></a> client) shutdown a client connection immediately
boolean	<a href="#"><u>startMediaCasterStream</u></a> (String streamName, String mediaCasterType) Start a media caster stream
boolean	<a href="#"><u>startMediaCasterStream</u></a> (String streamName, String streamExt, String mediaCasterType) Start a media caster stream
void	<a href="#"><u>stopMediaCasterStream</u></a> (String streamName) Stop a media caster stream
void	<a href="#"><u>touch</u></a> () Touch the application instance so that it stays loaded for at least applicationInstanceTouchTimeout
boolean	<a href="#"><u>writeAppInstConfig</u></a> (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	<a href="#">onApplicationInstanceCreate</a> ( <a href="#">IApplicationInstance</a> applicationInstance) Triggered when applicationInstance created
void	<a href="#">onApplicationInstanceDestroy</a> ( <a href="#">IApplicationInstance</a> 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	<a href="#">onApplicationCreate</a> ( <a href="#">IApplication</a> application) Triggered when application created
void	<a href="#">onApplicationDestroy</a> ( <a href="#">IApplication</a> 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	<a href="#">WMSProperties()</a>
--------	---------------------------------

## Method Summary

static void	<a href="#">cloneProperties</a> ( <a href="#">WMSProperties</a> from, <a href="#">WMSProperties</a> to) Copy all properties from "from" properties object to "to" properties object.
String[]	<a href="#">getAllAsStrings()</a> Return all properties as String[].
Object	<a href="#">getProperty</a> (String name) Get property value as generic object.
boolean	<a href="#">getPropertyBoolean</a> (String name, boolean defaultVal) Get property as boolean, return default value if does not exist.
double	<a href="#">getPropertyDouble</a> (String name, double defaultVal) Get property as double, return default value if does not exist.
int	<a href="#">getPropertyInt</a> (String name, int defaultVal) Get property as int, return default value if does not exist.
long	<a href="#">getPropertyLong</a> (String name, long defaultVal) Get property as long, return default value if does not exist.
String	<a href="#">getPropertyStr</a> (String name) Get property as String
String	<a href="#">getPropertyStr</a> (String name, String defaultVal) Get property as String, return default value if does not exist.
void	<a href="#">putAll</a> (java.util.Map m)

void	<a href="#">setProperty</a> (String name, Object value) Set property to generic object.
String	<a href="#">toString</a> ()

#### 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	<a href="#">client</a>
protected	<a href="#">rtpSession</a>
protected	<a href="#">vhost</a>

#### Constructor Summary

public	<a href="#">AuthenticateUsernamePasswordProviderBase()</a>
--------	--

#### Method Summary

<a href="#">IClient</a>	<a href="#">getClient()</a> Get client
<a href="#">RTPSession</a>	<a href="#">getRTPSession()</a> Get RTP Session
<a href="#">IVHost</a>	<a href="#">getVHost()</a> Get vhost
void	<a href="#">setClient(<a href="#">IClient</a> client)</a> Set client
void	<a href="#">setRTPSession(<a href="#">RTPSession</a> rtpSession)</a> Set RTP Session
void	<a href="#">setVHost(<a href="#">IVHost</a> vhost)</a> 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

**vhost**protected com.wowza.wms.vhost.IVHost **vhost****client**protected com.wowza.wms.client.IClient **client****rtpSession**protected com.wowza.wms.rtp.model.RTPSession **rtpSession**

## Constructors

**AuthenticateUsernamePasswordProviderBase**public **AuthenticateUsernamePasswordProviderBase**()

## Methods

**getVHost**public [IVHost](#) **getVHost**()

Get vhost

**Returns:**

vhost

**setVHost**public 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	<a href="#">PASSWORDFILEFORMAT_CLEAR</a> Value: <b>1</b>
public static final	<a href="#">PASSWORDFILEFORMAT_UNKNOWN</a> Value: <b>0</b>

### Method Summary

void	<a href="#">init</a> ( <a href="#">IApplicationInstance</a> appInstance, AuthenticationItem authenticationItem) Initialize authentication class when instantiated as part of an application instance
void	<a href="#">init</a> ( <a href="#">IVHost</a> 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	<a href="#">authenticateHTTPProvider</a> ( <a href="#">IVHost</a> vhost, <a href="#">IHTTPRequest</a> req, <a href="#">IHTTPResponse</a> 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><a href="#">authenticateRTSP</a>(<a href="#">RTPSession</a> 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><a href="#">authenticateSIP</a>(<a href="#">RTPSession</a> 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

<a href="#">IClient</a>	<a href="#">getClient()</a> Get client
String	<a href="#">getPassword(String username)</a> Get password for a given user
<a href="#">RTPSession</a>	<a href="#">getRTPSession()</a> Get RTP session
<a href="#">IVHost</a>	<a href="#">getVHost()</a> Get vhost
void	<a href="#">setClient(<a href="#">IClient</a> client)</a> Set client
void	<a href="#">setRTPSession(<a href="#">RTPSession</a> rtpSession)</a> Set RTP session
void	<a href="#">setVHost(<a href="#">IVHost</a> vhost)</a> Set vhost
boolean	<a href="#">userExists(String username)</a> 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	<a href="#">ConnectionCounter()</a> Create empty ConnectionCounter
--------	---

### Method Summary

void	<a href="#">acceptConnection</a> (ConnectionHolder connectionHolder, byte[] license) Accept a new connection.
void	<a href="#">addConnectionListener</a> (IConnectionNotify connectionNotify) Add a connection listener.
void	<a href="#">decrement</a> (ConnectionHolder connectionHolder, boolean isValid, java.util.Date date, long stamp, byte[] license) Decrement connection counters.
void	<a href="#">disconnect</a> (ConnectionHolder connectionHolder, byte[] license) Disconnect connection.
long	<a href="#">getCurrent</a> () Get total number of client currently conencted to this object.
long	<a href="#">getLastConnectAcceptedStamp</a> () Get time (milliseconds) of the last conenction to this object.
String	<a href="#">getLastConnectAcceptedStampString</a> () Get time (milliseconds) of the last conenction to this object as formatted string.
java.util.Date	<a href="#">getLastConnectAcceptedTime</a> () Get time (milliseconds) of the last accepted conenction to this object.
String	<a href="#">getLastConnectAcceptedTimeString</a> () Get time (milliseconds) of the last accepted conenction to this object as formatted string.
long	<a href="#">getLastConnectRejectedByReasonStamp</a> (int reason) Get time (milliseconds) of the last rejected conenction by reason to this object.

String	<a href="#"><u>getLastConnectRejectedByReasonStampString</u></a> (int reason) Get time (milliseconds) of the last rejected connection by reason to this object as formatted string.
java.util.Date	<a href="#"><u>getLastConnectRejectedByReasonTime</u></a> (int reason) Get date and time of last rejected connection by reason to this object as Date object.
String	<a href="#"><u>getLastConnectRejectedByReasonTimeString</u></a> (int reason) Get date and time of last rejected connection by reason to this object as formatted string.
long	<a href="#"><u>getLastConnectRejectedStamp</u></a> () Get time (milliseconds) of the last rejected connection to this object.
String	<a href="#"><u>getLastConnectRejectedStampString</u></a> () Get time (milliseconds) of the last rejected connection to this object as formatted string.
java.util.Date	<a href="#"><u>getLastConnectRejectedTime</u></a> () Get date and time of last rejected connection to this object as Date object.
String	<a href="#"><u>getLastConnectRejectedTimeString</u></a> () Get date and time of last rejected connection to this object as formatted string.
long	<a href="#"><u>getLastDisconnectStamp</u></a> () Get time (milliseconds) of the last disconnected connection to this object.
String	<a href="#"><u>getLastDisconnectStampString</u></a> () Get time (milliseconds) of the last disconnected connection to this object as formatted string.
java.util.Date	<a href="#"><u>getLastDisconnectTime</u></a> () Get date and time of last disconnected connection to this object as Date object.
String	<a href="#"><u>getLastDisconnectTimeString</u></a> () Get date and time of last disconnected connection to this object as Date object as formatted string.
long	<a href="#"><u>getTotal</u></a> () Get total number of connection attempts to this object.
long	<a href="#"><u>getTotalAccepted</u></a> () Get total number of accepted connections to this object.
long	<a href="#"><u>getTotalRejected</u></a> () Get total number of rejected connections to this object.
void	<a href="#"><u>incrementAccept</u></a> (ConnectionHolder connectionHolder, java.util.Date date, long stamp, byte[] license) Increment accepted connections.
void	<a href="#"><u>incrementReject</u></a> (ConnectionHolder connectionHolder, int reason, java.util.Date date, long stamp, byte[] license) Increment reject connection.
void	<a href="#"><u>rejectConnection</u></a> (ConnectionHolder connectionHolder, int reason, byte[] license) Reject connection.
void	<a href="#"><u>removeConnectionListener</u></a> (IConnectionNotify connectionNotify) Remove connection listener
void	<a href="#"><u>setConnectionValidator</u></a> (IConnectionValidator connectionValidator) Set the connection validator.

boolean

[validateNewConnection](#)(ConnectionHolder connectionHolder, byte[] license)

Validate a new connection.

Methods inherited from class java.lang.Object

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

## Constructors

### ConnectionCounter

```
public ConnectionCounter()
```

Create empty ConnectionCounter

## Methods

### addConnectionListener

```
public void addConnectionListener(IConnectionNotify connectionNotify)
```

Add a connection listener. Receives following events: onClientConnect, onClientDisconnect, onClientAccept and onClientReject.

**Parameters:**

connectionNotify - connection listener

### removeConnectionListener

```
public void removeConnectionListener(IConnectionNotify connectionNotify)
```

Remove connection listener

**Parameters:**

connectionNotify - connection listener

### setConnectionValidator

```
public void setConnectionValidator(IConnectionValidator connectionValidator)
```

Set the connection validator.

**Parameters:**

connectionValidator - connection validator

### validateNewConnection

```
public boolean validateNewConnection(ConnectionHolder connectionHolder,
    byte[] license)
```

Validate a new connection.

**NOTE:** This is a private internal call. Server will not work properly if this method is not called properly using internal security mechanism.

(continued from last page)

**Parameters:**

connectionHolder - connection holder  
license - security data

**Returns:**

true if connection is accepted

---

## acceptConnection

```
public void acceptConnection(ConnectionHolder connectionHolder,  
    byte[] license)
```

Accept a new connection.

**NOTE:** This is a private internal call. Server will not work properly if this method is not called properly using internal security mechanism.

**Parameters:**

connectionHolder - connection holder  
license - security data

---

## incrementAccept

```
public void incrementAccept(ConnectionHolder connectionHolder,  
    java.util.Date date,  
    long stamp,  
    byte[] license)
```

Increment accepted connections.

**NOTE:** This is a private internal call. Server will not work properly if this method is not called properly using internal security mechanism.

**Parameters:**

connectionHolder - connection holder  
date - date the connection occurred  
stamp - time stamp connection occurred (milliseconds)  
license - security data

---

## rejectConnection

```
public void rejectConnection(ConnectionHolder connectionHolder,  
    int reason,  
    byte[] license)
```

Reject connection.

**NOTE:** This is a private internal call. Server will not work properly if this method is not called properly using internal security mechanism.

**Parameters:**

connectionHolder - connection holder  
reason - reason the connection was refused. See REJECTREASON\_\*  
license - security data

---



(continued from last page)

## incrementReject

```
public void incrementReject(ConnectionHolder connectionHolder,  
    int reason,  
    java.util.Date date,  
    long stamp,  
    byte[] license)
```

Increment reject connection.

**NOTE:** This is a private internal call. Server will not work properly if this method is not called properly using internal security mechanism.

**Parameters:**

connectionHolder - connection holder  
reason - reason the connection was refused. See REJECTREASON\_\*  
date - date the connection occurred  
stamp - time stamp connection occurred (milliseconds)  
license - security data

---

## disconnect

```
public void disconnect(ConnectionHolder connectionHolder,  
    byte[] license)
```

Disconnect connection.

**NOTE:** This is a private internal call. Server will not work properly if this method is not called properly using internal security mechanism.

**Parameters:**

connectionHolder - connection holder  
license - security data

---

## decrement

```
public void decrement(ConnectionHolder connectionHolder,  
    boolean isValid,  
    java.util.Date date,  
    long stamp,  
    byte[] license)
```

Decrement connection counters.

**NOTE:** This is a private internal call. Server will not work properly if this method is not called properly using internal security mechanism.

**Parameters:**

connectionHolder - connection holder  
isValid - is a valid connection  
date - date the connection occurred  
stamp - time stamp connection occurred (milliseconds)  
license - security data

---

## getCurrent

```
public long getCurrent()
```

Get total number of client currently connected to this object.

**Returns:**

total number of client currently connected to this object

## getLastConnectAcceptedStamp

```
public long getLastConnectAcceptedStamp()
```

Get time (milliseconds) of the last conenction to this object.

**Returns:**

time (milliseconds) of the last conenction to this object

---

## getLastConnectAcceptedStampString

```
public String getLastConnectAcceptedStampString()
```

Get time (milliseconds) of the last conenction to this object as formatted string.

**Returns:**

time (milliseconds) of the last conenction to this object as formatted string

---

## getLastConnectAcceptedTime

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

Get time (milliseconds) of the last accepeted conenction to this object.

**Returns:**

time (milliseconds) of the last accepeted conenction to this object

---

## getLastConnectAcceptedTimeString

```
public String getLastConnectAcceptedTimeString()
```

Get time (milliseconds) of the last accepeted conenction to this object as formatted string.

**Returns:**

time (milliseconds) of the last accepeted conenction to this object as formatted string

---

## getLastConnectRejectedStamp

```
public long getLastConnectRejectedStamp()
```

Get time (milliseconds) of the last rejected conenction to this object.

**Returns:**

time (milliseconds) of the last rejected conenction to this object

---

## getLastConnectRejectedStampString

```
public String getLastConnectRejectedStampString()
```

Get time (milliseconds) of the last rejected conenction to this object as formatted string.

**Returns:**

time (milliseconds) of the last rejected conenction to this object as formatted string

---

## getLastConnectRejectedTime

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

---

(continued from last page)

Get date and time of last rejected connection to this object as Date object.

**Returns:**

date and time of last reject connection to this object as Date object

---

## getLastConnectRejectedTimeString

```
public String getLastConnectRejectedTimeString()
```

Get date and time of last rejected connection to this object as formatted string.

**Returns:**

date and time of last reject connection to this object as formatted string

---

## getLastConnectRejectedByReasonStampString

```
public String getLastConnectRejectedByReasonStampString(int reason)
```

Get time (milliseconds) of the last rejected connection by reason to this object as formatted string.

**Parameters:**

reason - reason, see REJECTREASON\_\*

**Returns:**

time (milliseconds) of the last rejected connection by reason to this object as formatted string

---

## getLastConnectRejectedByReasonStamp

```
public long getLastConnectRejectedByReasonStamp(int reason)
```

Get time (milliseconds) of the last rejected connection by reason to this object.

**Parameters:**

reason - reason, see REJECTREASON\_\*

**Returns:**

time (milliseconds) of the last rejected connection by reason to this object

---

## getLastConnectRejectedByReasonTime

```
public java.util.Date getLastConnectRejectedByReasonTime(int reason)
```

Get date and time of last rejected connection by reason to this object as Date object.

**Parameters:**

reason - reason, see REJECTREASON\_\*

**Returns:**

date and time of last reject connection by reason to this object as Date object

---

## getLastConnectRejectedByReasonTimeString

```
public String getLastConnectRejectedByReasonTimeString(int reason)
```

Get date and time of last rejected connection by reason to this object as formatted string.

**Parameters:**

reason - reason, see REJECTREASON\_\*

(continued from last page)

**Returns:**

date and time of last rejected connection by reason to this object as formatted string

---

**getLastDisconnectStampString**

```
public String getLastDisconnectStampString()
```

Get time (milliseconds) of the last disconnected connection to this object as formatted string.

**Returns:**

time (milliseconds) of the last disconnected connection to this object as formatted string

---

**getLastDisconnectStamp**

```
public long getLastDisconnectStamp()
```

Get time (milliseconds) of the last disconnected connection to this object.

**Returns:**

time (milliseconds) of the last disconnected connection to this object

---

**getLastDisconnectTime**

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

Get date and time of last disconnected connection to this object as Date object.

**Returns:**

date and time of last disconnected connection to this object as Date object

---

**getLastDisconnectTimeString**

```
public String getLastDisconnectTimeString()
```

Get date and time of last disconnected connection to this object as Date object as formatted string.

**Returns:**

date and time of last disconnected connection to this object as Date object as formatted string

---

**getTotal**

```
public long getTotal()
```

Get total number of connection attempts to this object.

**Returns:**

total number of connection attempts to this object

---

**getTotalAccepted**

```
public long getTotalAccepted()
```

Get total number of accepted connections to this object.

**Returns:**

total number of accepted connections to this object

## getTotalRejected

```
public long getTotalRejected()
```

Get total number of rejected connections to this object.

**Returns:**

total number of rejected connections to this object

## com.wowza.wms.client Interface IClient

public interface **IClient**  
extends

IClient: public interface to Client object.

### Field Summary

public static final	<a href="#">AUDIOSAMPLE_ACCESS_ALL</a> Value: *
public static final	<a href="#">AUDIOSAMPLE_ACCESS_NONE</a> Value:
public static final	<a href="#">READ_ACCESS_ALL</a> Value: *
public static final	<a href="#">READ_ACCESS_NONE</a> Value:
public static final	<a href="#">VIDEOSAMPLE_ACCESS_ALL</a> Value: *
public static final	<a href="#">VIDEOSAMPLE_ACCESS_NONE</a> Value:
public static final	<a href="#">WRITE_ACCESS_ALL</a> Value: *
public static final	<a href="#">WRITE_ACCESS_NONE</a> Value:

### Method Summary

void	<a href="#">acceptConnection</a> ( ) Accept connection
void	<a href="#">acceptConnection</a> ( <a href="#">AMFData</a> successObj) Accept connection
void	<a href="#">acceptConnection</a> (String successStr) Accept connection
void	<a href="#">addAcceptConnectionAttribute</a> (String key, <a href="#">AMFDataObj</a> item) Add and attribute to the resultObj that gets passed back to the client on successful connection

void	<a href="#"><u>addAcceptConnectionAttribute</u></a> (String key, String item) Add and attribute to the resultObj that gets passed back to the client on successful connection
void	<a href="#"><u>call</u></a> (String handlerName) Simplified call client method/handler call.
void	<a href="#"><u>call</u></a> (String handlerName, <a href="#"><u>IModuleCallResult</u></a> resultObj, Object[] params) Call client method/handler.
void	<a href="#"><u>clearFastPlaySettings</u></a> ( ) Force clear the fastPlay settings
void	<a href="#"><u>fcSubscribe</u></a> (String streamName) Subscribe to a live stream (for live stream repeater to start start from edge to origin)
void	<a href="#"><u>fcSubscribe</u></a> (String streamName, String mediaCasterType) Subscribe to a live stream (use a particular mediaCasterType)
void	<a href="#"><u>fcUnSubscribe</u></a> (String streamName) UnSubscribe from a stream
void	<a href="#"><u>fcUnSubscribeAll</u></a> ( ) UnSubscribe to all streams that this client is current subscribed to
<a href="#"><u>IApplicationInstance</u></a>	<a href="#"><u>getAppInstance</u></a> ( ) Get parent applicationInstance.
<a href="#"><u>IApplication</u></a>	<a href="#"><u>getApplication</u></a> ( ) Get parent application.
int	<a href="#"><u>getBufferTime</u></a> ( ) Get default buffer time for newly created mediaStream objects
int	<a href="#"><u>getClientId</u></a> ( ) Get client id.
long	<a href="#"><u>getConnectTime</u></a> ( ) Get time in milliseconds the client connected to the server.
String	<a href="#"><u>getDateStarted</u></a> ( ) Get date and time of client connection
<a href="#"><u>ElapsedTimer</u></a>	<a href="#"><u>getElapsedTime</u></a> ( ) Get elapsed time client has been connected.
<a href="#"><u>FastPlaySettings</u></a>	<a href="#"><u>getFastPlaySettings</u></a> ( ) Get the current fastPlay settings.
String	<a href="#"><u>getFlashVer</u></a> ( ) Get client flash version (same as FMS getAgent())
int	<a href="#"><u>getIdleFrequency</u></a> ( ) Get client idle frequency (milliseconds)
String	<a href="#"><u>getIp</u></a> ( ) Client ip address
long	<a href="#"><u>getLastValidateTime</u></a> ( ) Get last time (millisecond) the connection was validated with a ping

int	<a href="#"><u>getLiveRepeaterCapabilities()</u></a> Get the live repeater capabilities of this connection
String	<a href="#"><u>getLiveStreamPacketizerList()</u></a> Get the comma separated list of LiveStreamPacketizers names being used by this client (see conf/LiveStreamPacketizers.xml)
String	<a href="#"><u>getLiveStreamTranscoderList()</u></a> Get the comma separated list of LiveStreamTranscoders names being used by this client (see conf/LiveStreamTranscoders.xml)
int	<a href="#"><u>getMaximumPendingWriteBytes()</u></a> Get maximum number a bytes a client connection can have waiting to be sent before the connection is terminated.
int	<a href="#"><u>getMaximumSetBufferTime()</u></a> Get maximum number of milliseconds allowed for the NetStream.setBufferTime(secs) call.
<a href="#"><u>IOPerformanceCounter</u></a>	<a href="#"><u>getMediaIOPerformanceCounter()</u></a> Get performance counter for media bytes transferred to this client.
int	<a href="#"><u>getObjectEncoding()</u></a> Get the object encoding level
String	<a href="#"><u>getPageUrl()</u></a> Get the pageUrl for this connection.
long	<a href="#"><u>getPingRoundTripTime()</u></a> Get turn around time (milliseconds) of last ping request
int	<a href="#"><u>getPingTimeout()</u></a> Get the ping timeout (milliseconds)
java.util.List	<a href="#"><u>getPlayStreams()</u></a> Get a collection of all play streams.
<a href="#"><u>WMSProperties</u></a>	<a href="#"><u>getProperties()</u></a> Get client properties
int	<a href="#"><u>getProtocol()</u></a> Get connection protocol (1 = RTMP, 3 = RTMPT)
java.util.List	<a href="#"><u>getPublishStreams()</u></a> Get a collection of publish streams.
String	<a href="#"><u>getQueryStr()</u></a> Get the query string part of the connection string.
String	<a href="#"><u>getReferrer()</u></a> Get the referrer data for this connection.
String	<a href="#"><u>getRepeaterOriginUrl()</u></a> Get the origin URL used by the Live Stream Repeater
<a href="#"><u>AMFObj</u></a>	<a href="#"><u>getRespAMFAudioObj()</u></a> ( <a href="#"><u>IMediaStream</u></a> stream) Get the audio response AMFObj for a given mediaStream
<a href="#"><u>AMFObj</u></a>	<a href="#"><u>getRespAMFDataObj()</u></a> ( <a href="#"><u>IMediaStream</u></a> stream) Get the data response AMFObj for a given mediaStream



<a href="#">AMFObj</a>	<a href="#">getRespAMFVideoObj</a> ( <a href="#">IMediaStream</a> stream) Get the video response AMFObj for a given mediaStream
<a href="#">ResponseFunctions</a>	<a href="#">getRespFunctions</a> () Get client responseFunctions object.
<a href="#">AMFObj</a>	<a href="#">getResponseAMFObj</a> (int index) Get the response channel AMFObj for channel index.
<a href="#">RTPStream</a>	<a href="#">getRTPStream</a> () If this client was created due to an RTP connection to the server return the underlying RTPStream object
<a href="#">HostPort</a>	<a href="#">getServerHostPort</a> () Get the hostPort object for the connection that is servicing this client
String	<a href="#">getSharedObjectReadAccess</a> () Get the shared object read access value.
String	<a href="#">getSharedObjectWriteAccess</a> () Get the shared object write access value.
String	<a href="#">getStreamAudioSampleAccess</a> () Get the audio sample access value.
java.io.File	<a href="#">getStreamFile</a> (String streamName) Get File object for stream with given name.
java.io.File	<a href="#">getStreamFile</a> (String streamName, String streamExt) Get File object for stream with given name and extension.
java.io.File	<a href="#">getStreamFile</a> (String streamName, String streamExt, boolean doCreateFolder) Get File object for stream with given name and extension.
String	<a href="#">getStreamReadAccess</a> () Get the stream read access value.
String	<a href="#">getStreamType</a> () Get default streamType
String	<a href="#">getStreamVideoSampleAccess</a> () Get the video sample access value.
String	<a href="#">getStreamWriteAccess</a> () Get the stream write access value.
String	<a href="#">getTimeRunning</a> () Get elapsed time of connection
double	<a href="#">getTimeRunningSeconds</a> () Get time running in seconds
<a href="#">IOPerformanceCounter</a>	<a href="#">getTotalIOPerformanceCounter</a> () Get performance counter for all bytes transferred to this client.
String	<a href="#">getUri</a> () Get the full URI of the connection string

<a href="#">IVHost</a>	<a href="#">getVHost()</a> Get parent vHost
ClientWriteListener	<a href="#">getWriteListener()</a> Object that tracks write operations
boolean	<a href="#">isAcceptConnection()</a> Is auto accept connection
boolean	<a href="#">isConnected()</a> Is this client connected
boolean	<a href="#">isEncrypted()</a> Is this connection encrypted (RTMPE or RTMPTE)
boolean	<a href="#">isFlashMediaLiveEncoder()</a> Returns true if this connection is the Flash Media Live Encoder
boolean	<a href="#">isFlashVersion10()</a> Returns true if the Flash version is equal or greater than 10.x.x.x
boolean	<a href="#">isFlashVersion90115()</a> Returns true if the Flash version is equal or greater than 9.0.115.x
boolean	<a href="#">isFlashVersionH264Capable()</a> Returns true if the connected client is capable of playing H.264 video (Flash player 9.0.45.x or greater)
boolean	<a href="#">isLiveRepeater()</a> Returns true if this connection is from the live stream repeater
boolean	<a href="#">isObjectEncodingAMF0()</a> Is the object encoding for this client AMF0
boolean	<a href="#">isObjectEncodingAMF3()</a> Is the object encoding for this client AMF3
boolean	<a href="#">isSecure()</a> Is this connection protected by either SSL or encryption (RTMPE, RTMPTE, RTMPS)
boolean	<a href="#">isSSL()</a> Is this connection SSL (RTMPS)
boolean	<a href="#">isValidFMLEConnections()</a> Returns true if validating FMLE connection (default is false)
int	<a href="#">ping(IModulePingResult pingResult)</a> Ping client.
void	<a href="#">redirectConnection(String url)</a> Redirection connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected and info.ex.code of 302)
void	<a href="#">redirectConnection(String url, String description)</a> Redirection connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected and info.ex.code of 302)
void	<a href="#">redirectConnection(String url, String description, <a href="#">AMFData</a> errorObj)</a> Redirection connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected and info.ex.code of 302).

void	<a href="#"><u>redirectConnection</u></a> (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	<a href="#"><u>rejectConnection</u></a> ( ) Reject connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected)
void	<a href="#"><u>rejectConnection</u></a> ( <a href="#"><u>AMFData</u></a> errorObj) Reject connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected)
void	<a href="#"><u>rejectConnection</u></a> (String errorStr) Reject connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected)
void	<a href="#"><u>rejectConnection</u></a> (String description, <a href="#"><u>AMFData</u></a> errorObj) Reject connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected)
void	<a href="#"><u>rejectConnection</u></a> (String description, String errorStr) Reject connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected)
void	<a href="#"><u>reparentClient</u></a> ( <a href="#"><u>IVHost</u></a> vhost) Move a client object to a new vhost.
void	<a href="#"><u>setAcceptConnection</u></a> (boolean acceptConnection) Set the default for accept connection
void	<a href="#"><u>setAcceptConnectionDescription</u></a> (String description) Call this method from onConnect to set the info.description property returned in NetConnection onStatus handler
void	<a href="#"><u>setAcceptConnectionExObj</u></a> ( <a href="#"><u>AMFDataObj</u></a> acceptConnectionExObj) Call this method from onConnect to set the info.ex property returned in NetConnection onStatus handler
void	<a href="#"><u>setAcceptConnectionObj</u></a> ( <a href="#"><u>AMFData</u></a> 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	<a href="#"><u>setBufferTime</u></a> (int bufferTime) Set default buffer time for newly created mediaStream objects
void	<a href="#"><u>setFastPlaySettings</u></a> ( <a href="#"><u>FastPlaySettings</u></a> fastPlaySettings) Set fastPlay settings
void	<a href="#"><u>setFlashVer</u></a> (String flashVer) Set client flash version
void	<a href="#"><u>setIdleFrequency</u></a> (int idleFrequency) Set client idle frequency (milliseconds)
void	<a href="#"><u>setLastValidateTime</u></a> (long lastValidateTime) Set the last time (milliseconds) the connection was validated with ping
void	<a href="#"><u>setLiveRepeaterCapabilities</u></a> (int liveRepeaterCapabilities) Set the live repeater capabilities of this connection

void	<a href="#"><u>setLiveStreamPacketizerList</u></a> (String liveStreamPacketizerList) Set the comma separated list of LiveStreamPacketizers names being used by this client (see conf/LiveStreamPacketizers.xml)
void	<a href="#"><u>setLiveStreamTranscoderList</u></a> (String liveStreamTranscoderList) Set the comma separated list of LiveStreamTranscoders names being used by this client (see conf/LiveStreamTranscoders.xml)
void	<a href="#"><u>setObjectEncoding</u></a> (int objectEncoding) Set the object encoding level
void	<a href="#"><u>setRepeaterOriginUrl</u></a> (String repeaterOriginUrl) Set the origin URL used by the Live Stream Repeater
void	<a href="#"><u>setSharedObjectReadAccess</u></a> (String sharedObjectReadAccess) Set the shared object read access value.
void	<a href="#"><u>setSharedObjectWriteAccess</u></a> (String sharedObjectWriteAccess) Set the shared object write access value.
void	<a href="#"><u>setShutdownClient</u></a> (boolean shutdownClient) Gracefully and forcefully shutdown a client.
void	<a href="#"><u>setStreamAudioSampleAccess</u></a> (String audioSampleAccess) Set the stream audio sample access value.
void	<a href="#"><u>setStreamReadAccess</u></a> (String streamReadAccess) Set the stream object read access value.
void	<a href="#"><u>setStreamType</u></a> (String streamType) Set default streamType
void	<a href="#"><u>setStreamVideoSampleAccess</u></a> (String videoSampleAccess) Set the stream video sample access value.
void	<a href="#"><u>setStreamWriteAccess</u></a> (String streamWriteAccess) Set the stream object write access value.
void	<a href="#"><u>setThreadContext</u></a> ( ) Set the thread logging context to this client
void	<a href="#"><u>setValidateFMLEConnections</u></a> (boolean validateFMLEConnections) Returns true if validating FMLE connection (default is false)
void	<a href="#"><u>shutdownClient</u></a> ( ) Gracefully shutdown a client.
int	<a href="#"><u>testFlashVersion</u></a> (int[] version) Test to see if the connected client flash version is equal to or greater than a given value.
void	<a href="#"><u>touch</u></a> ( ) 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 then

---

## isObjectEncodingAMF3

```
public boolean isObjectEncodingAMF3()
```

Is the object encoding for this client AMF3

**Returns:**

true of the object encoding for this client is AMF3

---

## isObjectEncodingAMF0

```
public boolean isObjectEncodingAMF0()
```

Is the object encoding for this client AMF0

**Returns:**

true of the object encoding for this client is AMF0

---

## setObjectEncoding

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

Set the object encoding level

**Parameters:**

objectEncoding - object encoding level

---

## getObjectEncoding

```
public int getObjectEncoding()
```

Get the object encoding level

**Returns:**

object encoding level

---

## setAcceptConnectionObj

```
public void setAcceptConnectionObj(AMFData acceptConnectionObj)
```

Call this method from onConnect to add an "application" property to the info object that is passed back to the Flash client in the onStatus handler.

**Parameters:**

acceptConnectionObj - connection object

---

## setAcceptConnectionDescription

```
public void setAcceptConnectionDescription(String description)
```

Call this method from onConnect to set the info.description property returned in NetConnection onStatus handler

**Parameters:**

description - description

---

## setAcceptConnectionExObj

```
public void setAcceptConnectionExObj(AMFDataObj acceptConnectionExObj)
```

Call this method from onConnect to set the info.ex property returned in NetConnection onStatus handler

**Parameters:**

acceptConnectionExObj - exObj AMFData item return in info.ex property of NetConnection onStatus handler

---

(continued from last page)

## getRTPStream

```
public RTPStream getRTPStream( )
```

If this client was created due to an RTP connection to the server return the underlying RTPStream object

**Returns:**

underlying RTPStream object

---

## setAcceptConnection

```
public void setAcceptConnection(boolean acceptConnection)
```

Set the default for accept connection

**Parameters:**

acceptConnection - default for accept connection

---

## setShutdownClient

```
public void setShutdownClient(boolean shutdownClient)
```

Gracefully and forcefully shutdown a client.

**Parameters:**

shutdownClient - set to true to gracefully and forcefully shutdown a client

---

## reparentClient

```
public void reparentClient(IVHost vhost)
```

Move a client object to a new vhost. This can only be done right after the handshake process has completed. See IVHostNotify.onVHostClientConnect.

**Parameters:**

vhost - new vhost

---

## getMaximumSetBufferTime

```
public int getMaximumSetBufferTime( )
```

Get maximum number of milliseconds allowed for the NetStream.setBufferTime(secs) call. If set to zero this feature is turned off.

**Returns:**

maximum number of milliseconds allowed for the NetStream.setBufferTime(secs) call

---

## getMaximumPendingWriteBytes

```
public int getMaximumPendingWriteBytes( )
```

Get maximum number a bytes a client connection can have waiting to be sent before the connection is terminated. If set to zero this feature is turned off.

**Returns:**

maximum number a bytes a client connection can have waiting to be sent before the connection is terminated

(continued from last page)

---

## getLiveStreamPacketizerList

```
public String getLiveStreamPacketizerList()
```

Get the comma separated list of LiveStreamPacketizers names being used by this client (see conf/LiveStreamPacketizers.xml)

**Returns:**

comma separated list of LiveStreamPacketizers names

---

## setLiveStreamPacketizerList

```
public void setLiveStreamPacketizerList(String liveStreamPacketizerList)
```

Set the comma separated list of LiveStreamPacketizers names being used by this client (see conf/LiveStreamPacketizers.xml)

**Parameters:**

liveStreamPacketizerList - comma separated list of LiveStreamPacketizers names

---

## getLiveStreamTranscoderList

```
public String getLiveStreamTranscoderList()
```

Get the comma separated list of LiveStreamTranscoders names being used by this client (see conf/LiveStreamTranscoders.xml)

**Returns:**

comma separated list of LiveStreamTranscoders names

---

## setLiveStreamTranscoderList

```
public void setLiveStreamTranscoderList(String liveStreamTranscoderList)
```

Set the comma separated list of LiveStreamTranscoders names being used by this client (see conf/LiveStreamTranscoders.xml)

**Parameters:**

liveStreamTranscoderList - comma separated list of LiveStreamTranscoders names

---

## getElapsedTime

```
public ElapsedTimer getElapsedTime()
```

Get elapsed time client has been connected.

**Returns:**

elapsed time client has been connected.

---

## getLiveRepeaterCapabilities

```
public int getLiveRepeaterCapabilities()
```

Get the live repeater capabilities of this connection

**Returns:**

live repeater capabilities of this connection

---

## setLiveRepeaterCapabilities

```
public void setLiveRepeaterCapabilities(int liveRepeaterCapabilities)
```

---

(continued from last page)

Set the live repeater capabilities of this connection

**Parameters:**

liveRepeaterCapabilities - live repeater capabilities of this connection

---

## fcSubscribe

```
public void fcSubscribe(String streamName)
```

Subscribe to a live stream (for live stream repeater to start start from edge to origin)

**Parameters:**

streamName - stream name

---

## fcSubscribe

```
public void fcSubscribe(String streamName,  
String mediaCasterType)
```

Subscribe to a live stream (use a particular mediaCasterType)

**Parameters:**

streamName - stream name

mediaCasterType - media caster type name

---

## fcUnSubscribeAll

```
public void fcUnSubscribeAll()
```

UnSubscribe to all streams that this client is current subscribed to

---

## fcUnSubscribe

```
public void fcUnSubscribe(String streamName)
```

UnSubscribe from a stream

**Parameters:**

streamName - stream name

---

## isValidFMLEConnections

```
public boolean isValidFMLEConnections()
```

Returns true if validating FMLE connection (default is false)

**Returns:**

true if validating FMLE connection

---

## setValidateFMLEConnections

```
public void setValidateFMLEConnections(boolean validateFMLEConnections)
```

Returns true if validating FMLE connection (default is false)

**Parameters:**

validateFMLEConnections - true if validating FMLE connection

(continued from last page)

## setThreadContext

```
public void setThreadContext( )
```

Set the thread logging context to this client

## 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	<a href="#">onClientAccept</a> ( <a href="#">IClient</a> client) Triggered when client connection accepted
void	<a href="#">onClientConnect</a> ( <a href="#">IClient</a> client) Triggered when client attempt connection
void	<a href="#">onClientDisconnect</a> ( <a href="#">IClient</a> client) Triggered when client disconnected
void	<a href="#">onClientReject</a> ( <a href="#">IClient</a> 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	<a href="#">onAcceptConnection</a> ( <a href="#">ConnectionCounter</a> connectionCounter, <a href="#">ConnectionHolder</a> connectionHolder, java.util.Date date, long stamp) Triggered when client connection accepted
void	<a href="#">onDisconnect</a> ( <a href="#">ConnectionCounter</a> connectionCounter, <a href="#">ConnectionHolder</a> connectionHolder, boolean isValid, java.util.Date date, long stamp) Triggered when client disconnected
void	<a href="#">onRejectConnection</a> ( <a href="#">ConnectionCounter</a> connectionCounter, <a href="#">ConnectionHolder</a> 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	<a href="#">validateConnection</a> ( <a href="#">ConnectionCounter</a> connectionCounter, <a href="#">ConnectionHolder</a> 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	<a href="#"><code>onAcceptLicense</code></a> (LicenseCounter licenseCounter, LicenseHolder licenseHolder, java.util.Date date, long stamp) Triggered when license accepted
void	<a href="#"><code>onDisconnect</code></a> (LicenseCounter licenseCounter, LicenseHolder licenseHolder, boolean isValid, java.util.Date date, long stamp) Triggered when client disconnected
void	<a href="#"><code>onRejectLicense</code></a> (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	<a href="#">validateLicense</a> (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	<a href="#">DefaultDvrStreamVersionHandler()</a>
--------	--

## Method Summary

<a href="#">IDvrStreamStore</a>	<a href="#">determineExistingStoreForPlaying</a> ( <a href="#">IDvrStreamManager</a> dvrMgr, String baseStreamName)
<a href="#">IDvrStreamStore</a>	<a href="#">determineExistingStoreForRecording</a> ( <a href="#">IDvrStreamManager</a> dvrMgr, String baseStreamName)
String	<a href="#">getArchiveStrategy</a> ( <a href="#">IDvrStreamManager</a> dvrMgr, String baseStreamName)
boolean	<a href="#">handleArchivedStream</a> ( <a href="#">IDvrStreamManager</a> dvrMgr, String baseStreamName, String vStreamName, java.util.SortedSet versions, DvrManifestHolder manifestHolder)
boolean	<a href="#">shouldDeleteArchivedStream</a> ( <a href="#">IDvrStreamManager</a> dvrMgr, <a href="#">IDvrStreamStore</a> store)
boolean	<a href="#">shouldLoadArchivedStream</a> ( <a href="#">IDvrStreamManager</a> 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)
```

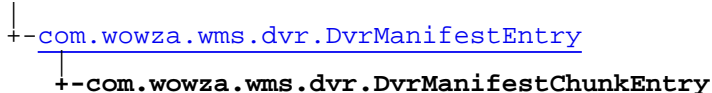
### shouldDeleteArchivedStream

```
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	<a href="#">artifact</a>
protected	<a href="#">encryptions</a>

#### 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	<a href="#">DvrManifestChunkEntry</a> (int type, int index, long dvrStart, long dvrStop, long packetTime, long utcTime, DvrChunkArtifact artifact) Constructor
public	<a href="#">DvrManifestChunkEntry</a> (int type, int index, long dvrStart, long dvrStop, long packetTime, long utcTime, DvrChunkArtifact artifact, DvrEncryptionInfoHolder encryptions) Constructor

#### Method Summary

String	<a href="#">getArtifactsTextRepresentation</a> ()
DvrChunkArtifact	<a href="#">getDvrArtifact</a> () Get DVR artifact reference.
DvrEncryptionInfoHolder	<a href="#">getEncryptions</a> () Get associated encryptions.
String	<a href="#">getEncryptionsTextRepresentation</a> ()
String	<a href="#">getManifestRepresentation</a> ()

void	<a href="#">serialize</a> (java.io.DataOutputStream out)
void	<a href="#">setEncryptions</a> (DvrEncryptionInfoHolder encryptions) Set associated encryptions.
String	<a href="#">toString</a> ()

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()
```

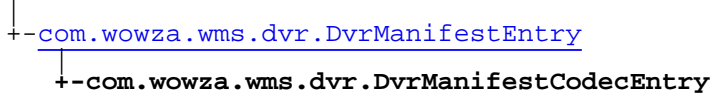
---

## **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	<a href="#">DvrManifestCodecEntry</a> (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	<a href="#">getAudioCodec</a> () Get audio codec information
String	<a href="#">getManifestRepresentation</a> ()
com.wowza.wms.media.model.MediaCodecInfoVideo	<a href="#">getVideoCodec</a> () Get video codec information
void	<a href="#">serialize</a> (java.io.DataOutputStream out)
String	<a href="#">toString</a> ()

### 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	<a href="#">dvrStart</a>
protected	<a href="#">dvrStop</a>
protected	<a href="#">index</a>
public static final	<a href="#">MANIFESTFILE_KEY_ARTIFACT</a> Value: <b>artifact</b>
public static final	<a href="#">MANIFESTFILE_KEY_AUDIO_CODEC</a> Value: <b>aCodec</b>
public static final	<a href="#">MANIFESTFILE_KEY_CHUNKINDEX</a> Value: <b>chunkIndex</b>
public static final	<a href="#">MANIFESTFILE_KEY_DVRTIME</a> Value: <b>dvrTime</b>
public static final	<a href="#">MANIFESTFILE_KEY_ENCRYPTIONS</a> Value: <b>enc</b>
public static final	<a href="#">MANIFESTFILE_KEY_INDEX</a> Value: <b>index</b>
public static final	<a href="#">MANIFESTFILE_KEY_METADATA</a> Value: <b>metadata</b>
public static final	<a href="#">MANIFESTFILE_KEY_NAME</a> Value: <b>name</b>

public static final	<a href="#">MANIFESTFILE_KEY_PACKETTIME</a> Value: <b>packetTime</b>
public static final	<a href="#">MANIFESTFILE_KEY_SIZE</a> Value: <b>size</b>
public static final	<a href="#">MANIFESTFILE_KEY_START</a> Value: <b>start</b>
public static final	<a href="#">MANIFESTFILE_KEY_STOP</a> Value: <b>stop</b>
public static final	<a href="#">MANIFESTFILE_KEY_TYPE</a> Value: <b>type</b>
public static final	<a href="#">MANIFESTFILE_KEY_UTCTIME</a> Value: <b>utcTime</b>
public static final	<a href="#">MANIFESTFILE_KEY_VIDEO_CODEC</a> Value: <b>vCodec</b>
protected	<a href="#">packetTime</a>
protected static final	<a href="#">SERIALIZE_CURRENT_VERSION</a> Value: <b>2</b>
protected	<a href="#">type</a>
protected	<a href="#">utcTime</a>

## Constructor Summary

public	<a href="#">DvrManifestEntry</a> (int type, int index, long dvrStart, long dvrStop, long packetTime, long utcTime) Constructor Valid types include: <a href="#">IVHost.CONTENTTYPE_AUDIO</a> , <a href="#">IVHost.CONTENTTYPE_VIDEO</a> , <a href="#">IVHost.CONTENTTYPE_DATA</a> , <a href="#">IDvrManifest.ON_METADATA_TYPE</a> , <a href="#">IDvrManifest.CODEC_TYPE</a> , or <a href="#">IDvrManifest.TIME_MAP_TYPE</a>
--------	--

## Method Summary

String	<a href="#">encodeBytes</a> (byte[] bytes) Encode string of bytes as Base64.
String	<a href="#">getCommonInitialTextRepString</a> ()
long	<a href="#">getDuration</a> () Get duration
int	<a href="#">getIndex</a> () Get manifest index.

abstract String	<a href="#"><code>getManifestRepresentation()</code></a> Get textual representation of record for textual manifest usage.
long	<a href="#"><code>getPacketStartTime()</code></a> Get chunk start timecode in packetTime units.
long	<a href="#"><code>getStartTimecode()</code></a> Get start timecode.
long	<a href="#"><code>getStopTimecode()</code></a> Get stop timecode.
int	<a href="#"><code>getType()</code></a> Get type of manifest record.
long	<a href="#"><code>getUtcStartTime()</code></a> Get chunk start timecode in UTC units.
byte[]	<a href="#"><code>serialize()</code></a> Serialize manifest record.
abstract void	<a href="#"><code>serialize(java.io.DataOutputStream out)</code></a> Serialize manifest record.

#### Methods inherited from class `java.lang.Object`

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

## Fields

### MANIFESTFILE\_KEY\_INDEX

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

Constant value: **index**

### MANIFESTFILE\_KEY\_TYPE

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

Constant value: **type**

### MANIFESTFILE\_KEY\_START

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

Constant value: **start**

### MANIFESTFILE\_KEY\_STOP

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

(continued from last page)

Constant value: **stop**

---

## MANIFESTFILE\_KEY\_SIZE

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

Constant value: **size**

---

## MANIFESTFILE\_KEY\_ARTIFACT

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

Constant value: **artifact**

---

## MANIFESTFILE\_KEY\_DVRTIME

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

Constant value: **dvrTime**

---

## MANIFESTFILE\_KEY\_CHUNKINDEX

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

Constant value: **chunkIndex**

---

## MANIFESTFILE\_KEY\_PACKETTIME

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

Constant value: **packetTime**

---

## MANIFESTFILE\_KEY\_UTCTIME

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

Constant value: **utcTime**

---

## MANIFESTFILE\_KEY\_NAME

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

Constant value: **name**

---

## MANIFESTFILE\_KEY\_AUDIO\_CODEC

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

Constant value: **aCodec**

(continued from last page)

---

## MANIFESTFILE\_KEY\_VIDEO\_CODEC

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

Constant value: **vCodec**

---

## MANIFESTFILE\_KEY\_ENCRYPTIONS

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

Constant value: **enc**

---

## MANIFESTFILE\_KEY\_METADATA

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

Constant value: **metadata**

---

## SERIALIZE\_CURRENT\_VERSION

```
protected static final byte SERIALIZE_CURRENT_VERSION
```

Constant value: **2**

---

## dvrStart

```
protected long dvrStart
```

---

## dvrStop

```
protected long dvrStop
```

---

## packetTime

```
protected long packetTime
```

---

## utcTime

```
protected long utcTime
```

---

## index

```
protected int index
```

---

(continued from last page)

## type

protected int **type**

## Constructors

### DvrManifestEntry

```
public DvrManifestEntry(int type,  
                        int index,  
                        long dvrStart,  
                        long dvrStop,  
                        long packetTime,  
                        long utcTime)
```

Constructor Valid types include: [IVHost.CONTENTTYPE\\_AUDIO](#), [IVHost.CONTENTTYPE\\_VIDEO](#), [IVHost.CONTENTTYPE\\_DATA](#), [IDvrManifest.ON\\_METADATA\\_TYPE](#), [IDvrManifest.CODEC\\_TYPE](#), or [IDvrManifest.TIME\\_MAP\\_TYPE](#)

#### Parameters:

type - type  
index - manifest index  
dvrStart - start time (ms in DVR time scale)  
dvrStop - stop time (ms in DVR time scale)  
packetTime  
utcTime

## Methods

### getStartTimecode

```
public long getStartTimecode()
```

Get start timecode. In milliseconds, DVR time base.

#### Returns:

start time

### getStopTimecode

```
public long getStopTimecode()
```

Get stop timecode. In milliseconds, DVR time base.

#### Returns:

stop time

### getPacketStartTime

```
public long getPacketStartTime()
```

Get chunk start timecode in packetTime units. In milliseconds.

#### Returns:

packet start time

(continued from last page)

## getUtcStartTime

```
public long getUtcStartTime()
```

Get chunk start timecode in UTC units. In milliseconds.

**Returns:**

UTC start time

## getDuration

```
public long getDuration()
```

Get duration

**Returns:**

duration in ms

## getType

```
public int getType()
```

Get type of manifest record. Valid types include: [IVHost.CONTENTTYPE\\_AUDIO](#), [IVHost.CONTENTTYPE\\_VIDEO](#), [IVHost.CONTENTTYPE\\_DATA](#), [IDvrManifest.ON\\_METADATA\\_TYPE](#), [IDvrManifest.CODEC\\_TYPE](#), or [IDvrManifest.TIME\\_MAP\\_TYPE](#)

**Returns:**

type

## getIndex

```
public int getIndex()
```

Get manifest index.

**Returns:**

index

## serialize

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

Serialize manifest record.

**Parameters:**

out - output stream

## serialize

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

Serialize manifest record.

**Returns:**

serialized data

---

(continued from last page)

## **getManifestRepresentation**

```
public abstract String getManifestRepresentation()
```

Get textual representation of record for textual manifest usage.

**Returns:**

textual representation of entry

---

## **getCommonInitialTextRepString**

```
protected String getCommonInitialTextRepString()
```

---

## **encodeBytes**

```
protected String encodeBytes(byte[] bytes)
```

Encode string of bytes as Base64. Provides check for null buffer or empty buffer.

**Parameters:**

bytes - buffer to encode.

**Returns:**

Base64 encoding or "".



com.wowza.wms.dvr

# Class DvrManifestEntryFactory



```
public class DvrManifestEntryFactory
extends Object
```

Deserialize manifest entry blobs.

## Constructor Summary

public	<a href="#">DvrManifestEntryFactory()</a>
--------	---

## Method Summary

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

#### 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	<a href="#">ranges</a>
--------	------------------------

### Constructor Summary

public	<a href="#">DvrManifestEntryRangeGroup()</a>
--------	--

### Method Summary

void	<a href="#">addRange(DvrManifestEntryRange range)</a> Add an index range to the group.
void	<a href="#">deserialize(byte[] data)</a> Deserialize.
boolean	<a href="#">isEmpty()</a> Determine if group of ranges is empty
boolean	<a href="#">isInRange(int type, int index)</a> Determine if index of given type is contained in the group of ranges.
byte[]	<a href="#">serialize()</a> Serialize range group
String	<a href="#">toString()</a>

### 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.

---

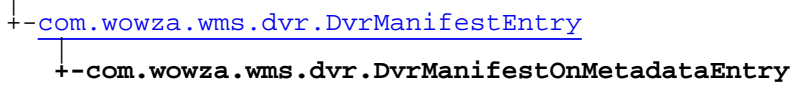
**toString**

```
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	<a href="#">artifact</a>
protected	<a href="#">data</a>

#### 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	<a href="#">DvrManifestOnMetadataEntry</a> (int index, long start, long packetTime, long utcTime, <a href="#">DvrChunkArtifact</a> artifact, byte[] data) Constructor
--------	--

#### Method Summary

String	<a href="#">getArtifactsTextRepresentation</a> ()
byte[]	<a href="#">getData</a> () Get onMetadata information.
<a href="#">DvrChunkArtifact</a>	<a href="#">getDvrArtifact</a> ()
String	<a href="#">getManifestRepresentation</a> ()
void	<a href="#">serialize</a> (java.io.DataOutputStream out)
String	<a href="#">toString</a> ()

#### 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	<a href="#">DvrManifestTimeMapEntry</a> (int index, int chunkIndex, long start, long packetTime, long utcTime, TimeMapRecord timeMap) Constructor
--------	--

### Method Summary

int	<a href="#">getChunkIndex</a> () Get chunk index that correlates to this time mapping
String	<a href="#">getManifestRepresentation</a> ()
TimeMapRecord	<a href="#">getTimeMapping</a> () Get time map record.
void	<a href="#">serialize</a> (java.io.DataOutputStream out)
String	<a href="#">toString</a> ()

### 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	<a href="#">expandEndTime</a> (long dvrEndTime)
long	<a href="#">expandStartTime</a> (long dvrStartTime)
long	<a href="#">getClosestStartTime</a> (long t)
<a href="#">DvrManifestEntry</a>	<a href="#">getFirstEntry</a> ()
int	<a href="#">getFirstIndex</a> ()
java.util.Map	<a href="#">getIndexMap</a> ()
<a href="#">DvrManifestEntry</a>	<a href="#">getLastLiveEntry</a> ()
<a href="#">DvrManifestEntry</a>	<a href="#">getLastRecordedEntry</a> ()
int	<a href="#">getLastRecordedIndex</a> ()
long	<a href="#">getLiveDuration</a> ()
java.util.List	<a href="#">getLiveEntries</a> ()
java.util.List	<a href="#">getLiveEntries</a> (long startTime)
java.util.List	<a href="#">getLiveEntriesWithLimit</a> (long t, int limit)
<a href="#">DvrManifestEntryRange</a>	<a href="#">getLiveRangeEndingBeforeTime</a> (long time)
<a href="#">DvrManifestEntryRange</a>	<a href="#">getLiveRangeEndingBeforeTime</a> (long time, boolean skipFirst)
java.util.List	<a href="#">getLiveTailEntries</a> (int index)
int	<a href="#">getNumberLiveEntries</a> (long dvrStart)
int	<a href="#">getNumberLiveEntries</a> (long dvrStart, long dvrEnd)

int	<a href="#"><u>getNumberRecordedEntries</u></a> (long dvrStart)
int	<a href="#"><u>getNumberRecordedEntries</u></a> (long dvrStart, long dvrEnd)
long	<a href="#"><u>getRecordedDuration</u></a> ()
java.util.List	<a href="#"><u>getRecordedEntries</u></a> ()
java.util.List	<a href="#"><u>getRecordedEntries</u></a> (long dvrStartTime)
java.util.List	<a href="#"><u>getRecordedEntries</u></a> (long dvrStartTime, long dvrEndTime)
java.util.List	<a href="#"><u>getRecordedEntriesInRange</u></a> (int startIndex, int endIndex)
java.util.List	<a href="#"><u>getRecordedEntriesWithLimit</u></a> (long t, int limit)
<a href="#"><u>DvrManifestEntry</u></a>	<a href="#"><u>getRecordedEntryByIndex</u></a> (int index)
<a href="#"><u>DvrManifestEntry</u></a>	<a href="#"><u>getRecordedEntryByTimeKey</u></a> (long t)
<a href="#"><u>DvrManifestEntry</u></a>	<a href="#"><u>getRecordedEntryStartingBeforeTime</u></a> (long t, boolean inclusive)
int	<a href="#"><u>getType</u></a> ()
boolean	<a href="#"><u>isEmpty</u></a> ()

## 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	<a href="#"><code>calculateChunkGroupTime</code></a> (long dvrTime)
String	<a href="#"><code>determineChunkGroupIdentifier</code></a> (long dvrTime)
int	<a href="#"><code>getChunkGroupDuration</code></a> ()
void	<a href="#"><code>setChunkGroupDuration</code></a> (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	<a href="#">addToCache</a> ( <a href="#">DvrManifestChunkEntry</a> entry, DvrChunk chunk) Provide in-memory cache the opportunity to add the DVR chunk to its cache.
void	<a href="#">init</a> ( <a href="#">IDvrStreamStore</a> store) Called to initialize the in-memory cache.
DvrChunk	<a href="#">retrieveRawChunk</a> ( <a href="#">DvrManifestChunkEntry</a> 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	<a href="#">determineUtcTime</a> (long currentUtc, long aChunkStartUtc, long aFirstPacketUtc, <a href="#">AMFPacket</a> aFirstPacket, long vChunkStartUtc, long vFirstPacketUtc, <a href="#">AMFPacket</a> vFirstPacket)
void	<a href="#">init</a> ( <a href="#">IDvrStreamStore</a> 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	<a href="#">IDvrConstants.DvrTimeScale</a> IDvrConstants.DvrTimeScale
-------	--

### Field Summary

public static final	<a href="#">ARCHIVE_STRATEGY_APPEND</a> Value: <b>append</b>
public static final	<a href="#">ARCHIVE_STRATEGY_DELETE</a> Value: <b>delete</b>
public static final	<a href="#">ARCHIVE_STRATEGY_VERSION</a> Value: <b>version</b>
public static final	<a href="#">DEFAULT_CHUNK_MEMORY_CACHESIZE</a> Default value for DVR Property "chunkMemoryCacheSize". Value: <b>10</b>
public static final	<a href="#">DEFAULT_CUPERTINO_PLAYLIST_GZIP_THRESHOLD</a> Default value for property "dvrCupertinoPlaylistGzipThreshold" #see <a href="#">PROPERTY_CUPERTINO_PLAYLIST_GZIP_THRESHOLD</a> Value: <b>4000</b>
public static final	<a href="#">DEFAULT_PROPERTY_ALLOWABLE_AV_PACKET_DELTA</a> Default value for DVR Property "dvrAllowableAVPacketDelta". Value: <b>2000</b>
public static final	<a href="#">DEFAULT_PROPERTY_APPEND_DISCONTINUITY_DELTA</a> Default value for DVR Property "dvrAppendDiscontinuityDelta". Value: <b>0</b>
public static final	<a href="#">DEFAULT_PROPERTY_AUDIO_ONLY_CHUNK_TARGET_DURATION</a> Default value for DVR Property "dvrAudioOnlyChunkTargetDuration". Value: <b>2000</b>
public static final	<a href="#">DEFAULT_PROPERTY_BREAK_ON_PTS</a> Default value for DVR Property "dvrChunkBreakOnPTS". Value: <b>true</b>
public static final	<a href="#">DEFAULT_PROPERTY_CHUNK_CACHE_CLASS</a> Default value for DVR Property "dvrChunkMemoryCacheClass". Value: <b>com.wowza.wms.dvr.impl.DvrDefaultChunkMemoryCache</b>

public static final	<a href="#">DEFAULT_PROPERTY_CHUNK_DURATION_MINIMUM</a> Default value for DVR Property "dvrChunkDurationMinimum". Value: <b>1500</b>
public static final	<a href="#">DEFAULT_PROPERTY_CHUNK_GROUPING_SECONDS</a> Default value for DVR Property "dvrChunkGroupingSeconds". Value: <b>600</b>
public static final	<a href="#">DEFAULT_PROPERTY_CHUNK_READER_CLASS</a> Default value for DVR Property "dvrChunkReaderClass". Value: <b>com.wowza.wms.dvr.impl.io.DvrFileChunkReader</b>
public static final	<a href="#">DEFAULT_PROPERTY_CHUNK_WRITER_CLASS</a> Default value for DVR Property "dvrChunkWriterClass". Value: <b>com.wowza.wms.dvr.impl.io.DvrFileChunkWriter</b>
public static final	<a href="#">DEFAULT_PROPERTY_DEBUG_MAX_INVALID_CHUNKS_LOGGED</a> Default value for DVR Property "dvrMaxInvalidChunksLogged". Value: <b>10</b>
public static final	<a href="#">DEFAULT_PROPERTY_DEBUG_MAX_RAW_PACKETS</a> Default value for DVR Property "dvrDebugMaximumRawPackets". Value: <b>200</b>
public static final	<a href="#">DEFAULT_PROPERTY_DEBUG_MAX_VALID_CHUNKS_LOGGED</a> Default value for DVR Property "dvrMaxValidChunksLogged". Value: <b>10</b>
public static final	<a href="#">DEFAULT_PROPERTY_DEBUG_RAW_PACKETS</a> Default value for DVR Property "dvrDebugRawPackets". Value: <b>false</b>
public static final	<a href="#">DEFAULT_PROPERTY_DVR_MAX_CHUNK_LOG</a> Default value for DVR Property "dvrMaxChunkLogCount". Value: <b>10</b>
public static final	<a href="#">DEFAULT_PROPERTY_FILE_SYSTEM_CLASS</a> Default value for DVR Property "dvrFileSystemClass". Value: <b>com.wowza.wms.dvr.impl.io.DvrDefaultFileSystem</b>
public static final	<a href="#">DEFAULT_PROPERTY_MANIFEST_PERSISTER_CLASS</a> Default value for DVR Property "dvrManifestPersisterClass". Value: <b>com.wowza.wms.dvr.impl.DvrManifestPersister</b>
public static final	<a href="#">DEFAULT_PROPERTY_MAX_RECALC_DURATION_LOG</a> Default value for DVR Property "dvrMaxRecountDurationLogCount". Value: <b>30</b>
public static final	<a href="#">DEFAULT_PROPERTY_MAX_RECORDING_LENGTH</a> Default value for DVR Property "dvrMaximumRecordingLength". Value: <b>108000</b>
public static final	<a href="#">DEFAULT_PROPERTY_MBR_MINIMUM_PACKETTIME_GAP_SIZE</a> Default value for DVR Property "dvrMbrMinimumPacketTimeGapSize". Value: <b>100</b>

public static final	<a href="#">DEFAULT_PROPERTY_MBR_MINIMUM_UTCTIME_GAP_SIZE</a> Default value for DVR Property "dvrMbrMinimumPacketTimeGapSize". Value: <b>750</b>
public static final	<a href="#">DEFAULT_PROPERTY_MEDIACACHE_READER_CLASS</a> Default value for DVR Property "dvrMediaCacheReaderClass". Value: <b>com.wowza.wms.plugin.mediacache.impl.MediaCacheRandomAccessReader</b>
public static final	<a href="#">DEFAULT_PROPERTY_PACKET_DELTA_TO_NOTIFY</a> Value: <b>200</b>
public static final	<a href="#">DEFAULT_PROPERTY_PACKET_DELTA_TO_RESET_TIME</a> Default value for DVR Property "dvrResetTimePacketDelta". Value: <b>200</b>
public static final	<a href="#">DEFAULT_PROPERTY_PACKET_DURATION_MAXIMUM</a> Default value for DVR Property "dvrChunkDurationMinimum". Value: <b>5000</b>
public static final	<a href="#">DEFAULT_PROPERTY_PACKET_SORT_TIME</a> Default value for DVR Property "dvrPacketSortTime". Value: <b>0</b>
public static final	<a href="#">DEFAULT_PROPERTY_RECORDINGS_LOADER_CLASS</a> Default value for DVR Property "dvrRecordingsLoaderClass". Value: <b>com.wowza.wms.dvr.DvrRecordingsLoader</b>
public static final	<a href="#">DEFAULT_PROPERTY_REPEATER_HEARTBEAT_DURATION</a> Value: <b>4000</b>
public static final	<a href="#">DEFAULT_PROPERTY_SANJOSE_ABST_DURATION_TOLERANCE</a> Default value for property "dvrSanJosePlaylistAbstDurationTolerance" #see <a href="#">PROPERTY_SANJOSE_ABST_DURATION_TOLERANCE</a> Value: <b>50</b>
public static final	<a href="#">DEFAULT_PROPERTY_SANJOSE_ABST_TIMESCALE</a> Default value for property "dvrSanJosePlaylistAbstTimescale" #see <a href="#">PROPERTY_SANJOSE_ABST_TIMESCALE</a> Value: <b>1000</b>
public static final	<a href="#">DEFAULT_PROPERTY_SMOOTH_MANIFEST_LIVE_CAN_PAUSE</a> Value: <b>true</b>
public static final	<a href="#">DEFAULT_PROPERTY_SMOOTH_MANIFEST_LIVE_CAN_SEEK</a> Value: <b>true</b>
public static final	<a href="#">DEFAULT_PROPERTY_SMOOTH_MANIFEST_MAJOR_VERSION</a> Value: <b>2</b>
public static final	<a href="#">DEFAULT_PROPERTY_SMOOTH_MANIFEST_MINOR_VERSION</a> Value: <b>1</b>



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

public static final	<a href="#"><u>PROPERTY_APPEND_DISCONTINUITY_DELTA</u></a> DVR Property "dvrAppendDiscontinuityDelta": for specifying how many milliseconds of empty space are put between individual recordings that when in append mode. Value: <b>dvrAppendDiscontinuityDelta</b>
public static final	<a href="#"><u>PROPERTY_ARCHIVE_STRATEGY</u></a> DVR Property "dvrArchiveStrategy": for specifying the DVR archive strategy. Value: <b>dvrArchiveStrategy</b>
public static final	<a href="#"><u>PROPERTY_AUDIO_ONLY_CHUNK_TARGET_DURATION</u></a> DVR Property "dvrAudioOnlyChunkTargetDuration": for target duration when recording audio-only. Value: <b>dvrAudioOnlyChunkTargetDuration</b>
public static final	<a href="#"><u>PROPERTY_BREAK_ON_PTS</u></a> DVR Property "dvrChunkBreakOnPTS": for specifying that DVR should be broken on PTS. Value: <b>dvrChunkBreakOnPTS</b>
public static final	<a href="#"><u>PROPERTY_CHUNK_CACHE_CLASS</u></a> DVR Property "dvrChunkMemoryCacheClass": for controlling the class responsible for caching DVR chunks in memory. Value: <b>dvrChunkMemoryCacheClass</b>
public static final	<a href="#"><u>PROPERTY_CHUNK_DURATION_MINIMUM</u></a> DVR Property "dvrChunkDurationMinimum": for minimum chunk duration, in milliseconds. Value: <b>dvrChunkDurationMinimum</b>
public static final	<a href="#"><u>PROPERTY_CHUNK_GROUPING_SECONDS</u></a> DVR Property "dvrChunkGroupingSeconds": for determining how many seconds of DVR are stored in each stores sub-folder. Value: <b>dvrChunkGroupingSeconds</b>
public static final	<a href="#"><u>PROPERTY_CHUNK_MEMORY_CACHESIZE</u></a> DVR Property "chunkMemoryCacheSize": used by DvrDefaultChunkMemoryCache to set number of chunks stored in DVR in-memory cache. Value: <b>chunkMemoryCacheSize</b>
public static final	<a href="#"><u>PROPERTY_CHUNK_READER_CLASS</u></a> DVR Property "dvrChunkReaderClass": for controlling the class responsible for reading DVR Chunks. Value: <b>dvrChunkReaderClass</b>
public static final	<a href="#"><u>PROPERTY_CHUNK_WRITER_CLASS</u></a> DVR Property "dvrChunkWriterClass": for controlling the class responsible for writing DVR Chunks. Value: <b>dvrChunkWriterClass</b>
public static final	<a href="#"><u>PROPERTY_CUPERTINO_ON_CHUNK_START_RESET_COUNTER</u></a> DVR Property "dvrCupertinoOnChunkStartResetCounter": when a new chunk starts, reset internal tsPacketizer counters Value: <b>dvrCupertinoOnChunkStartResetCounter</b>
public static final	<a href="#"><u>PROPERTY_CUPERTINO_PLAYLIST_FORCE_LIVE</u></a> DVR Property "dvrCupertinoPlaylistForceLive": used to override playlist request delegate logic that determines if playlist is live. Value: <b>dvrCupertinoPlaylistForceLive</b>

public static final	<a href="#"><u>PROPERTY_CUPERTINO_PLAYLIST_FORCE_NONLIVE</u></a> DVR Property "dvrCupertinoPlaylistForceLive": used to override playlist request delegate logic that determines if playlist is live versus non-live. Value: <b>dvrCupertinoPlaylistForceNonLive</b>
public static final	<a href="#"><u>PROPERTY_CUPERTINO_PLAYLIST_GZIP_THRESHOLD</u></a> DVR Property "dvrCupertinoPlaylistGzipThreshold": when playlist is larger than this number of bytes, and gzip is enabled and accepted, the playlist will be compressed Value: <b>dvrCupertinoPlaylistGzipThreshold</b>
public static final	<a href="#"><u>PROPERTY_CUPERTINO_PLAYLIST_USE_GZIP</u></a> DVR Property "dvrCupertinoPlaylistUseGzip": used to force Cupertino playlist to use gzip if it is accepted Value: <b>dvrCupertinoPlaylistUseGzip</b>
public static final	<a href="#"><u>PROPERTY_DEBUG_CHUNK_RETRIEVALS</u></a> DVR Property "dvrDebugChunkRetrievals": for logging each chunk retrieval Value: <b>dvrDebugChunkRetrievals</b>
public static final	<a href="#"><u>PROPERTY_DEBUG_CUPERTINO_PLAYER_ADAPTER</u></a> HTTP Streamer Property "dvrDebugCupertinoPlayerAdapter": for turning on DVR Player Cupertino Adapter debug logging. Value: <b>dvrDebugCupertinoPlayerAdapter</b>
public static final	<a href="#"><u>PROPERTY_DEBUG_FAILED_CHUNK_RETRIEVALS</u></a> DVR Property "dvrDebugFailedChunkRetrievals": for logging info about each failed chunk retrieval Value: <b>dvrDebugFailedChunkRetrievals</b>
public static final	<a href="#"><u>PROPERTY_DEBUG_LOG_INVALID_CHUNK_DETAILS</u></a> DVR Property "dvrLogInvalidChunkDetails": to control detailed logging information of DVR invalid chunks Value: <b>dvrLogInvalidChunkDetails</b>
public static final	<a href="#"><u>PROPERTY_DEBUG_LOG_INVALID_CHUNK_MATCHER</u></a> DVR Property "dvrLogInvalidChunkMatcher": for matching stream names that will log chunk packets Value: <b>dvrLogInvalidChunkMatcher</b>
public static final	<a href="#"><u>PROPERTY_DEBUG_LOG_VALID_CHUNK_DETAILS</u></a> DVR Property "dvrLogValidChunkDetails": to control detailed logging information of DVR valid chunks Value: <b>dvrLogValidChunkDetails</b>
public static final	<a href="#"><u>PROPERTY_DEBUG_LOG_VALID_CHUNK_MATCHER</u></a> DVR Property "dvrLogValidChunkMatcher": for matching stream names that will log chunk packets Value: <b>dvrLogValidChunkMatcher</b>
public static final	<a href="#"><u>PROPERTY_DEBUG_MAX_INVALID_CHUNKS_LOGGED</u></a> DVR Property "dvrMaxInvalidChunksLogged": for controlling maximum number of invalid DVR chunks logged. Value: <b>dvrMaxInvalidChunksLogged</b>
public static final	<a href="#"><u>PROPERTY_DEBUG_MAX_VALID_CHUNKS_LOGGED</u></a> DVR Property "dvrMaxValidChunksLogged": for controlling maximum number of valid DVR chunks logged. Value: <b>dvrMaxValidChunksLogged</b>

public static final	<a href="#"><u>PROPERTY_DEBUG_MAXIMUM_RAW_PACKETS</u></a> DVR Property "dvrDebugMaximumRawPackets": for setting maximum number of logged raw packets. Value: <b>dvrDebugMaximumRawPackets</b>
public static final	<a href="#"><u>PROPERTY_DEBUG_MBR_ALIGNMENT</u></a> DVR Property "dvrDebugMbrAlignment": for turning on logging of mbr alignment Value: <b>dvrDebugMbrAlignment</b>
public static final	<a href="#"><u>PROPERTY_DEBUG_MBR_ALIGNMENT_RESOLUTION</u></a> DVR Property "dvrDebugMbrAlignmentResolution": for turning on logging of mbr alignment resolution (requested to actual) Value: <b>dvrDebugMbrAlignmentResolution</b>
public static final	<a href="#"><u>PROPERTY_DEBUG_MBR_PLAYER_ADAPTER</u></a> HTTP Streamer Property "dvrDebugMbrPlayerAdapter": for turning on DVR MBR Player Adapter debug logging. Value: <b>dvrDebugMbrPlayerAdapter</b>
public static final	<a href="#"><u>PROPERTY_DEBUG_METHODS</u></a> DVR Property "dvrDebugManagerLogMethods": for turning on DVR Manager debug logging. Value: <b>dvrDebugManagerLogMethods</b>
public static final	<a href="#"><u>PROPERTY_DEBUG_PLAYER_ADAPTER</u></a> HTTP Streamer Property "dvrDebugPlayerAdapter": for turning on DVR Player Adapter debug logging for all streamer types. Value: <b>dvrDebugPlayerAdapter</b>
public static final	<a href="#"><u>PROPERTY_DEBUG_PLAYLIST_REQUEST</u></a> DVR Property "dvrDebugPlaylistRequest": for turning on logging of DVR playlist requests. Value: <b>dvrDebugPlaylistRequest</b>
public static final	<a href="#"><u>PROPERTY_DEBUG_RAW_PACKETS</u></a> DVR Property "dvrDebugRawPackets": for turning on logging of incoming raw packets. Value: <b>dvrDebugRawPackets</b>
public static final	<a href="#"><u>PROPERTY_DEBUG_RAW_PACKETS_MATCHER</u></a> DVR Property "dvrDebugRawPacketsMatcher": for matching stream names that will dump raw packet. Value: <b>dvrDebugRawPacketsMatcher</b>
public static final	<a href="#"><u>PROPERTY_DEBUG_REPEATER</u></a> DVR Property "dvrDebugRepeater": for turning on logging of DVR repeater Value: <b>dvrDebugRepeater</b>
public static final	<a href="#"><u>PROPERTY_DEBUG_SANJOSE_PLAYER_ADAPTER</u></a> HTTP Streamer Property "dvrDebugSanJosePlayerAdapter": for turning on DVR Player San Jose Adapter debug logging. Value: <b>dvrDebugSanJosePlayerAdapter</b>
public static final	<a href="#"><u>PROPERTY_DEBUG_SMOOTH_PLAYER_ADAPTER</u></a> HTTP Streamer Property "dvrDebugSmoothPlayerAdapter": for turning on DVR Player Smooth Adapter debug logging. Value: <b>dvrDebugSmoothPlayerAdapter</b>

public static final	<a href="#"><u>PROPERTY_DEBUG_STATE_CHANGE</u></a> DVR Property "dvrDebugStateChange": for logging state changes of DVR store. Value: <b>dvrDebugStateChange</b>
public static final	<a href="#"><u>PROPERTY_DEBUG_TOSSED_HOLDERS</u></a> DVR Property "dvrDebugTossedHolders": for turning on logging of packets that are being tossed. Value: <b>dvrDebugTossedHolders</b>
public static final	<a href="#"><u>PROPERTY_ENCRYPTION_INFO_DELEGATE</u></a> DVR Property "dvrPlaylistEncryptionInfoDelegate": used to over-ride the encryption info on the playback side. Value: <b>dvrPlaylistEncryptionInfoDelegate</b>
public static final	<a href="#"><u>PROPERTY_FILE_SYSTEM_CLASS</u></a> DVR Property "dvrFileSystemClass": for controlling the class responsible for managing the DVR File System. Value: <b>dvrFileSystemClass</b>
public static final	<a href="#"><u>PROPERTY_MANIFEST_PERSISTER_CLASS</u></a> DVR Property "dvrManifestPersisterClass": for controlling the class responsible for persisting the manifest files. Value: <b>dvrManifestPersisterClass</b>
public static final	<a href="#"><u>PROPERTY_MAX_CHUNK_LOG</u></a> DVR Property "dvrMaxChunkLogCount": for maximum number of DVR chunks to log. Value: <b>dvrMaxChunkLogCount</b>
public static final	<a href="#"><u>PROPERTY_MAX_RECALC_DURATION_LOG</u></a> DVR Property "dvrMaxRecountDurationLogCount": for maximum number of messages to log regarding recalculating DVR chunk duration Value: <b>dvrMaxRecountDurationLogCount</b>
public static final	<a href="#"><u>PROPERTY_MAX_RECORDING_LENGTH</u></a> DVR Property "dvrMaximumRecordingLength": The maximum recording length in seconds. Value: <b>dvrMaximumRecordingLength</b>
public static final	<a href="#"><u>PROPERTY_MBR_MINIMUM_PACKETTIME_GAP_SIZE</u></a> DVR Property "dvrMbrMinimumPacketTimeGapSize": when doing mbr alignment gaps smaller than this are ignored. Value: <b>dvrMbrMinimumPacketTimeGapSize</b>
public static final	<a href="#"><u>PROPERTY_MBR_MINIMUM_UTCTIME_GAP_SIZE</u></a> DVR Property "dvrMbrMinimumUtcTimeGapSize": when doing mbr alignment gaps smaller than this are ignored. Value: <b>dvrMbrMinimumUtcTimeGapSize</b>
public static final	<a href="#"><u>PROPERTY_MBR_USE_UTC_FOR_ALIGNMENT</u></a> DVR Property "dvrMbrUseUtcForAlignment": when doing mbr alignment use utc time for alignment Value: <b>dvrMbrUseUtcForAlignment</b>
public static final	<a href="#"><u>PROPERTY_MEDIACACHE_ENABLED</u></a> DVR Property "dvrMediaCacheEnabled" Value: <b>dvrMediaCacheEnabled</b>

public static final	<a href="#">PROPERTY_MEDIACACHE_READER_CLASS</a> DVR Property "dvrMediaCacheReaderClass" Value: <b>dvrMediaCacheReaderClass</b>
public static final	<a href="#">PROPERTY_PACKET_DELTA_TO_NOTIFY</a> Value: <b>dvrPacketDeltaToNotify</b>
public static final	<a href="#">PROPERTY_PACKET_DELTA_TO_RESET_TIME</a> 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: <b>dvrResetTimePacketDelta</b>
public static final	<a href="#">PROPERTY_PACKET_DURATION_MAXIMUM</a> DVR Property "dvrPacketDurationMaximum": for maximum packet duration, in milliseconds. Value: <b>dvrPacketDurationMaximum</b>
public static final	<a href="#">PROPERTY_PACKET_SORT_TIME</a> DVR Property "dvrPacketSortTime": for sorting incoming packets. Value: <b>dvrPacketSortTime</b>
public static final	<a href="#">PROPERTY_PLAYLIST_REQUEST_DELEGATE</a> DVR Property "dvrPlaylistRequestDelegate": used to over-ride the class that generates a playlist request. Value: <b>dvrPlaylistRequestDelegate</b>
public static final	<a href="#">PROPERTY_RECORD_AUDIO</a> DVR Property "recordAudio": for determining if audio should be recorded. Value: <b>recordAudio</b>
public static final	<a href="#">PROPERTY_RECORD_DATA</a> DVR Property "recordData": for determining if data should be recorded. Value: <b>recordData</b>
public static final	<a href="#">PROPERTY_RECORD_VIDEO</a> DVR Property "recordVideo": for determining if video should be recorded. Value: <b>recordVideo</b>
public static final	<a href="#">PROPERTY_RECORDINGS_LOADER_CLASS</a> DVR Property "dvrRecordingsLoaderClass": for controlling the class responsible for loading DVR recordings. Value: <b>dvrRecordingsLoaderClass</b>
public static final	<a href="#">PROPERTY_REPEATER_HEARTBEAT_DURATION</a> DVR Property "dvrRepeaterHeartbeatDuration": for time in ms that origin pings edges Value: <b>dvrRepeaterHeartbeatDuration</b>
public static final	<a href="#">PROPERTY_REPEATER_SHARED_SECRET</a> DVR Property "dvrEncryptionSharedSecret": for encryption shared secret between origins and edges. Value: <b>dvrEncryptionSharedSecret</b>
public static final	<a href="#">PROPERTY_SANJOSE_ABST_DURATION_TOLERANCE</a> DVR Property "dvrSanJosePlaylistAbstDurationEqualityTolerance": used to over-ride the tolerance when determining equal chunks lengths. Value: <b>dvrSanJosePlaylistAbstDurationTolerance</b>

public static final	<a href="#"><u>PROPERTY_SANJOSE_ABST_TIMESCALE</u></a> DVR Property "dvrSanJosePlaylistAbstTimescale": used to over-ride the time-scale for abst files. Value: <b>dvrSanJosePlaylistAbstTimescale</b>
public static final	<a href="#"><u>PROPERTY_SANJOSE_PLAYLIST_DELIVERYTYPE</u></a> DVR Property "dvrSanJosePlaylistDeliveryType": used to over-ride the mime type for DVR San Jose F4m playlists. Value: <b>dvrSanJosePlaylistDeliveryType</b>
public static final	<a href="#"><u>PROPERTY_SANJOSE_PLAYLIST_LIVE_STREAMTYPE</u></a> DVR Property "dvrSanJosePlaylistLiveStreamType": used to over-ride the stream type for live DVR in San Jose f4m playlists. Value: <b>dvrSanJosePlaylistLiveStreamType</b>
public static final	<a href="#"><u>PROPERTY_SANJOSE_PLAYLIST_MIMETYPE</u></a> DVR Property "dvrSanJosePlaylistMimeType": used to over-ride the mime type for DVR San Jose F4m playlists. Value: <b>dvrSanJosePlaylistMimeType</b>
public static final	<a href="#"><u>PROPERTY_SANJOSE_PLAYLIST_RECORDED_STREAMTYPE</u></a> DVR Property "dvrSanJosePlaylistRecordedStreamType": used to over-ride the stream type for non-live (recorded) DVR in San Jose f4m playlists. Value: <b>dvrSanJosePlaylistRecordedStreamType</b>
public static final	<a href="#"><u>PROPERTY_SANJOSE_PLAYLIST_VERSION</u></a> DVR Property "dvrSanJosePlaylistVersion": used to over-ride the version of the San Jose f4m playlist. Value: <b>dvrSanJosePlaylistVersion</b>
public static final	<a href="#"><u>PROPERTY_SMOOTH_MANIFEST_H264_CODEC</u></a> DVR Property "dvrSmoothManifestH264Codec": used to force Smooth Manifest to use this as its FourCC H264 codec info. Value: <b>dvrSmoothManifestH264Codec</b>
public static final	<a href="#"><u>PROPERTY_SMOOTH_MANIFEST_LIVE_CAN_PAUSE</u></a> DVR Property "dvrSmoothManifestLiveCanPause": used to determine is CanPause is enabled in smooth Manifest. Value: <b>dvrSmoothManifestLiveCanPause</b>
public static final	<a href="#"><u>PROPERTY_SMOOTH_MANIFEST_LIVE_CAN_SEEK</u></a> DVR Property "dvrSmoothManifestLiveCanSeek": used to determine is CanSeek is enabled in smooth Manifest. Value: <b>dvrSmoothManifestLiveCanSeek</b>
public static final	<a href="#"><u>PROPERTY_SMOOTH_MANIFEST_MAJOR_VERSION</u></a> DVR Property "dvrSmoothManifestMajorVersion": used to determine Smooth manifest major version Value: <b>dvrSmoothManifestMajorVersion</b>
public static final	<a href="#"><u>PROPERTY_SMOOTH_MANIFEST_MINOR_VERSION</u></a> DVR Property "dvrSmoothManifestMinorVersion": used to determine Smooth manifest major version Value: <b>dvrSmoothManifestMinorVersion</b>
public static final	<a href="#"><u>PROPERTY_SMOOTH_MANIFEST_RECORDED_CAN_PAUSE</u></a> DVR Property "dvrSmoothManifestRecordedCanPause": used to determine is CanPause is enabled in smooth Manifest. Value: <b>dvrSmoothManifestRecordedCanPause</b>



public static final	<a href="#"><u>PROPERTY_SMOOTH_MANIFEST_RECORDED_CAN_SEEK</u></a> DVR Property "dvrSmoothManifestRecordedCanSeek": used to determine if CanSeek is enabled in smooth Manifest. Value: <b>dvrSmoothManifestRecordedCanSeek</b>
public static final	<a href="#"><u>PROPERTY_SMOOTH_MANIFEST_RECORDED_SPECIFY_DURATION</u></a> DVR Property "dvrSmoothManifestRecordedSpecifyDuration": used to determine if duration is specified. Value: <b>dvrSmoothManifestRecordedSpecifyDuration</b>
public static final	<a href="#"><u>PROPERTY_SMOOTH_MANIFEST_VERBOSE_DURATION</u></a> DVR Property "dvrSmoothManifestVerboseDuration": used to force Smooth Manifest to include durations for each record Value: <b>dvrSmoothManifestVerboseDuration</b>
public static final	<a href="#"><u>PROPERTY_START_RECORDING_ON_STARTUP</u></a> DVR Property "startRecordingOnStartup": for determining if DVR recorder should start recording immediately. Value: <b>startRecordingOnStartup</b>
public static final	<a href="#"><u>PROPERTY_STORAGE_DIRECTORY</u></a> DVR Property "dvrStorageDirectory": for overriding the DVR storage directory location Typically this is defined application-wide in Application.xml under Application/DVR/StorageDirectory. Value: <b>dvrStorageDirectory</b>
public static final	<a href="#"><u>PROPERTY_TEXT_READER_CLASS</u></a> DVR Property "dvrTextReaderClass": for controlling the class responsible for reading DVR text files. Value: <b>dvrTextReaderClass</b>
public static final	<a href="#"><u>PROPERTY_TEXT_WRITER_CLASS</u></a> DVR Property "dvrTextWriterClass": for controlling the class responsible for writing DVR text files. Value: <b>dvrTextWriterClass</b>
public static final	<a href="#"><u>PROPERTY_WINDOW_DURATION</u></a> DVR Property "dvrWindowDuration": for specifying the DVR window duration, in seconds. Value: <b>dvrWindowDuration</b>
public static final	<a href="#"><u>SANJOSE_F4M_STREAMINGTYPE_STREAMING</u></a> Constant for San Jose streaming type "streaming" Value: <b>streaming</b>
public static final	<a href="#"><u>SANJOSE_F4M_STREAMTYPE_DVR</u></a> Constant for San Jose stream type "dvr". Value: <b>dvr</b>
public static final	<a href="#"><u>SANJOSE_F4M_STREAMTYPE_LIVE</u></a> Constant for San Jose stream type "live". Value: <b>live</b>
public static final	<a href="#"><u>SANJOSE_F4M_STREAMTYPE_LIVEORRECORDED</u></a> Constant for San Jose stream type "liveOrRecorded". Value: <b>liveOrRecorded</b>



public static final	<a href="#">SANJOSE_F4M_STREAMTYPE_RECORDED</a> Constant for San Jose stream type "recorded". Value: <b>recorded</b>
public static final	<a href="#">SANJOSE_F4M_VERSION_1_0</a> Constant for San Jose f4m version "1.0". Value: <b>1.0</b>
public static final	<a href="#">SANJOSE_F4M_VERSION_2_0</a> Constant for San Jose f4m version "2.0". Value: <b>2.0</b>

#### 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**

(continued from last page)

---

## DVR\_DEFAULT\_RECORDER\_ID

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

The default DVR recorder ID: "dvrrecorder".

See DVR.xml and Application.xml.  
Constant value: **dvrrecorder**

---

## DVR\_DEFAULT\_FILESTORE

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

The default DVR store ID: "dvrfilestorage".

See DVR.xml and Application.xml.  
Constant value: **dvrfilestorage**

---

## MEDIACACHE\_PREFIX

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

The MediaCache prefix for dvr repeater

See MediaCache.xml  
Constant value: **dvrorigin**

---

## PROPERTY\_CHUNK\_GROUPING\_SECONDS

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

DVR Property "dvrChunkGroupingSeconds": for determining how many seconds of DVR are stored in each stores sub-folder.

The directory naming convention is HHHH\_MM\_SS with H = hours, M = minutes, S = seconds. Using the defaults, first directory would be named 0000\_00\_00 and the second directory would be named 0000\_10\_00, etc.

Valid values are integers greater than 60. Add the property to Application/DVR/Properties section of Application.xml.  
Constant value: **dvrChunkGroupingSeconds**

See Also:

[DEFAULT\\_PROPERTY\\_CHUNK\\_GROUPING\\_SECONDS](#)

---

## DEFAULT\_PROPERTY\_CHUNK\_GROUPING\_SECONDS

```
public static final int DEFAULT_PROPERTY_CHUNK_GROUPING_SECONDS
```

Default value for DVR Property "dvrChunkGroupingSeconds".

Default value is 600 seconds (10 minutes).  
Constant value: **600**

See Also:

[PROPERTY\\_CHUNK\\_GROUPING\\_SECONDS](#)

---

## PROPERTY\_APPEND\_DISCONTINUITY\_DELTA

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

(continued from last page)

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
```

(continued from last page)

Default value for DVR Property "dvrStorageDirectory".

Default value is "\${com.wowza.wms.context.VHostConfigHome}/dvr".

Constant value: **`${com.wowza.wms.context.VHostConfigHome}/dvr`**

See Also:

[PROPERTY\\_STORAGE\\_DIRECTORY](#)

---

## PROPERTY\_ARCHIVE\_STRATEGY

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

DVR Property "dvrArchiveStrategy": for specifying the DVR archive strategy.

Typically this is defined application-wide in Application.xml under Application/DVR/ArchiveStrategy However, to override this on a per stream basis, the property may be set on the IDvrStreamManager after creation and before initialization. Valid values are:

Constant value: **`dvrArchiveStrategy`**

See Also:

[ARCHIVE\\_STRATEGY\\_APPEND](#)

[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](#)

---

(continued from last page)

---

## DEFAULT\_PROPERTY\_CHUNK\_READER\_CLASS

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

Default value for DVR Property "dvrChunkReaderClass".

Default value is "com.wowza.wms.dvr.impl.io.DvrFileChunkReader".

Constant value: **com.wowza.wms.dvr.impl.io.DvrFileChunkReader**

See Also:

[PROPERTY\\_CHUNK\\_READER\\_CLASS](#)

---

## PROPERTY\_CHUNK\_WRITER\_CLASS

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

DVR Property "dvrChunkWriterClass": for controlling the class responsible for writing DVR Chunks.

The class must implement IDvrChunkWriter

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

Constant value: **dvrChunkWriterClass**

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".

(continued from last page)

Constant value: **com.wowza.wms.dvr.impl.io.DvrDefaultFileSystem**

See Also:

[PROPERTY\\_FILE\\_SYSTEM\\_CLASS](#)

---

## PROPERTY\_MANIFEST\_PERSISTER\_CLASS

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

DVR Property "dvrManifestPersisterClass": for controlling the class responsible for persisting the manifest files.

The class must implement `com.wowza.wms.dvr.io.IDvrManifestPersister`

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

Constant value: **dvrManifestPersisterClass**

See Also:

`com.wowza.wms.dvr.io.IDvrManifestPersister`

[DEFAULT\\_PROPERTY\\_MANIFEST\\_PERSISTER\\_CLASS](#)

---

## 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](#)

(continued from last page)

---

## PROPERTY\_RECORDINGS\_LOADER\_CLASS

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

DVR Property "dvrRecordingsLoaderClass": for controlling the class responsible for loading DVR recordings.

The class must implement [IDvrRecordingsLoader](#)

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

Constant value: **dvrRecordingsLoaderClass**

See Also:

[IDvrRecordingsLoader](#)

[DEFAULT\\_PROPERTY\\_RECORDINGS\\_LOADER\\_CLASS](#)

---

## DEFAULT\_PROPERTY\_RECORDINGS\_LOADER\_CLASS

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

Default value for DVR Property "dvrRecordingsLoaderClass".

Default value is "com.wowza.wms.dvr.DvrRecordingsLoader".

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
```

---

(continued from last page)

DVR Property "dvrTextReaderClass": for controlling the class responsible for reading DVR text files.

The class must implement [IDvrTextReader](#)

Add the property to Application/DVR/Properties section of Application.xml.  
Constant value: **dvrTextReaderClass**

See Also:

[IDvrTextReader](#)

[DEFAULT\\_PROPERTY\\_TEXT\\_READER\\_CLASS](#)

---

## DEFAULT\_PROPERTY\_TEXT\_READER\_CLASS

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

Default value for DVR Property "dvrTextReaderClass".

Default value is "com.wowza.wms.dvr.impl.io.DvrTextFileReader".  
Constant value: **com.wowza.wms.dvr.impl.io.DvrTextFileReader**

See Also:

[PROPERTY\\_TEXT\\_READER\\_CLASS](#)

---

## 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



(continued from last page)

Constant value: **dvrResetTimePacketDelta**

See Also:

[DEFAULT\\_PROPERTY\\_PACKET\\_DELTA\\_TO\\_RESET\\_TIME](#)

---

## DEFAULT\_PROPERTY\_PACKET\_DELTA\_TO\_RESET\_TIME

```
public static final int DEFAULT_PROPERTY_PACKET_DELTA_TO_RESET_TIME
```

Default value for DVR Property "dvrResetTimePacketDelta".

Default value is 200 ms.

Constant value: **200**

See Also:

[PROPERTY\\_PACKET\\_DELTA\\_TO\\_RESET\\_TIME](#)

---

## PROPERTY\_PACKET\_DELTA\_TO\_NOTIFY

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

Constant value: **dvrPacketDeltaToNotify**

---

## DEFAULT\_PROPERTY\_PACKET\_DELTA\_TO\_NOTIFY

```
public static final int DEFAULT_PROPERTY_PACKET_DELTA_TO_NOTIFY
```

Constant value: **200**

---

## PROPERTY\_RECORD\_DATA

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

DVR Property "recordData": for determining if data should be recorded.

Set property to: true to record data of incoming stream. Default is true. Set property to: false to ignore data packets during recording.

Constant value: **recordData**

---

## PROPERTY\_RECORD\_VIDEO

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

DVR Property "recordVideo": for determining if video should be recorded.

Set property to: true to record video of incoming stream. Default is true. Set property to: false to ignore video packets during recording. Either recordVideo or recordAudio property must be set to: true

Constant value: **recordVideo**

---

## PROPERTY\_RECORD\_AUDIO

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

DVR Property "recordAudio": for determining if audio should be recorded.

Set property to: true to record audio of incoming stream. Default is true. Set property to: false to ignore audio packets during recording. Either recordVideo or recordAudio property must be set to: true

Constant value: **recordAudio**

---

## PROPERTY\_START\_RECORDING\_ON\_STARTUP

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

DVR Property "startRecordingOnStartup": for determining if DVR recorder should start recording immediately.

Set property to: true (default) to start recording immediately when stream is detected. Set property to: false to init the DVR recorder but to not start recording immediately when stream starts.

Constant value: **startRecordingOnStartup**

---

## PROPERTY\_AUDIO\_ONLY\_CHUNK\_TARGET\_DURATION

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

DVR Property "dvrAudioOnlyChunkTargetDuration": for target duration when recording audio-only.

Applies to recording an audio-only stream and is the target chunk duration. The unit is in ms. The default value is 2000 ms. This setting is ignored if the stream contains video and audio, in which case the keyframe determines the chunk size.

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

Constant value: **dvrAudioOnlyChunkTargetDuration**

See Also:

[DEFAULT\\_PROPERTY\\_AUDIO\\_ONLY\\_CHUNK\\_TARGET\\_DURATION](#)

---

## DEFAULT\_PROPERTY\_AUDIO\_ONLY\_CHUNK\_TARGET\_DURATION

```
public static final int DEFAULT_PROPERTY_AUDIO_ONLY_CHUNK_TARGET_DURATION
```

Default value for DVR Property "dvrAudioOnlyChunkTargetDuration".

Default value is 2000 ms.

Constant value: **2000**

See Also:

[PROPERTY\\_AUDIO\\_ONLY\\_CHUNK\\_TARGET\\_DURATION](#)

---

## PROPERTY\_PACKET\_SORT\_TIME

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

DVR Property "dvrPacketSortTime": for sorting incoming packets. Specified in milliseconds.

Valid values are a integer greater than equal to 0. Zero means no sorting occurs. Add the property to Application/DVR/Properties section of Application.xml

Constant value: **dvrPacketSortTime**

See Also:

[DEFAULT\\_PROPERTY\\_PACKET\\_SORT\\_TIME](#)

---

## DEFAULT\_PROPERTY\_PACKET\_SORT\_TIME

```
public static final int DEFAULT_PROPERTY_PACKET_SORT_TIME
```

Default value for DVR Property "dvrPacketSortTime".

Default value is 0 ms (i.e. no sorting).

Constant value: **0**

See Also:

[PROPERTY\\_PACKET\\_SORT\\_TIME](#)

---

## PROPERTY\_BREAK\_ON\_PTS

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

DVR Property "dvrChunkBreakOnPTS": for specifying that DVR should be broken on PTS.

Valid values are "true" or "false" Add the property to Application/DVR/Properties section of Application.xml  
Constant value: **dvrChunkBreakOnPTS**

See Also:

[DEFAULT\\_PROPERTY\\_BREAK\\_ON\\_PTS](#)

---

## DEFAULT\_PROPERTY\_BREAK\_ON\_PTS

```
public static final boolean DEFAULT_PROPERTY_BREAK_ON_PTS
```

Default value for DVR Property "dvrChunkBreakOnPTS".

Default value is true.  
Constant value: **true**

See Also:

[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
```

---

---

(continued from last page)

DVR Property "dvrChunkDurationMinimum": for minimum chunk duration, in milliseconds.

Add the property to Application/DVR/Properties section of Application.xml  
Constant value: **dvrChunkDurationMinimum**

See Also:

[DEFAULT\\_PROPERTY\\_CHUNK\\_DURATION\\_MINIMUM](#)

---

## DEFAULT\_PROPERTY\_CHUNK\_DURATION\_MINIMUM

```
public static final int DEFAULT_PROPERTY_CHUNK_DURATION_MINIMUM
```

Default value for DVR Property "dvrChunkDurationMinimum".

Default value is 1500 ms.  
Constant value: **1500**

See Also:

[PROPERTY\\_CHUNK\\_DURATION\\_MINIMUM](#)

---

## PROPERTY\_CHUNK\_MEMORY\_CACHESIZE

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

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**

---

(continued from last page)

**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:**

(continued from last page)

[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 A value of zero means no maximum recording length is enforced. If used in conjunction with DVRWindow, should be slightly larger (1-2%) than DVRWindow.  
Constant value: **dvrMaximumRecordingLength**

See Also:

[DEFAULT\\_PROPERTY\\_MAX\\_RECORDING\\_LENGTH](#)

---

## DEFAULT\_PROPERTY\_MAX\_RECORDING\_LENGTH

```
public static final long DEFAULT_PROPERTY_MAX_RECORDING_LENGTH
```

Default value for DVR Property "dvrMaximumRecordingLength".

Default value is 108000 (30 hours)  
Constant value: **108000**

See Also:

[PROPERTY\\_MAX\\_RECORDING\\_LENGTH](#)

---

## PROPERTY\_MEDIACACHE\_ENABLED

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

DVR Property "dvrMediaCacheEnabled"

Add the property to Application/DVR/Properties section of Application.xml Default is false.  
Constant value: **dvrMediaCacheEnabled**

---

## PROPERTY\_MEDIACACHE\_READER\_CLASS

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

DVR Property "dvrMediaCacheReaderClass"

Add the property to Application/DVR/Properties section of Application.xml  
Constant value: **dvrMediaCacheReaderClass**

See Also:

[DEFAULT\\_PROPERTY\\_MEDIACACHE\\_READER\\_CLASS](#)

---

## DEFAULT\_PROPERTY\_MEDIACACHE\_READER\_CLASS

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

Default value for DVR Property "dvrMediaCacheReaderClass".  
Constant value: **com.wowza.wms.plugin.mediache.impl.MediaCacheRandomAccessReader**

See Also:

[PROPERTY\\_MEDIACACHE\\_READER\\_CLASS](#)

(continued from last page)

---

## 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
```

(continued from last page)

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
```

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
```

---

---

(continued from last page)

DVR Property "dvrMaxInvalidChunksLogged": for controlling maximum number of invalid DVR chunks logged.

The default is 10. Logging only occurs if property [PROPERTY\\_DEBUG\\_LOG\\_INVALID\\_CHUNK\\_DETAILS](#) is true. Add the property to Application/DVR/Properties section of Application.xml  
Constant value: **dvrMaxInvalidChunksLogged**

See Also:

[DEFAULT\\_PROPERTY\\_DEBUG\\_MAX\\_INVALID\\_CHUNKS\\_LOGGED](#)  
[PROPERTY\\_DEBUG\\_LOG\\_INVALID\\_CHUNK\\_DETAILS](#)

---

## DEFAULT\_PROPERTY\_DEBUG\_MAX\_INVALID\_CHUNKS\_LOGGED

```
public static final int DEFAULT_PROPERTY_DEBUG_MAX_INVALID_CHUNKS_LOGGED
```

Default value for DVR Property "dvrMaxInvalidChunksLogged".

Default value is 10.  
Constant value: **10**

See Also:

[PROPERTY\\_DEBUG\\_LOG\\_INVALID\\_CHUNK\\_DETAILS](#)  
[PROPERTY\\_DEBUG\\_MAX\\_INVALID\\_CHUNKS\\_LOGGED](#)

---

## PROPERTY\_DEBUG\_LOG\_VALID\_CHUNK\_DETAILS

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

DVR Property "dvrLogValidChunkDetails": to control detailed logging information of DVR valid chunks

The default is false, meaning do not log valid chunks. Add the property to Application/DVR/Properties section of Application.xml  
Constant value: **dvrLogValidChunkDetails**

See Also:

[PROPERTY\\_DEBUG\\_MAX\\_VALID\\_CHUNKS\\_LOGGED](#)

---

## PROPERTY\_DEBUG\_MAX\_VALID\_CHUNKS\_LOGGED

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

DVR Property "dvrMaxValidChunksLogged": for controlling maximum number of valid DVR chunks logged.

The default is 10. Logging only occurs if property [PROPERTY\\_DEBUG\\_LOG\\_VALID\\_CHUNK\\_DETAILS](#) is true. Add the property to Application/DVR/Properties section of Application.xml  
Constant value: **dvrMaxValidChunksLogged**

See Also:

[DEFAULT\\_PROPERTY\\_DEBUG\\_MAX\\_VALID\\_CHUNKS\\_LOGGED](#)  
[PROPERTY\\_DEBUG\\_LOG\\_VALID\\_CHUNK\\_DETAILS](#)

---

## DEFAULT\_PROPERTY\_DEBUG\_MAX\_VALID\_CHUNKS\_LOGGED

```
public static final int DEFAULT_PROPERTY_DEBUG_MAX_VALID_CHUNKS_LOGGED
```

Default value for DVR Property "dvrMaxValidChunksLogged".

Default value is 10.  
Constant value: **10**

See Also:

[PROPERTY\\_DEBUG\\_LOG\\_VALID\\_CHUNK\\_DETAILS](#)  
[PROPERTY\\_DEBUG\\_MAX\\_VALID\\_CHUNKS\\_LOGGED](#)

---

---

## PROPERTY\_DEBUG\_LOG\_VALID\_CHUNK\_MATCHER

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

DVR Property "dvrLogValidChunkMatcher": for matching stream names that will log chunk packets

Only affects logging if [PROPERTY\\_DEBUG\\_MAX\\_VALID\\_CHUNKS\\_LOGGED](#) is true.

Constant value: **dvrLogValidChunkMatcher**

See Also:

[PROPERTY\\_DEBUG\\_MAX\\_VALID\\_CHUNKS\\_LOGGED](#)

---

## PROPERTY\_DEBUG\_TOSSED\_HOLDERS

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

DVR Property "dvrDebugTossedHolders": for turning on logging of packets that are being tossed.

Valid values are "true" or "false". Default is false.

Constant value: **dvrDebugTossedHolders**

---

## PROPERTY\_DEBUG\_RAW\_PACKETS

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

DVR Property "dvrDebugRawPackets": for turning on logging of incoming raw packets.

Valid values are "true" or "false".

Constant value: **dvrDebugRawPackets**

See Also:

[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](#)

---

(continued from last page)

---

## PROPERTY\_DEBUG\_MAXIMUM\_RAW\_PACKETS

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

DVR Property "dvrDebugMaximumRawPackets": for setting maximum number of logged raw packets.

Valid values is an integer. 0 means there is no limit.

Only affects logging if [PROPERTY\\_DEBUG\\_RAW\\_PACKETS](#) is true.

Constant value: **dvrDebugMaximumRawPackets**

See Also:

[PROPERTY\\_DEBUG\\_RAW\\_PACKETS](#)

---

## DEFAULT\_PROPERTY\_DEBUG\_MAX\_RAW\_PACKETS

```
public static final int DEFAULT_PROPERTY_DEBUG_MAX_RAW_PACKETS
```

Default value for DVR Property "dvrDebugMaximumRawPackets".

Default value is 200 log statements.

Constant value: **200**

See Also:

[PROPERTY\\_DEBUG\\_MAXIMUM\\_RAW\\_PACKETS](#)

---

## PROPERTY\_DEBUG\_STATE\_CHANGE

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

DVR Property "dvrDebugStateChange": for logging state changes of DVR store.

Constant value: **dvrDebugStateChange**

---

## PROPERTY\_DEBUG\_CHUNK\_RETRIEVALS

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

DVR Property "dvrDebugChunkRetrievals": for logging each chunk retrieval

Constant value: **dvrDebugChunkRetrievals**

---

## PROPERTY\_DEBUG\_FAILED\_CHUNK\_RETRIEVALS

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

DVR Property "dvrDebugFailedChunkRetrievals": for logging info about each failed chunk retrieval

Constant value: **dvrDebugFailedChunkRetrievals**

---

## PROPERTY\_MAX\_CHUNK\_LOG

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

DVR Property "dvrMaxChunkLogCount": for maximum number of DVR chunks to log.

Maximum number of chunks to log recording information about in the wowzamedia\_access.log file. As recording is continued past this value, there is no feedback for normal operation in the log. View directories and files created in [install-dir]/dvr for on-going feedback that recording is occurring. Add the property to Application/DVR/Properties section of Application.xml

Constant value: **dvrMaxChunkLogCount**

See Also:

[DEFAULT\\_PROPERTY\\_DVR\\_MAX\\_CHUNK\\_LOG](#)

---

---

## DEFAULT\_PROPERTY\_DVR\_MAX\_CHUNK\_LOG

```
public static final int DEFAULT_PROPERTY_DVR_MAX_CHUNK_LOG
```

Default value for DVR Property "dvrMaxChunkLogCount".

Default value is 10 logging statements.  
Constant value: **10**

See Also:

[PROPERTY\\_MAX\\_CHUNK\\_LOG](#)

---

## PROPERTY\_MAX\_RECALC\_DURATION\_LOG

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

DVR Property "dvrMaxRecountDurationLogCount": for maximum number of messages to log regarding recalculating DVR chunk duration

Maximum number of log statements recorded regarding chunk duration being recalculated. A value less than 0 means all statements will be logged with no limit.

Constant value: **dvrMaxRecountDurationLogCount**

See Also:

[DEFAULT\\_PROPERTY\\_MAX\\_RECALC\\_DURATION\\_LOG](#)

---

## DEFAULT\_PROPERTY\_MAX\_RECALC\_DURATION\_LOG

```
public static final int DEFAULT_PROPERTY_MAX_RECALC_DURATION_LOG
```

Default value for DVR Property "dvrMaxRecountDurationLogCount".

Default value is 30 logging statements.  
Constant value: **30**

See Also:

[PROPERTY\\_MAX\\_RECALC\\_DURATION\\_LOG](#)

---

## PROPERTY\_DEBUG\_METHODS

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

DVR Property "dvrDebugManagerLogMethods": for turning on DVR Manager debug logging.

Valid values are "true" or "false".

Add the property to Application/DVR/Properties section of Application.xml  
Constant value: **dvrDebugManagerLogMethods**

---

## PROPERTY\_DEBUG\_REPEATER

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

DVR Property "dvrDebugRepeater": for turning on logging of DVR repeater

Valid values are "true" or "false".

Constant value: **dvrDebugRepeater**

---

(continued from last page)

---

## 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**

---

## 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
```

---

(continued from last page)

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**

---

## PROPERTY\_SANJOSE\_PLAYLIST\_MIMETYPE

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

DVR Property "dvrSanJosePlaylistMimeType": used to over-ride the mime type for DVR San Jose F4m playlists.

Default value is "video/mp4"

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

Constant value: **dvrSanJosePlaylistMimeType**

---

## PROPERTY\_SANJOSE\_PLAYLIST\_DELIVERYTYPE

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

(continued from last page)

DVR Property "dvrSanJosePlaylistDeliveryType": used to over-ride the mime type for DVR San Jose F4m playlists.

Default value is "streaming"

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

Constant value: **dvrSanJosePlaylistDeliveryType**

---

## PROPERTY\_SANJOSE\_PLAYLIST\_VERSION

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

DVR Property "dvrSanJosePlaylistVersion": used to over-ride the version of the San Jose f4m playlist.

Default value is "2.0". Valid values are "1.0" and "2.0".

Constant value: **dvrSanJosePlaylistVersion**

See Also:

[SANJOSE\\_F4M\\_VERSION\\_1\\_0](#)

[Add this to Application/DVR/Properties section of Application.xml](#)

---

## PROPERTY\_SANJOSE\_PLAYLIST\_RECORDED\_STREAMTYPE

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

DVR Property "dvrSanJosePlaylistRecordedStreamType": used to over-ride the stream type for non-live (recorded) DVR in San Jose f4m playlists.

Default value is #SANJOSE\_F4M\_STREAMTYPE\_RECORDED.\*

Constant value: **dvrSanJosePlaylistRecordedStreamType**

See Also:

[SANJOSE\\_F4M\\_STREAMTYPE\\_DVR](#)

[SANJOSE\\_F4M\\_STREAMTYPE\\_LIVE](#)

[SANJOSE\\_F4M\\_STREAMTYPE\\_LIVEORRECORDED](#)

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

---

## PROPERTY\_SANJOSE\_PLAYLIST\_LIVE\_STREAMTYPE

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

DVR Property "dvrSanJosePlaylistLiveStreamType": used to over-ride the stream type for live DVR in San Jose f4m playlists.

Default value is #SANJOSE\_F4M\_STREAMTYPE\_DVR.

Constant value: **dvrSanJosePlaylistLiveStreamType**

See Also:

[SANJOSE\\_F4M\\_STREAMTYPE\\_DVR](#)

[SANJOSE\\_F4M\\_STREAMTYPE\\_LIVE](#)

[SANJOSE\\_F4M\\_STREAMTYPE\\_LIVEORRECORDED](#)

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

---

## SANJOSE\_F4M\_STREAMTYPE\_DVR

```
public static final java.lang.String SANJOSE_F4M_STREAMTYPE_DVR
```

Constant for San Jose stream type "dvr".

---



(continued from last page)

Constant value: **dvr**

---

## SANJOSE\_F4M\_STREAMTYPE\_RECORDED

```
public static final java.lang.String SANJOSE_F4M_STREAMTYPE_RECORDED
```

Constant for San Jose stream type "recorded".  
Constant value: **recorded**

---

## SANJOSE\_F4M\_STREAMTYPE\_LIVE

```
public static final java.lang.String SANJOSE_F4M_STREAMTYPE_LIVE
```

Constant for San Jose stream type "live".  
Constant value: **live**

---

## SANJOSE\_F4M\_STREAMTYPE\_LIVEORRECORDED

```
public static final java.lang.String SANJOSE_F4M_STREAMTYPE_LIVEORRECORDED
```

Constant for San Jose stream type "liveOrRecorded".  
Constant value: **liveOrRecorded**

---

## SANJOSE\_F4M\_VERSION\_1\_0

```
public static final java.lang.String SANJOSE_F4M_VERSION_1_0
```

Constant for San Jose f4m version "1.0".  
Constant value: **1.0**

---

## SANJOSE\_F4M\_VERSION\_2\_0

```
public static final java.lang.String SANJOSE_F4M_VERSION_2_0
```

Constant for San Jose f4m version "2.0".  
Constant value: **2.0**

---

## SANJOSE\_F4M\_STREAMINGTYPE\_STREAMING

```
public static final java.lang.String SANJOSE_F4M_STREAMINGTYPE_STREAMING
```

Constant for San Jose streaming type "streaming".  
Constant value: **streaming**

---

## MIMETYPE\_VIDEO\_MP4

```
public static final java.lang.String MIMETYPE_VIDEO_MP4
```

Constant for mime type "video/mp4".  
Constant value: **video/mp4**

---

## PROPERTY\_SANJOSE\_ABST\_TIMESCALE

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

DVR Property "dvrSanJosePlaylistAbstTimescale": used to over-ride the time-scale for abst files.

Default value is #SANJOSE\_ABST\_DEFAULT\_TIMESCALE (1000), which means milliseconds.

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

(continued from last page)

Constant value: **dvrSanJosePlaylistAbstTimescale**

---

## PROPERTY\_SANJOSE\_ABST\_DURATION\_TOLERANCE

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

DVR Property "dvrSanJosePlaylistAbstDurationEqualityTolerance": used to over-ride the tolerance when determining equal chunks lengths.

Default value is #SANJOSE\_ABST\_DEFAULT\_DURATION\_EQUALITY\_TOLERANCE (50), which means 50 milliseconds.

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

Constant value: **dvrSanJosePlaylistAbstDurationTolerance**

---

## DEFAULT\_PROPERTY\_SANJOSE\_ABST\_TIMESCALE

```
public static final int DEFAULT_PROPERTY_SANJOSE_ABST_TIMESCALE
```

Default value for property "dvrSanJosePlaylistAbstTimescale" #see [PROPERTY\\_SANJOSE\\_ABST\\_TIMESCALE](#)

Constant value: **1000**

---

## DEFAULT\_PROPERTY\_SANJOSE\_ABST\_DURATION\_TOLERANCE

```
public static final int DEFAULT_PROPERTY_SANJOSE_ABST_DURATION_TOLERANCE
```

Default value for property "dvrSanJosePlaylistAbstDurationTolerance" #see [PROPERTY\\_SANJOSE\\_ABST\\_DURATION\\_TOLERANCE](#)

Constant value: **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
```

DVR Property "dvrCupertinoPlaylistForceLive": used to override playlist request delegate logic that determines if playlist is live versus non-live.

Default value is false

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

Constant value: **dvrCupertinoPlaylistForceNonLive**

---

## PROPERTY\_CUPERTINO\_PLAYLIST\_USE\_GZIP

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

DVR Property "dvrCupertinoPlaylistUseGzip": used to force Cupertino playlist to use gzip if it is accepted

Default value is true

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

(continued from last page)

Constant value: **dvrCupertinoPlaylistUseGzip**

---

## PROPERTY\_CUPERTINO\_PLAYLIST\_GZIP\_THRESHOLD

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

DVR Property "dvrCupertinoPlaylistGzipThreshold": when playlist is larger than this number of bytes, and gzip is enabled and accepted, the playlist will be compressed

Default value is [DEFAULT\\_CUPERTINO\\_PLAYLIST\\_GZIP\\_THRESHOLD](#)

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

Constant value: **dvrCupertinoPlaylistGzipThreshold**

---

## DEFAULT\_CUPERTINO\_PLAYLIST\_GZIP\_THRESHOLD

```
public static final int DEFAULT_CUPERTINO_PLAYLIST_GZIP_THRESHOLD
```

Default value for property "dvrCupertinoPlaylistGzipThreshold" #see  
[PROPERTY\\_CUPERTINO\\_PLAYLIST\\_GZIP\\_THRESHOLD](#)

Constant value: **4000**

---

## PROPERTY\_CUPERTINO\_ON\_CHUNK\_START\_RESET\_COUNTER

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

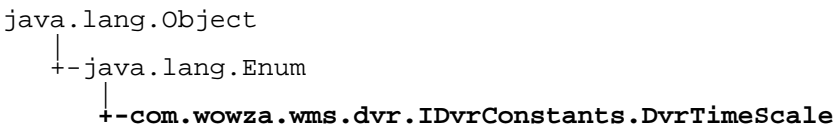
DVR Property "dvrCupertinoOnChunkStartResetCounter": when a new chunk starts, reset internal tsPacketizer counters

Default value is false

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

Constant value: **dvrCupertinoOnChunkStartResetCounter**

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	<a href="#">DVR_TIME</a>
public static final	<a href="#">PACKET_TIME</a>
public static final	<a href="#">UTC_TIME</a>

Method Summary

static <a href="#">IDvrConstants.DvrTimeScale</a>	<a href="#">valueOf</a> (String name)
static <a href="#">IDvrConstants.DvrTimeScale[]</a>	<a href="#">values</a> ()

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	<a href="#">ENCRYPTION_TYPE_CUPERTINO</a> Cupertino AES encryption type. Value: <b>1</b>
public static final	<a href="#">ENCRYPTION_TYPE_PLAYREADY</a> Playready encryption type. Value: <b>2</b>

### Method Summary

Object	<a href="#">clone()</a>
int	<a href="#">getEncryptionType()</a> Serialize encryption info.
int	<a href="#">getSerializeSize()</a> Return number of bytes required to serialize this encryption info.
byte[]	<a href="#">serialize()</a> Serialize the encryption info, returning a byte buffer containing the serialized data.
int	<a href="#">serialize(byte[] buffer, int pos)</a> 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	<a href="#">CODEC_TYPE</a> Constant for codec manifest type. Value: <b>999</b>
public static final	<a href="#">MANIFEST_TAGNAME_CAN_PLAY</a> Value: <b>CanPlay</b>
public static final	<a href="#">MANIFEST_TAGNAME_CAN_RECORD</a> Value: <b>CanRecord</b>
public static final	<a href="#">MANIFEST_TAGNAME_CHUNK_GROUPING</a> Value: <b>ChunkGrouping</b>
public static final	<a href="#">MANIFEST_TAGNAME_CURRENT_TIME</a> Value: <b>CurrentTime</b>
public static final	<a href="#">MANIFEST_TAGNAME_HAS_ENCRYPTION</a> Value: <b>HasEncryption</b>
public static final	<a href="#">MANIFEST_TAGNAME_PURGE_TIME</a> Value: <b>PurgeTime</b>
public static final	<a href="#">ON_METADATA_TYPE</a> Constant for 'onMetadata' manifest type. Value: <b>0</b>
public static final	<a href="#">TIME_MAP_TYPE</a> Constant for time map manifest type. Value: <b>998</b>



## Method Summary

void	<a href="#"><code>addToManifest</code></a> (java.util.List entries) Add manifest entries to the manifest
void	<a href="#"><code>deserialize</code></a> (byte[] bytes) Deserialize the manifest.
long	<a href="#"><code>expandEndTime</code></a> (int type, long dvrEndTime)
long	<a href="#"><code>expandStartTime</code></a> (int type, long dvrStartTime)
long	<a href="#"><code>getClosestStartTime</code></a> (int type, long dvrTime) Given a dvrTime and a manifest type, find the closest chunk starting time.
<a href="#"><code>DvrManifestCodecEntry</code></a>	<a href="#"><code>getCodecEntryForTime</code></a> (long dvrTime) Return most recent codec entry for a given DVR time.
long	<a href="#"><code>getDvrTime</code></a> (int type)
<a href="#"><code>DvrManifestEntryRangeGroup</code></a>	<a href="#"><code>getEntriesToPurge</code></a> (long purgeTime) Given a purge time, return a group of manifest ranges to purge.
<a href="#"><code>DvrManifestEntry</code></a>	<a href="#"><code>getFirstEntry</code></a> (int type) Given manifest type, get the first playlist manifest entry.
<a href="#"><code>DvrManifestEntry</code></a>	<a href="#"><code>getLastLiveEntry</code></a> (int type) Given manifest type, get the last playlist manifest entry.
<a href="#"><code>DvrManifestEntry</code></a>	<a href="#"><code>getLastRecordedEntry</code></a> (int type) Given manifest type and DVR time, get last entry.
int	<a href="#"><code>getLastRecordedIndex</code></a> (int type) Return last index of DVR entry for given type.
long	<a href="#"><code>getLiveDuration</code></a> (int type) Given manifest type, return DVR live duration in seconds
java.util.List	<a href="#"><code>getLiveEntries</code></a> (int type, long dvrStart) Given manifest type and DVR time, get a list of the live entries.
java.util.List	<a href="#"><code>getLiveEntriesWithLimit</code></a> (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	<a href="#"><code>getManifestChannel</code></a> (int type) For a given manifest type, return the channel manifest.
<a href="#"><code>DvrManifestOnMetadataEntry</code></a>	<a href="#"><code>getMetadataEntryForTime</code></a> (long dvrTime) Return most recent metadataEntry for a given DVR time.
int	<a href="#"><code>getNextChunkIndex</code></a> ()
int	<a href="#"><code>getNextCodecIndex</code></a> ()
int	<a href="#"><code>getNextMetadataIndex</code></a> ()

int	<a href="#"><u>getNextTimeMapIndex()</u></a>
int	<a href="#"><u>getNumberLiveEntries</u></a> (int type, long dvrStart) Given manifest type and DVR time, get the number of live entries at the given time.
int	<a href="#"><u>getNumberLiveEntries</u></a> (int type, long dvrStart, long dvrEnd) Given manifest type and DVR time, get the number of live entries at the given time.
int	<a href="#"><u>getNumberRecordedEntries</u></a> (int type, long dvrStart) Given manifest type and DVR time, get the number of entries at the given time.
int	<a href="#"><u>getNumberRecordedEntries</u></a> (int type, long dvrStart, long dvrEnd) Given manifest type and DVR time, get the number of live entries at the given time.
long	<a href="#"><u>getRecordedDuration</u></a> (int type) Given manifest type, return DVR recorded duration in seconds
java.util.List	<a href="#"><u>getRecordedEntries</u></a> (int type) Get a copy of all manifest entries of a given type.
java.util.List	<a href="#"><u>getRecordedEntries</u></a> (int type, long dvrStartTime) Given manifest type and DVR time, get a list of recorded entries.
java.util.List	<a href="#"><u>getRecordedEntries</u></a> (int type, long dvrStartTime, long dvrEndTime) Given manifest type and DVR time, get a list of recorded entries.
java.util.List	<a href="#"><u>getRecordedEntriesInRange</u></a> ( <a href="#"><u>DvrManifestEntryRange</u></a> range) Given a DvrManifestEntryRange, get a list of recorded entries in this range.
java.util.Map	<a href="#"><u>getRecordedEntriesMap</u></a> (int type) Get a copy of all manifest entries of a given type as a Map of indices.
java.util.List	<a href="#"><u>getRecordedEntriesWithLimit</u></a> (int type, long dvrTime, int limit) Given manifest type and DVR time, get a list of recorded entries limiting number of returned items.
<a href="#"><u>DvrManifestEntry</u></a>	<a href="#"><u>getRecordedEntryByIndex</u></a> (int type, int index) Get the manifest entry given a manifest type and an index.
<a href="#"><u>DvrManifestEntry</u></a>	<a href="#"><u>getRecordedEntryByTimeKey</u></a> (int type, long dvrTime) Get the manifest entry given a manifest type and a time (in DVR units)
<a href="#"><u>IDvrTimeMap</u></a>	<a href="#"><u>getTimeMap</u></a> ()
boolean	<a href="#"><u>hasAudio</u></a> () Does manifest contain audio.
boolean	<a href="#"><u>hasCodecData</u></a> () Does manifest contain codec data.
boolean	<a href="#"><u>hasData</u></a> () Does manifest contain data.
boolean	<a href="#"><u>hasOnMetadata</u></a> () Does manifest contain onMetadata.
boolean	<a href="#"><u>hasTimeMapData</u></a> () Does manifest contain time map info.

boolean	<a href="#"><code>hasVideo()</code></a> Does manifest contain video.
void	<a href="#"><code>importManifest(IDvrManifest manifest, boolean persist)</code></a> Import the specified manifest into this manifest
void	<a href="#"><code>initialize()</code></a> Initialize the manifest.
java.util.List	<a href="#"><code>purgeEntries(DvrManifestEntryRangeGroup ranges)</code></a> Given a group of ranges, purge the manifest entries.
void	<a href="#"><code>refreshManifest()</code></a> Refresh the manifest.
byte[]	<a href="#"><code>serialize(boolean ignoreEntries)</code></a> 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	<a href="#">createDvrAudioChunk</a> (long dvrTime, int index, int duration, <a href="#">DvrPacketHolder</a> holder) Create audio chunk from set of packets.
DvrChunk	<a href="#">createDvrDataChunk</a> (long dvrTime, int index, int duration, <a href="#">DvrPacketHolder</a> holder) Create data chunk from set of packets.
DvrChunk	<a href="#">createDvrOnMetadataChunk</a> (long dvrTime, long pt, <a href="#">AMFPacket</a> metaPacket) Create onMetadata chunk from set of packets.
DvrChunk	<a href="#">createDvrVideoChunk</a> (long dvrTime, int index, int duration, <a href="#">DvrPacketHolder</a> 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	<a href="#"><u>addToChunk</u></a> (DvrPacketHolder holder)
boolean	<a href="#"><u>canRecordAudio</u></a> ()
boolean	<a href="#"><u>canRecordData</u></a> ()
boolean	<a href="#"><u>canRecordVideo</u></a> ()
void	<a href="#"><u>endChunk</u></a> (long videoEndTime, long audioEndTime)
void	<a href="#"><u>resetStream</u></a> ()
void	<a href="#"><u>sendOnMetadata</u></a> (long pt, <a href="#"><u>AMFPacket</u></a> metaPacket)
void	<a href="#"><u>setCodecInfoAudio</u></a> (com.wowza.wms.media.model.MediaCodecInfoAudio codecInfoAudio)
void	<a href="#"><u>setCodecInfoVideo</u></a> (com.wowza.wms.media.model.MediaCodecInfoVideo codecInfoVideo)
void	<a href="#"><u>startChunk</u></a> (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	<a href="#"><u>QUERY_PARAM_DVR</u></a> Value: <b>DVR</b>
---------------------	---

### Method Summary

int	<a href="#"><u>calcBitrate</u></a> ( <a href="#"><u>IDvrStreamStore</u></a> store, int type, long t)
String	<a href="#"><u>determineStreamVersion</u></a> ( <a href="#"><u>IDvrStreamManager</u></a> dvrMgr, String baseStreamName)
DvrPlaylistRequest	<a href="#"><u>getDvrPlaylistRequest</u></a> ( <a href="#"><u>IHTTPStreamerApplicationContext</u></a> appContext, <a href="#"><u>IDvrStreamStore</u></a> store, java.util.Map queryMap)
DvrPlaylistRequest	<a href="#"><u>getDvrPlaylistRequest</u></a> ( <a href="#"><u>IHTTPStreamerApplicationContext</u></a> appContext, java.util.List stores, java.util.Map queryMap)
IDvrMbrPlaylistAlignment	<a href="#"><u>getPlaylistAlignment</u></a> ( <a href="#"><u>IHTTPStreamerSession</u></a> httpStreamerSession)
boolean	<a href="#"><u>isPlaylistReady</u></a> ( <a href="#"><u>IDvrStreamStore</u></a> store, DvrPlaylistRequest playlistRequest)
boolean	<a href="#"><u>isPlaylistReady</u></a> (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	<a href="#"><code>getCurrentTime()</code></a> Get current DVR time.
long	<a href="#"><code>getLastPurgeTime()</code></a> Get the DVR time when the last purge occurred.
void	<a href="#"><code>init(IDvrStreamStore store)</code></a> Initialize the controller.
boolean	<a href="#"><code>isPurgingEnabled()</code></a> Is Purging enabled for this controller.
void	<a href="#"><code>setCurrentDvrTime(long newDvrTime)</code></a> 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	<a href="#">retrieveRawChunk</a> ( <a href="#">DvrManifestChunkEntry</a> 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	<a href="#">recordingPaused</a> ( <a href="#">IDvrStreamStore</a> store) Callback when DVR moves to paused state.
void	<a href="#">recordingReset</a> ( <a href="#">IDvrStreamStore</a> store) Callback when DVR gets reset.
void	<a href="#">recordingResumed</a> ( <a href="#">IDvrStreamStore</a> store) Callback when DVR moves out of paused state.
void	<a href="#">recordingStarted</a> ( <a href="#">IDvrStreamStore</a> store) Callback when DVR moves to recording state.
void	<a href="#">recordingStopped</a> ( <a href="#">IDvrStreamStore</a> store) Callback when DVR recording stops
void	<a href="#">timeReset</a> ( <a href="#">IDvrStreamStore</a> 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	<a href="#">init</a> ( <a href="#">IDvrStreamManager</a> dvrMgr) Initialize recordings loader
void	<a href="#">loadArchivedRecordings</a> () Discover and load archived recording
boolean	<a href="#">shouldLoadStream</a> (String streamName, java.util.SortedSet versions) Should the given archived streams be laoded.
boolean	<a href="#">shouldLoadStreamVersion</a> (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	<a href="#">postChunkAdded</a> ( <a href="#">IDvrStreamStore</a> store) Callback after a chunk is written to DVR store
void	<a href="#">postChunksPurged</a> ( <a href="#">IDvrStreamStore</a> store, <a href="#">DvrManifestEntryRangeGroup</a> entries, java.util.List deletedEntries) Callback after DVR store performs a purge
void	<a href="#">preChunkAdded</a> ( <a href="#">IDvrStreamStore</a> store) Callback before a chunk is written to DVR store
void	<a href="#">preChunksPurged</a> ( <a href="#">IDvrStreamStore</a> store, <a href="#">DvrManifestEntryRangeGroup</a> 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	<a href="#">dvrStreamStorageDeleted</a> ( <a href="#">IDvrStreamStore</a> store, <a href="#">IDvrFileSystem</a> <a href="#">FileSystem</a> , boolean success) Callback after DVR store is deleted from disk.
void	<a href="#">dvrStreamStorageLoaded</a> ( <a href="#">IDvrStreamManager</a> dvrMgr, <a href="#">IDvrStreamStore</a> store) Callback after DVR store is loaded from disk.
void	<a href="#">dvrStreamStoreCreate</a> ( <a href="#">IDvrStreamStore</a> store) Callback after DVR stream store is created.
void	<a href="#">dvrStreamStoreDestroy</a> ( <a href="#">IDvrStreamStore</a> store) Callback after DVR stream store is destroyed.
void	<a href="#">dvrStreamStoreInit</a> ( <a href="#">IDvrStreamStore</a> 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	<a href="#">addDvrRecordingListener</a> ( <a href="#">IDvrRecordingListener</a> listener)	Add listener to DVR recording events.
void	<a href="#">addDvrStoreListener</a> ( <a href="#">IDvrStoreListener</a> listener)	Add listener to DVR store lifecycle events.
void	<a href="#">addManifestEntries</a> (String vStreamName, java.util.List entries)	Add to stream stores manifest.
void	<a href="#">addRepeaterHeartBeatItem</a> ()	
boolean	<a href="#">canRecord</a> ()	Is this stream able to record.
void	<a href="#">deleteArchivedStore</a> (String vStreamName)	Delete archived stream store.
String	<a href="#">getArchiveStrategy</a> ()	Get the DVR archive strategy.
String	<a href="#">getContextStr</a> ()	Get stream context string, useful for logging.
<a href="#">IDvrStreamStore</a>	<a href="#">getDefaultStreamingStore</a> ()	Get the store to be used for streaming.
String	<a href="#">getDvrFileSystemClass</a> ()	Get the class used for the DVR file system.
String	<a href="#">getDvrStorageDir</a> ()	Get the storage directory.
int	<a href="#">getDvrStorageWindowSeconds</a> ()	Get DVR window size.
DvrBaseEncryptionInfo Delegate	<a href="#">getEncryptionDelegate</a> ()	Get the encryption delegate for providing streaming side encryption info objects.
String	<a href="#">getEncryptionRepeaterSharedSecret</a> ()	Get the DVR encryption shared secret.

<a href="#">IDvrStreamStore</a>	<a href="#">getHighestVersionedStore()</a> Return stream store that is highest known version.
void	<a href="#">getInitialRepeaterItems</a> (java.util.List items) Get initial repeater items to send to repeater receiver.
int	<a href="#">getMinimumAvailableChunks()</a> Return number of chunks that must be available to stream.
String	<a href="#">getPacketizerName()</a> Get live stream packetizer name.
<a href="#">IDvrStreamStore</a>	<a href="#">getRecordingStreamStore()</a> Get current recording store
void	<a href="#">getRepeaterItemsDvr</a> (java.util.List items, long lastSeq)
String	<a href="#">getStreamBaseName()</a> Get input streamName (no version info).
<a href="#">IDvrStreamStore</a>	<a href="#">getStreamStore</a> (String vStreamName) Given a stream name containing version info, return the associated stream store.
java.util.List	<a href="#">getStreamStores()</a> Get list of all stream stores known to this stream manager.
<a href="#">IDvrStreamVersionHandler</a>	<a href="#">getStreamVersionHandler()</a> Get the stream version handler object.
void	<a href="#">initialManifest</a> (String vStreamName, <a href="#">IDvrManifest</a> manifest)
void	<a href="#">initialManifestEnd</a> (String vStreamName)
boolean	<a href="#">isRecording()</a> Is this stream currently recording.
boolean	<a href="#">isRecordingPaused()</a> Is this stream currently paused from recording.
<a href="#">IDvrStreamStore</a>	<a href="#">loadArchivedStore</a> (String vStreamName, DvrManifestHolder manifestHolder) Load archived stream store.
void	<a href="#">notifyDvrStreamStorageDeleted</a> ( <a href="#">IDvrStreamStore</a> store, IDvrFileSystem fileSystem, boolean success)
void	<a href="#">notifyDvrStreamStoreCreate</a> ( <a href="#">IDvrStreamStore</a> store)
void	<a href="#">notifyDvrStreamStoreDestroy</a> ( <a href="#">IDvrStreamStore</a> store)
void	<a href="#">notifyDvrStreamStoreInit</a> ( <a href="#">IDvrStreamStore</a> store)
void	<a href="#">notifyDvrStreamStoreLoaded</a> ( <a href="#">IDvrStreamStore</a> store)
void	<a href="#">notifyTimeReset</a> ( <a href="#">IDvrStreamStore</a> store, long oldDvrTime, long oldPacketTime, TimeMapRecord newTimeMapRecord)



<a href="#">IDvrStreamStore</a>	<a href="#">pauseRecording()</a> Request that stream recording pause.
void	<a href="#">purgeManifestEntries</a> (String vStreamName, <a href="#">DvrManifestEntryRangeGroup</a> rangeGroup) Purge entries from store
void	<a href="#">removeDvrRecordingListener</a> ( <a href="#">IDvrRecordingListener</a> listener) Remove listener to DVR recording events.
void	<a href="#">removeDvrStoreListener</a> ( <a href="#">IDvrStoreListener</a> listener) Remove listener of DVR store lifecycle events.
void	<a href="#">resetStream()</a> Reset the stream.
<a href="#">IDvrStreamStore</a>	<a href="#">resumeRecording()</a> Request that stream recording resume.
void	<a href="#">setDefaultStreamingStore</a> ( <a href="#">IDvrStreamStore</a> store) Set the store to be used for streaming.
void	<a href="#">setRecordingStreamStore</a> ( <a href="#">IDvrStreamStore</a> store) Set the store used for recording.
void	<a href="#">setStreamVersionHandler</a> ( <a href="#">IDvrStreamVersionHandler</a> handler) Set the stream version handler object.
<a href="#">IDvrStreamStore</a>	<a href="#">startRecording()</a> Request that stream recording start.
void	<a href="#">stateChange</a> (String vStreamName, <a href="#">DvrStreamStoreState</a> state)
<a href="#">IDvrStreamStore</a>	<a href="#">stopRecording()</a> Request that stream recording stop.
long	<a href="#">storeChunks</a> (int vDuration, <a href="#">DvrPacketHolder</a> vPackets, int aDuration, <a href="#">DvrPacketHolder</a> aPackets, int dDuration, <a href="#">DvrPacketHolder</a> dPackets)
boolean	<a href="#">storeOnMetadata</a> (long pt, long utc, <a href="#">AMFPacket</a> 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	<a href="#"><u>addDvrChunkListener</u></a> ( <a href="#"><u>IDvrStoreChunkListener</u></a> listener) Add (very fine) listener for chunk events.
void	<a href="#"><u>addManifestEntries</u></a> (java.util.List entries)
boolean	<a href="#"><u>canPlay</u></a> () Is this store capable of playing.
boolean	<a href="#"><u>canRecord</u></a> () Is this store capable of recording.
<a href="#"><u>IApplicationInstance</u></a>	<a href="#"><u>getAppInstance</u></a> () Get associated application instance.
<a href="#"><u>IDvrChunker</u></a>	<a href="#"><u>getChunker</u></a> ()
long	<a href="#"><u>getClosestStartTime</u></a> (int type, long t)
String	<a href="#"><u>getContextStr</u></a> () Get stream context string, useful for logging.
DvrChunk	<a href="#"><u>getDvrChunkAtTime</u></a> (int fragmentType, long t)
DvrChunk	<a href="#"><u>getDvrChunkByIndex</u></a> (int fragmentType, int index)
DvrChunk	<a href="#"><u>getDvrChunkNearTime</u></a> (int fragmentType, long t, long delta)
<a href="#"><u>IDvrStreamManager</u></a>	<a href="#"><u>getDvrManager</u></a> () Get DVR Stream Manager
int	<a href="#"><u>getDvrStorageWindowSeconds</u></a> () Get DVR window size.
IDvrFileSystem	<a href="#"><u>getFileSystem</u></a> ()
<a href="#"><u>IDvrManifest</u></a>	<a href="#"><u>getManifest</u></a> () Get manifest
<a href="#"><u>WMSProperties</u></a>	<a href="#"><u>getProperties</u></a> () Get stream store properties.
<a href="#"><u>IDvrPurgeController</u></a>	<a href="#"><u>getPurgeController</u></a> () Get purge controller for stream store.

java.util.List	<a href="#"><u>getRecordedEntriesWithLimit</u></a> (int fragmentType, long t, int limit)
<a href="#"><u>DvrManifestEntry</u></a>	<a href="#"><u>getRecordedEntryByIndex</u></a> (int fragmentType, int index)
<a href="#"><u>DvrManifestEntry</u></a>	<a href="#"><u>getRecordedEntryByTime</u></a> (int fragmentType, long t)
String	<a href="#"><u>getStreamName</u></a> () Get versioned stream name.
DvrTimeMapper	<a href="#"><u>getTimeMapper</u></a> () Get time mapper, which maps between DVR, real and packet time.
boolean	<a href="#"><u>hasAudio</u></a> () Does this store have audio.
boolean	<a href="#"><u>hasData</u></a> () Does this store have data.
boolean	<a href="#"><u>hasEncryption</u></a> () Does this store contain encryption information
boolean	<a href="#"><u>hasOnMetadata</u></a> () Does this store have on metadata.
boolean	<a href="#"><u>hasVideo</u></a> () Does this store have video.
void	<a href="#"><u>init</u></a> () Initialize DVR stream store.
boolean	<a href="#"><u>isLive</u></a> () Is store currently live For an origin, <a href="#"><u>isRecording()</u></a> and <a href="#"><u>isLive()</u></a> will typically return the same result.
boolean	<a href="#"><u>isLoading</u></a> ()
boolean	<a href="#"><u>isRecording</u></a> () Is store currently recording.
boolean	<a href="#"><u>isRecordingPaused</u></a> () Is store currently paused while recording.
boolean	<a href="#"><u>pauseRecording</u></a> () Request that stream recording pause.
void	<a href="#"><u>purgeEntries</u></a> ( <a href="#"><u>DvrManifestEntryRangeGroup</u></a> rangeGroup) Purge entries from store
void	<a href="#"><u>removeDvrChunkListener</u></a> ( <a href="#"><u>IDvrStoreChunkListener</u></a> listener) Remove (very fine) listener for chunk events.
void	<a href="#"><u>reset</u></a> ()
boolean	<a href="#"><u>resumeRecording</u></a> () Request that stream recording resume.

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

(continued from last page)

Get DVR window size. This may be set by setting the IDvrStreamStore's property [IDvrConstants.PROPERTY\\_WINDOW\\_DURATION](#) after the store is created and before it is initialized by using the [IDvrStoreListener.dvrStreamStoreCreate\(IDvrStreamStore\)](#)

**Returns:**

window size in seconds (0 means infinite window size)

---

## getStreamName

```
public String getStreamName()
```

Get versioned stream name.

**Returns:**

(versioned) stream name

---

## getContextStr

```
public String getContextStr()
```

Get stream context string, useful for logging.

**Returns:**

stream context string

---

## getDvrManager

```
public IDvrStreamManager getDvrManager()
```

Get DVR Stream Manager

**Returns:**

DVR Stream Manager

---

## getAppInstance

```
public IApplicationInstance getAppInstance()
```

Get associated application instance.

**Returns:**

application instance.

---

## getManifest

```
public IDvrManifest getManifest()
```

Get manifest

**Returns:**

manifest

---

## getTimeMapper

```
public DvrTimeMapper getTimeMapper()
```

Get time mapper, which maps between DVR, real and packet time.

**Returns:**

(continued from last page)

time mapper

---

## getPurgeController

```
public IDvrPurgeController getPurgeController()
```

Get purge controller for stream store.

**Returns:**

purge controller

---

## getFileSystem

```
public IDvrFileSystem getFileSystem()
```

---

## getChunker

```
public IDvrChunker getChunker()
```

---

## canRecord

```
public boolean canRecord()
```

Is this store capable of recording.

**Returns:**

true if store can record

---

## isLoaded

```
public boolean isLoaded()
```

---

## setCanRecord

```
public void setCanRecord(boolean canRecord)
```

Set the Stream Stores ability to record.

Note: Must be called when stream is not recording.

**Parameters:**

canRecord

---

## canPlay

```
public boolean canPlay()
```

Is this store capable of playing.

**Returns:**

true if store can play

---

## setCanPlay

```
public void setCanPlay(boolean canPlay)
```

Set the Stream Stores ability to play.

Note: Must be called when stream is not playing.

**Parameters:**

canPlay

---

## setHasEncryption

```
public void setHasEncryption(boolean hasEncryption)
```

Set whether the store has encryption. Used internally.

Note: Must be called when stream is not playing.

**Parameters:**

hasEncryption - if store has encryption

---

## hasEncryption

```
public boolean hasEncryption()
```

Does this store contain encryption information

**Returns:**

true if store contains encryption

---

## hasVideo

```
public boolean hasVideo()
```

Does this store have video.

**Returns:**

true if store contains video, false otherwise.

---

## hasAudio

```
public boolean hasAudio()
```

Does this store have audio.

**Returns:**

true if store contains audio, false otherwise.

---

## hasData

```
public boolean hasData()
```

Does this store have data.

**Returns:**

true if store contains data, false otherwise.

---

(continued from last page)

## hasOnMetadata

```
public boolean hasOnMetadata( )
```

Does this store have on metadata.

**Returns:**

true if store contains on metadata, false otherwise.

---

## startRecording

```
public boolean startRecording( )
```

Request that recording start.

Used internally. Clients should call [ILiveStreamDvrRecorder.startRecording\(\)](#).

Success occurs if the stream canRecord [canRecord\(\)](#) and the stream is not already in the recording state [isRecording\(\)](#).

**Returns:**

true if successful.

---

## pauseRecording

```
public boolean pauseRecording( )
```

Request that stream recording pause.

Used internally. Clients should call [ILiveStreamDvrRecorder.pauseRecording\(\)](#).

Success occurs if the stream is not already paused.

**Returns:**

true if successful.

---

## resumeRecording

```
public boolean resumeRecording( )
```

Request that stream recording resume.

Used internally. Clients should call [ILiveStreamDvrRecorder.resumeRecording\(\)](#).

Success occurs if the stream was previously paused.

**Returns:**

true if successful.

---

## stopRecording

```
public boolean stopRecording( )
```

Request that stream recording stop. This shuts down saving of chunks

Used internally. Clients should call [ILiveStreamDvrRecorder.stopRecording\(\)](#).

Success only occurs if the stream is already in the recording state [isRecording\(\)](#).

**Returns:**

true if successful.



## isRecordingPaused

```
public boolean isRecordingPaused()
```

Is store currently paused while recording.

**Returns:**

true if recording is paused.

---

## isRecording

```
public boolean isRecording()
```

Is store currently recording. This will always be false for an edge.

**Returns:**

true if recording

---

## isLive

```
public boolean isLive()
```

Is store currently live For an origin, [isRecording\(\)](#) and [isLive\(\)](#) will typically return the same result. On an origin, [isRecording\(\)](#) will always be false. But [isLive\(\)](#) will reflect the status of the repeated stream.

**Returns:**

true if live

---

## storeChunks

```
public 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)
```

---

## getRecordedEntryByTime

```
public DvrManifestEntry getRecordedEntryByTime(int fragmentType,  
    long t)
```

---

## getDvrChunkByIndex

```
public DvrChunk getDvrChunkByIndex(int fragmentType,  
    int index)
```

---

## getDvrChunkAtTime

```
public DvrChunk getDvrChunkAtTime(int fragmentType,  
    long t)
```

---

---

### **getDvrChunkNearTime**

```
public DvrChunk getDvrChunkNearTime(int fragmentType,  
    long t,  
    long delta)
```

---

### **getClosestStartTime**

```
public long getClosestStartTime(int type,  
    long t)
```

## com.wowza.wms.dvr Interface IDvrStreamVersionHandler

All Known Implementing Classes:  
[DefaultDvrStreamVersionHandler](#)

public interface **IDvrStreamVersionHandler**  
 extends

Callback to manage (archival) versions of DVR stream stores

### Method Summary

<a href="#">IDvrStreamStore</a>	<a href="#">determineExistingStoreForPlaying</a> ( <a href="#">IDvrStreamManager</a> 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.
<a href="#">IDvrStreamStore</a>	<a href="#">determineExistingStoreForRecording</a> ( <a href="#">IDvrStreamManager</a> 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	<a href="#">getArchiveStrategy</a> ( <a href="#">IDvrStreamManager</a> dvrManager, String baseStreamName) Determine the archive strategy for a given set of streams.
boolean	<a href="#">handleArchivedStream</a> ( <a href="#">IDvrStreamManager</a> 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	<a href="#">shouldDeleteArchivedStream</a> ( <a href="#">IDvrStreamManager</a> dvrManager, <a href="#">IDvrStreamStore</a> store) Determine if a given stream store version should be deleted.
boolean	<a href="#">shouldLoadArchivedStream</a> ( <a href="#">IDvrStreamManager</a> 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	<a href="#">init</a> ( <a href="#">IApplicationInstance</a> appInstance, <a href="#">IMediaStream</a> stream, <a href="#">IDvrFileSystem</a> 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><a href="#">init</a>(<a href="#">IApplicationInstance</a> appInstance, <a href="#">IMediaStream</a> stream, <a href="#">IDvrFileSystem</a> 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	<a href="#">containsTime</a> (long startTime, <a href="#">IDvrConstants.DvrTimeScale</a> timeScale)
long	<a href="#">dvrToPt</a> (long dt)
long	<a href="#">dvrToUtc</a> (long dt)
java.util.List	<a href="#">getTimeMap</a> ()
java.util.List	<a href="#">getTimeMapEntries</a> ()
long	<a href="#">ptToDvr</a> (long pt)
long	<a href="#">toDvr</a> (long t, <a href="#">IDvrConstants.DvrTimeScale</a> timeScale)
long	<a href="#">utcToDvr</a> (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	<a href="#">authenticateHandler</a>
protected	<a href="#">authenticateHTTPProviderHandler</a>
protected	<a href="#">authenticationMethod</a>
protected	<a href="#">filters</a>
protected	<a href="#">properties</a>
protected	<a href="#">requestFilters</a>

## Constructor Summary

public	<a href="#">HTTPProvider2Base()</a>
--------	-------------------------------------

## Method Summary

boolean	<a href="#">canHandle</a> (String path) Return true if can handle the request
boolean	<a href="#">doHTTPAuthentication</a> ( <a href="#">IVHost</a> vhost, <a href="#">IHTTPRequest</a> req, <a href="#">IHTTPResponse</a> resp) Handle authentication request
String	<a href="#">getAuthenticationMethod</a> () Get the authentication method: digest, basic, none...
String	<a href="#">getPath</a> ( <a href="#">IHTTPRequest</a> req, boolean removeFilter) Get the request path
String	<a href="#">getRequestFilters</a> () Get the request filter
void	<a href="#">init</a> () Initialize the HTTPProvider
void	<a href="#">onBind</a> ( <a href="#">IVHost</a> vhost, <a href="#">HostPort</a> hostPort) Called when bind is called on port
void	<a href="#">onUnbind</a> ( <a href="#">IVHost</a> vhost, <a href="#">HostPort</a> hostPort) Called when unbind is called on port
void	<a href="#">setAuthenticationMethod</a> (String authenticationMethod) Set authentication method: digest, basic, none...
void	<a href="#">setProperties</a> ( <a href="#">WMSProperties</a> properties) Set properties
void	<a href="#">setRequestFilters</a> (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

## setProperties

```
public void setProperties(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	<a href="#">onBind</a> ( <a href="#">IVHost</a> vhost, <a href="#">HostPort</a> hostPort) Triggered after hostPort binds to socket
void	<a href="#">onHTTPRequest</a> ( <a href="#">IVHost</a> vhost, <a href="#">IHTTPRequest</a> req, <a href="#">IHTTPResponse</a> resp) Triggered for each HTTP request to the given hostPort that is not an RTMPT request.
void	<a href="#">onUnbind</a> ( <a href="#">IVHost</a> vhost, <a href="#">HostPort</a> hostPort) Triggered after hostPort unbinds
void	<a href="#">setProperties</a> ( <a href="#">WMSProperties</a> 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	<a href="#">canHandle</a> (String path)
String	<a href="#">getAuthenticationMethod</a> ()
String	<a href="#">getRequestFilters</a> ()
void	<a href="#">init</a> ()
void	<a href="#">setAuthenticationMethod</a> (String authenticationMethod)
void	<a href="#">setRequestFilters</a> (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	<a href="#"><u>getContentLength()</u></a> Get the content length of the body of the message
String	<a href="#"><u>getContentType()</u></a> Get the request content type
String	<a href="#"><u>getHeader</u></a> (String name) Get a HTTP header value such as 'Content-Length'
byte[]	<a href="#"><u>getHeaderBytes()</u></a> Returns the header as bytes
java.util.Map	<a href="#"><u>getHeaderMap()</u></a> Get a copy of the HTTP request header map
java.util.Set	<a href="#"><u>getHeaderNames()</u></a> Get a Set of the header names
java.io.InputStream	<a href="#"><u>getInputStream()</u></a> Get the body of the message as an input stream
int	<a href="#"><u>getIntHeader</u></a> (String name) Get a HTTP header value such as 'Content-Length' and return as int
java.util.Locale	<a href="#"><u>getLocale()</u></a> Get locale of request (Example: en-us)
String	<a href="#"><u>getMethod()</u></a> Get the method invocation method: GET, POST, HEAD
byte[]	<a href="#"><u>getMsgBytes()</u></a> Return the message bytes
String	<a href="#"><u>getParameter</u></a> (String name) Get a parameter value
java.util.Map	<a href="#"><u>getParameterMap()</u></a> Get the entire parameter Map
java.util.Set	<a href="#"><u>getParameterNames()</u></a> Get a Set of parameter names
String[]	<a href="#"><u>getParameterValues</u></a> (String name) Get a multi-value parameter as an array of String
String	<a href="#"><u>getPath()</u></a> Returns the HTTP path element of the request

String	<a href="#"><u>getProtocol()</u></a> Get the request protocol (example: HTTP/1.1)
String	<a href="#"><u>getQueryString()</u></a> Get the query string part of the url (everything after the ?)
String	<a href="#"><u>getRemoteAddr()</u></a> Get the remote ip address of the request
String	<a href="#"><u>getRemoteHost()</u></a> Get the remote host name (if known) if not return ip address
String	<a href="#"><u>getRequestURI()</u></a> Get the full request URI
String	<a href="#"><u>getRequestURL()</u></a> Get the request url (same as URI minus the query string)
String	<a href="#"><u>getScheme()</u></a> Get the request scheme (Example "http")
String	<a href="#"><u>getServerName()</u></a> Get the name of the server (Example: "Wowza Media Server Pro")
int	<a href="#"><u>getServerPort()</u></a> Get the port this request was received on
boolean	<a href="#"><u>isSecure()</u></a> Returns true is the request is protected by SSL
void	<a href="#"><u>parseBodyForParams()</u></a> 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	<a href="#"><u>parseBodyForParams(boolean doDecode)</u></a> 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	<a href="#"><code>containsHeader</code></a> (String name) Returns true if reponse header contains parameter name
String	<a href="#"><code>getHeader</code></a> (String name) Get header value
java.util.Map	<a href="#"><code>getHeaders</code></a> ( ) Get the current response headers as a map
int	<a href="#"><code>getHeaderSize</code></a> ( ) Get the size in bytes of the HTTP header
int	<a href="#"><code>getIntHeader</code></a> (String name) Get header value as int
java.io.OutputStream	<a href="#"><code>getOutputStream</code></a> ( ) Get the output stream for the reponse.
void	<a href="#"><code>removeHeader</code></a> (String name) Remove header value
void	<a href="#"><code>setHeader</code></a> (String name, String value) Set header value
void	<a href="#"><code>setIntHeader</code></a> (String name, int value) Set header value as int
void	<a href="#"><code>setResponseCode</code></a> (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)

(continued from last page)

Returns true if reponse header contains parameter name

**Parameters:**

name - header parameter name

**Returns:**

true if header contains value

---

## setHeader

```
public void setHeader(String name,  
                      String value)
```

Set header value

**Parameters:**

name - header parameter name

value - parameter value

---

## removeHeader

```
public void removeHeader(String name)
```

Remove header value

**Parameters:**

name

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

---

(continued from last page)

**Returns:**parameter value

---

**getIntHeader**

```
public int getIntHeader(String name)
```

Get header value as int

**Parameters:**

name - header parameter name

**Returns:**parameter value

---

**setResponseCode**

```
public void setResponseCode(int responseCode)
```

Set the HTTP response code

**Parameters:**responseCode - HTTP response code

---

**getHeaderSize**

```
public int getHeaderSize()
```

Get the size in bytes of the HTTP header

**Returns:**header size in bytes

---



---

Package

**com.wowza.wms.httpstreamer.cupertinostrea  
ming.httpstreamer**

## com.wowza.wms.httpstreamer.cupertinostreaming.httpstreamer Class HTTPStreamerSessionCupertino

java.lang.Object

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

└─

com.wowza.wms.httpstreamer.cupertinostreaming.httpstreamer.HTTPStreamerSessionCupertino

All Implemented Interfaces:

[IHTTPSessionNotify](#), [IHTTPStreamerSession](#)

public class **HTTPStreamerSessionCupertino**  
 extends HTTPStreamerSessionBase  
 implements [IHTTPStreamerSession](#), [IHTTPSessionNotify](#)

### Fields inherited from class com.wowza.wms.httpstreamer.model.HTTPStreamerSessionBase

acceptSession, appInstance, connectionHolder, cookieStr, DATEFORMAT, elapsedTime, fastDateFormat, fileInfoMap, firstCheck, httpHeaders, httpStreamerAdapter, ioPerformanceCounter, ipAddress, isActive, isHTTPOrigin, isPlayLogged, 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	<a href="#">HTTPStreamerSessionCupertino()</a>
--------	--

## Method Summary

void	<a href="#">clearLoggingValues()</a>
boolean	<a href="#">containsIndex</a> (String streamName)
static boolean	<a href="#">doesFileExist</a> ( <a href="#">IHTTPStreamerApplicationContext</a> appContext, String rawStreamName, String streamExt, String streamName, <a href="#">IHTTPStreamerSession</a> httpStreamerSession)

IHTTPStreamerCupertinoIndex	<a href="#">getIndex(IHTTPStreamerApplicationContext appContext, IHTTPStreamerSession httpStreamerSession, String rawStreamName, String streamExt, String streamName, long playStart, long playDuration)</a>
void	<a href="#">logLiveChunk</a> (LiveStreamPacketizerCupertinoChunk chunk)
void	<a href="#">logVODChunk</a> (LiveStreamPacketizerCupertinoChunk chunk)
void	<a href="#">notifyHTTPSessionCreate(IApplicationInstance appInstance, IHTTPStreamerSession httpStreamerSession)</a>
void	<a href="#">notifyHTTPSessionDestroy(IApplicationInstance appInstance, IHTTPStreamerSession httpStreamerSession)</a>
void	<a href="#">shutdown()</a>
void	<a href="#">updateLoggingValues()</a>

#### 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, isHTTPOrigin, isPlayLogged, isRedirectSession, isTimeout, isTimeoutSession, isValidated, isValidStreamDomainStr, lockRepeaterStreams, putFileInfo, putStreamNameParts, redirectSession, redirectSession, rejectSession, removeStreamDomainStr, setAcceptSession, setActive, setAppInstance, setCookieStr, setDvrSessionInfo, setHTTPOrigin, 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](#), [isHTTPOrigin](#), [isPlayLogged](#),  
[isRedirectSession](#), [isTimeout](#), [isTimeoutSession](#), [isValidated](#), [isValidStreamDomainStr](#),  
[lockRepeaterStreams](#), [putFileInfo](#), [putStreamNameParts](#), [redirectSession](#),  
[redirectSession](#), [rejectSession](#), [removeStreamDomainStr](#), [setAcceptSession](#), [setActive](#),  
[setAppInstance](#), [setCookieStr](#), [setDvrSessionInfo](#), [setHTTPOrigin](#),  
[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](#)

Methods inherited from interface [com.wowza.wms.httpstreamer.model.IHTTPSessionNotify](#)

[notifyHTTPSessionCreate](#), [notifyHTTPSessionDestroy](#)

## 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)
```

---

## notifyHTTPSessionCreate

```
public void notifyHTTPSessionCreate(IApplicationInstance appInstance,  
    IHTTPStreamerSession httpStreamerSession)
```

---

## notifyHTTPSessionDestroy

```
public void notifyHTTPSessionDestroy(IApplicationInstance appInstance,  
    IHTTPStreamerSession httpStreamerSession)
```

---

## 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	<a href="#">onCreate</a> (IHTTPStreamerCupertinoIndex fileIndex, <a href="#">IHTTPStreamerApplicationContext</a> appContext, <a href="#">IHTTPStreamerSession</a> httpStreamerSession, String rawStreamName, String streamExt, String streamName) Called when file index created
void	<a href="#">onDestroy</a> (IHTTPStreamerCupertinoIndex fileIndex) Called after file index is destroyed
void	<a href="#">onFillChunkEnd</a> (IHTTPStreamerCupertinoIndex fileIndex, IHTTPStreamerCupertinoIndexItem item, LiveStreamPacketizerCupertinoChunk chunk, boolean audioOnly) Called after each chunk is filled.
void	<a href="#">onFillChunkStart</a> (IHTTPStreamerCupertinoIndex fileIndex, IHTTPStreamerCupertinoIndexItem item, LiveStreamPacketizerCupertinoChunk chunk, boolean audioOnly) Called each time a chunk is filled.
void	<a href="#">onIndex</a> (IHTTPStreamerCupertinoIndex fileIndex, <a href="#">IHTTPStreamerApplicationContext</a> appContext, <a href="#">IHTTPStreamerSession</a> httpStreamerSession, String rawStreamName, String streamExt, String streamName) Called after file is indexed
void	<a href="#">onInit</a> (IHTTPStreamerCupertinoIndex fileIndex, <a href="#">IHTTPStreamerApplicationContext</a> appContext, <a href="#">IHTTPStreamerSession</a> httpStreamerSession, String rawStreamName, String streamExt, String streamName) Called after initialized
void	<a href="#">onOpen</a> (IHTTPStreamerCupertinoIndex fileIndex, <a href="#">IHTTPStreamerApplicationContext</a> appContext, <a href="#">IHTTPStreamerSession</a> 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	<a href="#">onFillChunkDataPacket</a> (IHTTPStreamerCupertinoIndex fileIndex, IHTTPStreamerCupertinoIndexItem item, LiveStreamPacketizerCupertinoChunk chunk, boolean audioOnly, <a href="#">AMFPacket</a> packet, <a href="#">ID3Frames</a> 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 IHTTPSessionFactory

---

public interface **IHTTPSessionFactory**  
extends

---

### Method Summary

<a href="#">IHTTPStreamerSession</a>	<a href="#">createHTTPSessionInstance()</a>
--------------------------------------	---

---

### Methods

#### **createHTTPSessionInstance**

public [IHTTPStreamerSession](#) **createHTTPSessionInstance()**

## com.wowza.wms.httpstreamer.model Interface IHTTPSessionNotify

All Known Implementing Classes:

[HTTPStreamerSessionSmoothStreamer](#), [HTTPStreamerSessionSanJose](#), [HTTPStreamerSessionCupertino](#)

public interface **IHTTPSessionNotify**  
extends

### Method Summary

void	<a href="#">notifyHTTPSessionCreate(IApplicationInstance</a> appInstance, <a href="#">IHTTPStreamerSession</a> httpStreamerSession)
void	<a href="#">notifyHTTPSessionDestroy(IApplicationInstance</a> appInstance, <a href="#">IHTTPStreamerSession</a> httpStreamerSession)

### Methods

#### notifyHTTPSessionCreate

```
public void notifyHTTPSessionCreate(IApplicationInstance appInstance,
IHTTPStreamerSession httpStreamerSession)
```

#### notifyHTTPSessionDestroy

```
public void notifyHTTPSessionDestroy(IApplicationInstance appInstance,
IHTTPStreamerSession httpStreamerSession)
```

## com.wowza.wms.httpstreamer.model Interface IHTTPStreamerAdapter

public interface **IHTTPStreamerAdapter**  
extends

IHTTPStreamerAdapter: HTTP streaming adapter interface

### Method Summary

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

void

[shutdownSession](#)([IApplicationInstance](#) appInstance, [IHTTPStreamerSession](#) session)

Called when an HTTP streaming session is shutdown

## Methods

### canHandle

```
public boolean canHandle(String path)
```

Return true if can handle request

**Parameters:**

path - request path

**Returns:**

true if can handle request

### service

```
public void service(org.apache.mina.common.Session session,
    RtmpRequestMessage req,
    RtmpResponseMessage resp)
```

Called to service each request

**Parameters:**

session - io session

req - request

resp - response

### getProperties

```
public WMSPProperties getProperties()
```

Get properties

**Returns:**

properties

### setProperties

```
public void setProperties(WMSPProperties properties)
```

Set properties

**Parameters:**

properties - properties

### getHTTPStreamerItem

```
public HTTPStreamerItem getHTTPStreamerItem()
```

Get the HTTP streamer item associated with this adapter

**Returns:**

(continued from last page)

HTTP streamer item

---

## setHTTPStreamerItem

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

Set the HTTP streamer item associated with this adapter

**Parameters:**

httpStreamerItem - HTTP streamer item

---

## getVHost

```
public IVHost getVHost()
```

Get the vhost associated with this adapter

**Returns:**

vhost

---

## setVHost

```
public void setVHost(IVHost vhost)
```

Set the vhost associated with this adapter

**Parameters:**

vhost - vhost

---

## init

```
public void init()
```

Initialize the HTTP streaming adapter

---

## shutdownSession

```
public void shutdownSession(IApplicationInstance appInstance,  
    IHTTPStreamerSession session)
```

Called when an HTTP streaming session is shutdown

**Parameters:**

session - HTTP streaming session

---

## getIdleFrequency

```
public int getIdleFrequency()
```

Get the idle frequency (milliseconds) for HTTP requests. This is how often the session is called back while active.

**Returns:**

idle frequency (milliseconds)

---

## getAdapterName

```
public String getAdapterName()
```

(continued from last page)

Get the name of this adapter

**Returns:**

name of this adapter

---

**getID**

```
public String getID()
```

Get the id of this adapter

**Returns:**

id of this adapter

---

**setID**

```
public void setID(String id)
```

Set the id of this adapter

**Parameters:**

id - id of this adapter

---

**getSessionProtocol**

```
public int getSessionProtocol()
```

Set the adapter session protocol. See IHTTPStreamerSession.SESSIONPROTOCOL\_\*

**Returns:**

adapter session protocol

---

**setSessionProtocol**

```
public void setSessionProtocol(int sessionProtocol)
```

Get the adapter session protocol. See IHTTPStreamerSession.SESSIONPROTOCOL\_\*

**Parameters:**

sessionProtocol - adapter session protocol

---



## com.wowza.wms.httpstreamer.model Interface IHTTPStreamerApplicationContext

public interface **IHTTPStreamerApplicationContext**  
extends

IHTTPStreamerApplicationContext: HTTP streamer application context interface. Used by HTTP streaming adapter to store per-application information.

### Field Summary

public static final	<a href="#"><u>HTTPORIGINMODE_AUTO</u></a> Value: <b>1</b>
public static final	<a href="#"><u>HTTPORIGINMODE_OFF</u></a> Value: <b>0</b>
public static final	<a href="#"><u>HTTPORIGINMODE_ON</u></a> Value: <b>2</b>

### Method Summary

<a href="#"><u>IApplicationInstance</u></a>	<a href="#"><u>getAppInstance()</u></a> Get application instance
int	<a href="#"><u>getHTTPOriginMode()</u></a> Get current HTTP origin mode.
<a href="#"><u>WMSProperties</u></a>	<a href="#"><u>getProperties()</u></a> Get properties
<a href="#"><u>MediaCasterItem</u></a>	<a href="#"><u>getRepeaterMediaCasterDef()</u></a> Get the live repeater media caster definition
String	<a href="#"><u>getStreamTypeStr()</u></a> Get stream type
<a href="#"><u>IVHost</u></a>	<a href="#"><u>getVHost()</u></a> Get vhost
void	<a href="#"><u>init(IApplicationInstance appInstance, HTTPStreamerItem httpStreamerItem)</u></a> Initialize context
boolean	<a href="#"><u>isHTTPOriginOn()</u></a> Returns true if HTTP origin mode is set to ON.
boolean	<a href="#"><u>isStreamDomainProtectionActive()</u></a> Return true if stream domain protection is on
void	<a href="#"><u>setHTTPOriginMode(int httpOriginMode)</u></a> Set current HTTP origin mode.

void	<a href="#">setRepeaterMediaCasterDef</a> ( <a href="#">MediaCasterItem</a> repeaterMediaCasterDef) Set the live repeater media caster definition
void	<a href="#">setStreamDomainProtectionActive</a> (boolean streamDomainProtectionActive) Turn on/off stream domain protection
void	<a href="#">setStreamTypeStr</a> (String streamTypeStr) Set stream type

Fields

HTTPORIGINMODE\_OFF

```
public static final int HTTPORIGINMODE_OFF
```

Constant value: 0

HTTPORIGINMODE\_AUTO

```
public static final int HTTPORIGINMODE_AUTO
```

Constant value: 1

HTTPORIGINMODE\_ON

```
public static final int HTTPORIGINMODE_ON
```

Constant value: 2

Methods

init

```
public void init(IApplicationInstance appInstance,  
                HTTPStreamerItem httpStreamerItem)
```

Initialize context

Parameters:

- appInstance - application instance
- httpStreamerItem - HTTP streamer

getProperties

```
public WMSProperties getProperties()
```

Get properties

Returns:

properties

(continued from last page)

## 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)
```

Set stream type

**Parameters:**

streamTypeStr - stream type

---

## isStreamDomainProtectionActive

```
public boolean isStreamDomainProtectionActive()
```

---

(continued from last page)

Return true if stream domain protection is on

**Returns:**

true if stream domain protection is on

---

## setStreamDomainProtectionActive

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

Turn on/off stream domain protection

**Parameters:**

streamDomainProtectionActive - true if stream domain protection is on

---

## getHTTPOriginMode

```
public int getHTTPOriginMode()
```

Get current HTTP origin mode. See HTTPORIGINMODE\_ (auto is currently not supported)

**Returns:**

HTTP origin mode

---

## setHTTPOriginMode

```
public void setHTTPOriginMode(int httpOriginMode)
```

Set current HTTP origin mode. See HTTPORIGINMODE\_ (auto is currently not supported)

**Parameters:**

httpOriginMode - HTTP origin mode

---

## isHTTPOriginOn

```
public boolean isHTTPOriginOn()
```

Returns true if HTTP origin mode is set to ON.

**Returns:**

true if HTTP origin mode is set to ON.

---

# com.wowza.wms.httpstreamer.model

## Interface IHTTPStreamerRepeater

public interface IHTTPStreamerRepeater  
extends

IHTTPStreamerRepeater: stream switching constants

Field Summary	
public static final	<a href="#">MSG_STARTSTREAM</a> Value: <b>startStream</b>
public static final	<a href="#">MSG_SWITCHSTREAM</a> Value: <b>switchStream</b>

### 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	<a href="#">SESSIONPROTOCOL_COUNT</a> Value: <b>6</b>
public static final	<a href="#">SESSIONPROTOCOL_CUPERTINOSTREAMING</a> Value: <b>1</b>
public static final	<a href="#">SESSIONPROTOCOL_DVRCHUNKSTREAMING</a> Value: <b>5</b>
public static final	<a href="#">SESSIONPROTOCOL_MPEGDASHSTREAMING</a> Value: <b>4</b>
public static final	<a href="#">SESSIONPROTOCOL_SANJOSESTREAMING</a> Value: <b>2</b>
public static final	<a href="#">SESSIONPROTOCOL_SMOOTHSTREAMING</a> Value: <b>0</b>
public static final	<a href="#">SESSIONPROTOCOL_UNKNOWN</a> Value: <b>-1</b>
public static final	<a href="#">SESSIONPROTOCOL_WEBMSTREAMING</a> Value: <b>3</b>
public static final	<a href="#">SESSIONTYPE_LIVE</a> Value: <b>1</b>
public static final	<a href="#">SESSIONTYPE_LIVEDVR</a> Value: <b>3</b>
public static final	<a href="#">SESSIONTYPE_UNKNOWN</a> Value: <b>0</b>
public static final	<a href="#">SESSIONTYPE_VOD</a> Value: <b>2</b>

## Method Summary

void	<a href="#"><u>acceptSession()</u></a> Accept the HTTP session.
void	<a href="#"><u>addIOPerformance()</u></a> <a href="#"><u>IOPerformanceCounter</u></a> totalIOPerformanceResult) Internal user, keep track of IO performance
void	<a href="#"><u>addIOPerformance2()</u></a> <a href="#"><u>IOPerformanceCounter</u></a> totalIOPerformanceResult) Internal user, keep track of IO performance
void	<a href="#"><u>addStreamDomainStr()</u></a> (String streamDomainStr) Internal user, add stream name
void	<a href="#"><u>addStreamDomainStrs()</u></a> (java.util.List streamNames) Internal user, add stream names
void	<a href="#"><u>addUserHTTPHeaders()</u></a> ( <a href="#"><u>IHTTPResponse</u></a> resp) Internal
boolean	<a href="#"><u>checkAndSetPlayLogged()</u></a> ( If play has not been logged return false, else returns true, sets play has been logged
void	<a href="#"><u>clearLoggingValues()</u></a> ( Internal user, clear logging values
boolean	<a href="#"><u>containsStreamDomainStr()</u></a> (String streamDomainStr) Internal user, test stream name
boolean	<a href="#"><u>containsStreamNameParts()</u></a> (String streamName) Return true if stream name in stream name parts
void	<a href="#"><u>doSessionRedirect()</u></a> ( <a href="#"><u>IHTTPResponse</u></a> resp) Internal
void	<a href="#"><u>extractHTTPRequestInfo()</u></a> ( <a href="#"><u>IHTTPRequest</u></a> req) Extract information from HTTP request
<a href="#"><u>IApplicationInstance</u></a>	<a href="#"><u>getAppInstance()</u></a> ( Get the application instance associated with this HTTP session
ConnectionHolder	<a href="#"><u>getConnectionHolder()</u></a> ( Connection holder for this session
String	<a href="#"><u>getCookieStr()</u></a> ( Get cookie string
DvrSessionInfo	<a href="#"><u>getDvrSessionInfo()</u></a> ( This information is used to manage the connection to the DVR store.
<a href="#"><u>ElapsedTimer</u></a>	<a href="#"><u>getElapsedTime()</u></a> ( Get the elapsed timer to see how long this session has been running
HTTPStreamerFileInfo	<a href="#"><u>getFileInfo()</u></a> (String streamName) Get the file information if video on demand streaming
String	<a href="#"><u>getHTTPHeader()</u></a> (String name) Get a HTTP header value such as 'Content-Length'

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



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

boolean	<a href="#"><u>isPlayLogged</u></a> ( ) true, if play has been logged
boolean	<a href="#"><u>isRedirectSession</u></a> ( ) Is session redirect
boolean	<a href="#"><u>isTimeout</u></a> (long timecode) Return true if this session is timed out.
boolean	<a href="#"><u>isTimeoutSession</u></a> ( ) Get is session timeout.
boolean	<a href="#"><u>isValidated</u></a> ( ) Has this session been validated
boolean	<a href="#"><u>isValidStreamDomainStr</u></a> (String streamDomainStr) Internal user, is stream name valid for HTTP session
void	<a href="#"><u>lockRepeaterStreams</u></a> (java.util.List streamNames, String liveStreamPacketizer, String liveStreamRepeater, String streamTypeStr) Internal user, lock in reapter streams
void	<a href="#"><u>putFileInfo</u></a> (String streamName, HTTPStreamerFileInfo fileInfo) Set the file information
void	<a href="#"><u>putStreamNameParts</u></a> (String streamName, HTTPStreamerStreamNameParts streamNameParts) Add stream name to stream name parts
void	<a href="#"><u>redirectSession</u></a> (String redirectSessionURL) Redirect session
void	<a href="#"><u>redirectSession</u></a> (String redirectSessionURL, int redirectSessionCode) Redirect session
void	<a href="#"><u>rejectSession</u></a> ( ) Reject this HTTP session.
void	<a href="#"><u>removeStreamDomainStr</u></a> (String streamDomainStr) Internal user, remove stream name
void	<a href="#"><u>setAcceptSession</u></a> (boolean acceptSession) Set to false to reject session
void	<a href="#"><u>setActive</u></a> (boolean isActive) Set session active
void	<a href="#"><u>setAppInstance</u></a> ( <a href="#"><u>IApplicationInstance</u></a> appInstance) Set the application instance associated with this HTTP session
void	<a href="#"><u>setCookieStr</u></a> (String cookieStr) Set cookie string
void	<a href="#"><u>setDvrSessionInfo</u></a> (DvrSessionInfo dvr) This information is used to manage the connection to the DVR store.
void	<a href="#"><u>setHTTPOrigin</u></a> (boolean isHTTPOrigin) Set to true if HTTP origin session

void	<a href="#"><u>setHTTPStreamerAdapter</u></a> ( <a href="#"><u>IHTTPStreamerAdapter</u></a> httpStreamerAdapter) Set the HTTP streaming adapter associated with this HTTP session
void	<a href="#"><u>setIpAddress</u></a> (String ipAddress) Set the IP address
void	<a href="#"><u>setLiveStreamingPacketizer</u></a> (String liveStreamingPacketizer) Set the live stream packetizer name
void	<a href="#"><u>setPlayDuration</u></a> (long playDuration) Set the play duration (milliseconds) for video on demand playback.
void	<a href="#"><u>setPlayLogged</u></a> (boolean isPlayLogged) true, if play has been logged
void	<a href="#"><u>setPlayStart</u></a> (long playStart) Set the play start time offset (milliseconds) for video on demand playback.
void	<a href="#"><u>setQueryStr</u></a> (String queryStr) Set query string
void	<a href="#"><u>setRedirectSession</u></a> (boolean redirectSession) Set session redirect
void	<a href="#"><u>setRedirectSessionBody</u></a> (byte[] redirectSessionBody) Set redirect session body
void	<a href="#"><u>setRedirectSessionCode</u></a> (int redirectSessionCode) Set session redirect HTTP response code (default 302)
void	<a href="#"><u>setRedirectSessionContentType</u></a> (String redirectSessionContentType) Set redirect session HTTP Content-Type
void	<a href="#"><u>setRedirectSessionURL</u></a> (String redirectSessionURL) Set redirect session URL
void	<a href="#"><u>setReferrer</u></a> (String referrer) Set referrer
void	<a href="#"><u>setServerIp</u></a> (String serverIp) Set server IP address
void	<a href="#"><u>setServerPort</u></a> (int serverPort) Set server port
void	<a href="#"><u>setSessionId</u></a> (String sessionId) Set session id
void	<a href="#"><u>setSessionProtocol</u></a> (int sessionProtocol) Set protocol, see SESSIONPROTOCOL_*
void	<a href="#"><u>setSessionTimeout</u></a> (int sessionTimeout) Set the session timeout for this session (milliseconds)
void	<a href="#"><u>setSessionType</u></a> (int sessionType) Set session type: see SESSIONTYPE_*
void	<a href="#"><u>setStream</u></a> ( <a href="#"><u>IMediaStream</u></a> stream) Set the IMediaStream associated with this HTTP session

void	<a href="#"><code>setStreamExt</code></a> (String streamExt) Set stream extension
void	<a href="#"><code>setStreamName</code></a> (String streamName) Set stream name
void	<a href="#"><code>setStreamPosition</code></a> (long streamPosition) Set stream position (will not cause seek)
void	<a href="#"><code>setTimeoutSession</code></a> (boolean timeoutSession) Set is session timeout.
void	<a href="#"><code>setUri</code></a> (String uri) Set the URI associated with initial request
void	<a href="#"><code>setUserAgent</code></a> (String userAgent) Set user agent
void	<a href="#"><code>setUserHTTPHeader</code></a> (String name, String value) Set user HTTP header.
void	<a href="#"><code>setUserQueryStr</code></a> (String userQueryStr) This query string will be added to URLs used in HTTP streaming
void	<a href="#"><code>setVHost</code></a> ( <a href="#"><code>IVHost</code></a> vhost) Set vhost
void	<a href="#"><code>shutdown</code></a> ( ) Called then the HTTP session is shutting down
void	<a href="#"><code>touch</code></a> (long timecode) Touch this session to keep it active.
void	<a href="#"><code>updateLoggingValues</code></a> ( ) Internal user, update logging values
String	<a href="#"><code>validStreamDomainToString</code></a> ( ) Return the valid domain strings as a string

## Fields

### SESSIONTYPE\_UNKNOWN

```
public static final int SESSIONTYPE_UNKNOWN
```

Constant value: **0**

### SESSIONTYPE\_LIVE

```
public static final int SESSIONTYPE_LIVE
```

Constant value: **1**

(continued from last page)

---

## SESSIONTYPE\_VOD

```
public static final int SESSIONTYPE_VOD
```

Constant value: **2**

---

## SESSIONTYPE\_LIVEDVR

```
public static final int SESSIONTYPE_LIVEDVR
```

Constant value: **3**

---

## SESSIONPROTOCOL\_UNKNOWN

```
public static final int SESSIONPROTOCOL_UNKNOWN
```

Constant value: **-1**

---

## SESSIONPROTOCOL\_SMOOTHSTREAMING

```
public static final int SESSIONPROTOCOL_SMOOTHSTREAMING
```

Constant value: **0**

---

## SESSIONPROTOCOL\_CUPERTINOSTREAMING

```
public static final int SESSIONPROTOCOL_CUPERTINOSTREAMING
```

Constant value: **1**

---

## SESSIONPROTOCOL\_SANJOSESTREAMING

```
public static final int SESSIONPROTOCOL_SANJOSESTREAMING
```

Constant value: **2**

---

## SESSIONPROTOCOL\_WEBMSTREAMING

```
public static final int SESSIONPROTOCOL_WEBMSTREAMING
```

Constant value: **3**

---

## SESSIONPROTOCOL\_MPEGDASHSTREAMING

```
public static final int SESSIONPROTOCOL_MPEGDASHSTREAMING
```

Constant value: **4**

---

## SESSIONPROTOCOL\_DVRCHUNKSTREAMING

```
public static final int SESSIONPROTOCOL_DVRCHUNKSTREAMING
```

---

(continued from last page)

Constant value: **5**

---

## SESSIONPROTOCOL\_COUNT

```
public static final int SESSIONPROTOCOL_COUNT
```

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();

(continued from last page)

---

## isTimeout

```
public boolean isTimeout(long timecode)
```

Return true if this session is timed out.

**Parameters:**

timecode - last touch - System.currentTimeMillis();

**Returns:**

true, if timed out

---

## getLock

```
public Object getLock()
```

Get the synchronization lock for this HTTP session

**Returns:**

synchronization lock

---

## shutdown

```
public void shutdown()
```

Called then the HTTP session is shutting down

---

## isActive

```
public boolean isActive()
```

Is this session active, false after shutdown

**Returns:**

true if session is active

---

## setActive

```
public void setActive(boolean isActive)
```

Set session active

**Parameters:**

isActive - session active

---

## getStream

```
public IMediaStream getStream()
```

Get the IMediaStream associated with this HTTP session

**Returns:**

IMediaStream associated with this HTTP session

---

## setStream

```
public void setStream(IMediaStream stream)
```

---

(continued from last page)

Set the IMediaStream associated with this HTTP session

**Parameters:**

stream - IMediaStream associated with this HTTP session

---

**isTimeoutSession**

```
public boolean isTimeoutSession()
```

Get is session timeout.

**Returns:**

true if this session is timed out

---

**setTimeoutSession**

```
public void setTimeoutSession(boolean timeoutSession)
```

Set is session timeout.

**Parameters:**

timeoutSession - true if this session is timed out

---

**getSessionTimeout**

```
public int getSessionTimeout()
```

Get the session timeout for this session (milliseconds)

**Returns:**

session timeout for this session (milliseconds)

---

**setSessionTimeout**

```
public void setSessionTimeout(int sessionTimeout)
```

Set the session timeout for this session (milliseconds)

**Parameters:**

sessionTimeout - session timeout for this session (milliseconds)

---

**isValidated**

```
public boolean isValidated()
```

Has this session been validated

**Returns:**

true, if validated

---

**checkAndSetPlayLogged**

```
public boolean checkAndSetPlayLogged()
```

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)
```

---

(continued from last page)

Set the HTTP streaming adapter associated with this HTTP session

**Parameters:**

httpStreamerAdapter - HTTP streaming adapter

---

## getAppInstance

```
public IApplicationInstance getAppInstance()
```

Get the application instance associated with this HTTP session

**Returns:**

application instance associated with this HTTP session

---

## setAppInstance

```
public void setAppInstance(IApplicationInstance appInstance)
```

Set the application instance associated with this HTTP session

**Parameters:**

appInstance - application instance associated with this HTTP session

---

## getSessionType

```
public int getSessionType()
```

Get session type: see SESSIONTYPE\_\*

**Returns:**

session type: see SESSIONTYPE\_\*

---

## setSessionType

```
public void setSessionType(int sessionType)
```

Set session type: see SESSIONTYPE\_\*

**Parameters:**

sessionType - session type: see SESSIONTYPE\_\*

---

## getLiveStreamingPacketizer

```
public String getLiveStreamingPacketizer()
```

Get the live stream packetizer name

**Returns:**

live stream packetizer name

---

## setLiveStreamingPacketizer

```
public void setLiveStreamingPacketizer(String liveStreamingPacketizer)
```

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)
```

---

(continued from last page)

Set the URI associated with initial request

**Parameters:**

uri - URI associated with initial request

---

## getReferrer

```
public String getReferrer()
```

Get referrer

**Returns:**

referrer

---

## setReferrer

```
public void setReferrer(String referrer)
```

Set referrer

**Parameters:**

referrer - referrer

---

## getQueryStr

```
public String getQueryStr()
```

Get query string

**Returns:**

query string

---

## setQueryStr

```
public void setQueryStr(String queryStr)
```

Set query string

**Parameters:**

queryStr - query string

---

## lockRepeaterStreams

```
public void lockRepeaterStreams(java.util.List streamNames,  
    String liveStreamPacketizer,  
    String liveStreamRepeater,  
    String streamTypeStr)
```

Internal user, lock in reapter streams

**Parameters:**

streamNames - list of stream names

liveStreamPacketizer - live stream packetizer name

liveStreamRepeater - live repeater name

streamTypeStr - stream type

(continued from last page)

---

## 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

---

(continued from last page)

## setStreamName

```
public void setStreamName(String streamName)
```

Set stream name

**Parameters:**

streamName - stream name

---

## getStreamExt

```
public String getStreamExt()
```

Get stream extension

**Returns:**

stream extension

---

## setStreamExt

```
public void setStreamExt(String streamExt)
```

Set stream extension

**Parameters:**

streamExt - stream extension

---

## getStreamNameParts

```
public HTTPStreamerStreamNameParts getStreamNameParts(String streamName)
```

Break the stream name into parts

**Parameters:**

streamName - stream name

**Returns:**

stream name parts

---

## containsStreamNameParts

```
public boolean containsStreamNameParts(String streamName)
```

Return true if stream name in stream name parts

**Parameters:**

streamName - stream name

**Returns:**

true if stream name in stream name parts

---

## putStreamNameParts

```
public void putStreamNameParts(String streamName,  
    HTTPStreamerStreamNameParts streamNameParts)
```

Add stream name to stream name parts

**Parameters:**

---

(continued from last page)

streamName - stream name

streamNameParts - stream name parts

---

## getStreamPosition

```
public long getStreamPosition()
```

Get stream position

**Returns:**

stream position

---

## setStreamPosition

```
public void setStreamPosition(long streamPosition)
```

Set stream position (will not cause seek)

**Parameters:**

streamPosition - stream position

---

## getIOPerformanceCounter

```
public IOPerformanceCounter getIOPerformanceCounter()
```

Get IO performance counter

**Returns:**

IO performance counter

---

## getFileInfo

```
public HTTPStreamerFileInfo getFileInfo(String streamName)
```

Get the file information if video on demand streaming

**Parameters:**

streamName - stream name

**Returns:**

file information

---

## putFileInfo

```
public void putFileInfo(String streamName,  
    HTTPStreamerFileInfo fileInfo)
```

Set the file information

**Parameters:**

streamName - stream name

fileInfo - file information

---

## isFileInfo

```
public boolean isFileInfo(String streamName)
```

Return true if the is file information for a given stream name



---

(continued from last page)

**Parameters:**

streamName - stream name

**Returns:**

true if the is file information for a given stream name

---

## isValidStreamDomainStr

```
public boolean isValidStreamDomainStr(String streamDomainStr)
```

Internal user, is stream name valid for HTTP session

**Parameters:**

streamDomainStr - stream name

**Returns:**

true if valid

---

## validStreamDomainToString

```
public String validStreamDomainToString()
```

Return the valid domain strings as a string

**Returns:**

valid domain strings as a string

---

## containsStreamDomainStr

```
public boolean containsStreamDomainStr(String streamDomainStr)
```

Internal user, test stream name

**Parameters:**

streamDomainStr - stream name

**Returns:**

true if valid

---

## removeStreamDomainStr

```
public void removeStreamDomainStr(String streamDomainStr)
```

Internal user, remove stream name

**Parameters:**

streamDomainStr - stream name

---

## addStreamDomainStr

```
public void addStreamDomainStr(String streamDomainStr)
```

Internal user, add stream name

**Parameters:**

streamDomainStr - stream name

---

(continued from last page)

---

## addStreamDomainStrs

```
public void addStreamDomainStrs(java.util.List streamNames)
```

Internal user, add stream names

**Parameters:**

streamNames - stream names

---

## getElapsedTime

```
public ElapsedTimer getElapsedTime()
```

Get the elapsed timer to see how long this session has been running

**Returns:**

elapsed timer (milliseconds)

---

## getTimeRunning

```
public String getTimeRunning()
```

Get the time this session has been running (milliseconds)

**Returns:**

time this session has been running (milliseconds)

---

## getTimeRunningSeconds

```
public double getTimeRunningSeconds()
```

Get the time this session has been running (seconds)

**Returns:**

time this session has been running (seconds)

---

## getProperties

```
public WMSProperties getProperties()
```

Get the properties associated with this session

**Returns:**

properties

---

## getUserQueryStr

```
public String getUserQueryStr()
```

This query string will be added to URLs used in HTTP streaming

**Returns:**

user query string

---

## setUserQueryStr

```
public void setUserQueryStr(String userQueryStr)
```

---

(continued from last page)

This query string will be added to URLs used in HTTP streaming

**Parameters:**

userQueryStr - user query string

---

## setDvrSessionInfo

```
public void setDvrSessionInfo(DvrSessionInfo dvr)
```

This information is used to manage the connection to the DVR store.

**Parameters:**

dvr - The DVR session info.

---

## getDvrSessionInfo

```
public DvrSessionInfo getDvrSessionInfo()
```

This information is used to manage the connection to the DVR store.

**Returns:**

DVR session info.

---

## extractHTTPRequestInfo

```
public void extractHTTPRequestInfo(IHTTPRequest req)
```

Extract information from HTTP request

**Parameters:**

req - HTTP request

---

## getHTTPHeaderMap

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

Get a copy of the HTTP request header map

**Returns:**

copy of the HTTP request header map

---

## getHTTPHeader

```
public String getHTTPHeader(String name)
```

Get a HTTP header value such as 'Content-Length'

**Parameters:**

name - header name

**Returns:**

header value

---

## getHTTPIntHeader

```
public int getHTTPIntHeader(String name)
```

Get a HTTP header value such as 'Content-Length' and return as int

---

(continued from last page)

**Parameters:**

name - header name

**Returns:**

header value

---

## getHTTPHeaderNames

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

Get a Set of the header names

**Returns:**

Set of header names

---

## getPlayStart

```
public long getPlayStart()
```

Get the play start time offset (milliseconds) for video on demand playback.

**Returns:**

play start time offset (milliseconds)

---

## setPlayStart

```
public void setPlayStart(long playStart)
```

Set the play start time offset (milliseconds) for video on demand playback.

**Parameters:**

playStart - play start time offset (milliseconds)

---

## getPlayDuration

```
public long getPlayDuration()
```

Get the play duration (milliseconds) for video on demand playback. A values of -1 means play to end.

**Returns:**

play duration (milliseconds)

---

## setPlayDuration

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

Set the play duration (milliseconds) for video on demand playback. A values of -1 means play to end.

**Parameters:**

playDuration - play duration (milliseconds)

---

## setUserHTTPHeader

```
public void setUserHTTPHeader(String name,  
                               String value)
```

Set user HTTP header. This header value will be added to all HTTP responses

**Parameters:**

(continued from last page)

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

(continued from last page)

## setRedirectSessionCode

```
public void setRedirectSessionCode(int redirectSessionCode)
```

Set session redirect HTTP response code (default 302)

**Parameters:**

redirectSessionCode - session redirect HTTP response code

---

## getRedirectSessionURL

```
public String getRedirectSessionURL()
```

Get redirect session URL

**Returns:**

redirect session URL

---

## setRedirectSessionURL

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

Set redirect session URL

**Parameters:**

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

---

(continued from last page)

---

## setRedirectSessionBody

```
public void setRedirectSessionBody(byte[] redirectSessionBody)
```

Set redirect session body

**Parameters:**

redirectSessionBody - redirect session body

---

## getRedirectSessionContentType

```
public String getRedirectSessionContentType()
```

Get redirect session HTTP Content-Type

**Returns:**

redirect session HTTP Content-Type

---

## setRedirectSessionContentType

```
public void setRedirectSessionContentType(String redirectSessionContentType)
```

Set redirect session HTTP Content-Type

**Parameters:**

redirectSessionContentType

---

## getLastRequest

```
public long getLastRequest()
```

Get the last timestamp of the last Io request

**Returns:**

last timestamp of the last Io request (milliseconds)

---

## isHTTPOrigin

```
public boolean isHTTPOrigin()
```

Return true if session in an HTTP origin session

**Returns:**

true if session in an HTTP origin session

---

## setHTTPOrigin

```
public void setHTTPOrigin(boolean isHTTPOrigin)
```

Set to true if HTTP origin session

**Parameters:**

isHTTPOrigin - true if HTTP origin session

---

## com.wowza.wms.httpstreamer.model Interface IHTTPStreamerSessionNotify

public interface IHTTPStreamerSessionNotify  
extends

IHTTPStreamerSessionNotify: HTTP session create/destroy interface

### Method Summary

void	<a href="#">onHTTPStreamerSessionCreate</a> ( <a href="#">IHTTPStreamerSession</a> httpStreamerSession) Called when an HTTP streaming session is created
void	<a href="#">onHTTPStreamerSessionDestroy</a> ( <a href="#">IHTTPStreamerSession</a> 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	<a href="#"><code>addToChunk</code></a> ( <code>LiveStreamPacketizerPacketHolder holder</code> )
void	<a href="#"><code>endChunk</code></a> ( <code>long timecode</code> )
boolean	<a href="#"><code>isPacketizeAudio</code></a> ()
boolean	<a href="#"><code>isPacketizeData</code></a> ()
boolean	<a href="#"><code>isPacketizeVideo</code></a> ()
boolean	<a href="#"><code>isValidAudioCodec</code></a> ( <code>int codec</code> )
boolean	<a href="#"><code>isValidVideoCodec</code></a> ( <code>int codec</code> )
void	<a href="#"><code>resetStream</code></a> ()
void	<a href="#"><code>setCodecInfoAudio</code></a> ( <code>com.wowza.wms.media.model.MediaCodecInfoAudio codecInfoAudio</code> )
void	<a href="#"><code>setCodecInfoVideo</code></a> ( <code>com.wowza.wms.media.model.MediaCodecInfoVideo codecInfoVideo</code> )
void	<a href="#"><code>startChunk</code></a> ( <code>int streamMode, int videoCodecId, int audioCodecId, long timecode</code> )

### 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)
```

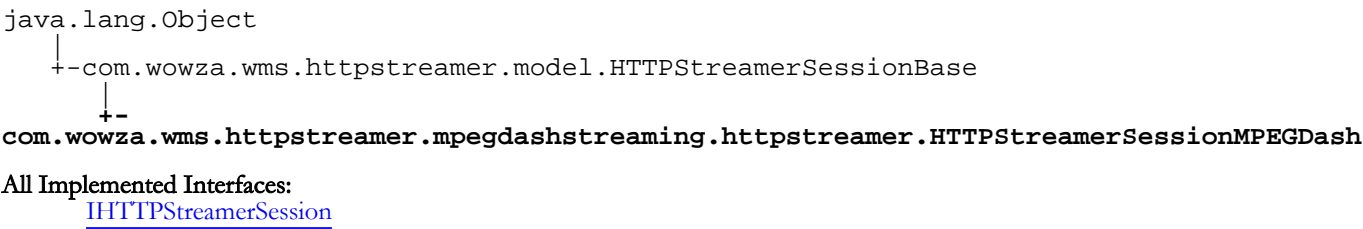
---

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, isHTTPOrigin, 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 <a href="#">com.wowza.wms.httpstreamer.model.IHTTPStreamerSession</a>
<a href="#">SESSIONPROTOCOL_COUNT</a> , <a href="#">SESSIONPROTOCOL_CUPERTINOSTREAMING</a> , <a href="#">SESSIONPROTOCOL_DVRCHUNKSTREAMING</a> , <a href="#">SESSIONPROTOCOL_MPEGDASHSTREAMING</a> , <a href="#">SESSIONPROTOCOL_SANJOSESTREAMING</a> , <a href="#">SESSIONPROTOCOL_SMOOTHSTREAMING</a> , <a href="#">SESSIONPROTOCOL_UNKNOWN</a> , <a href="#">SESSIONPROTOCOL_WEBMSTREAMING</a> , <a href="#">SESSIONTYPE_LIVE</a> , <a href="#">SESSIONTYPE_LIVEDVR</a> , <a href="#">SESSIONTYPE_UNKNOWN</a> , <a href="#">SESSIONTYPE_VOD</a>

Constructor Summary	
public	<a href="#">HTTPStreamerSessionMPGEDash()</a>

Method Summary	
void	<a href="#">logLiveChunk</a> (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,
isHTTPOrigin, isPlayLogged, isRedirectSession, isTimeout, isTimeoutSession,
isValidated, isValidStreamDomainStr, lockRepeaterStreams, putFileInfo,
putStreamNameParts, redirectSession, redirectSession, rejectSession,
removeStreamDomainStr, setAcceptSession, setActive, setAppInstance, setCookieStr,
setDvrSessionInfo, setHTTPOrigin, 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](#), [isHTTPOrigin](#), [isPlayLogged](#), [isRedirectSession](#), [isTimeout](#), [isTimeoutSession](#), [isValidated](#), [isValidStreamDomainStr](#), [lockRepeaterStreams](#), [putFileInfo](#), [putStreamNameParts](#), [redirectSession](#), [redirectSession](#), [rejectSession](#), [removeStreamDomainStr](#), [setAcceptSession](#), [setActive](#), [setAppInstance](#), [setCookieStr](#), [setDvrSessionInfo](#), [setHTTPOrigin](#), [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:

[IHTTPSessionNotify](#), [IHTTPStreamerSession](#)

public class **HTTPStreamerSessionSanJose**  
 extends HTTPStreamerSessionBase  
 implements [IHTTPStreamerSession](#), [IHTTPSessionNotify](#)

### Fields inherited from class com.wowza.wms.httpstreamer.model.HTTPStreamerSessionBase

acceptSession, appInstance, connectionHolder, cookieStr, DATEFORMAT, elapsedTime, fastDateFormat, fileInfoMap, firstCheck, httpHeaders, httpStreamerAdapter, ioPerformanceCounter, ipAddress, isActive, isHTTPOrigin, isPlayLogged, 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	<a href="#">HTTPStreamerSessionSanJose()</a>
--------	--

## Method Summary

void	<a href="#">addFirstABSTRequest</a> (String streamName)
void	<a href="#">clearLoggingValues</a> ()
boolean	<a href="#">containsIndex</a> (String streamName)
static IHTTPStreamerSanJoseIndex	<a href="#">createIndexLive</a> ( <a href="#">IHTTPStreamerApplicationContext</a> appContext, <a href="#">IHTTPStreamerSession</a> httpStreamerSession, String rawStreamName, String streamExt, String streamName)



IHTTPStreamerSanJoseIndex	<a href="#">getIndex(IHTTPStreamerApplicationContext appContext, IHTTPStreamerSession httpStreamerSession, String rawStreamName, String streamExt, String streamName, long playStart, long playDuration)</a>
IHTTPStreamerSanJoseIndex	<a href="#">getIndexLive(IHTTPStreamerApplicationContext appContext, IHTTPStreamerSession httpStreamerSession, String rawStreamName, String streamExt, String streamName)</a>
boolean	<a href="#">isFirstABSTRequest</a> (String streamName)
void	<a href="#">logLiveChunk</a> (LiveStreamPacketizerSanJoseChunk chunk)
void	<a href="#">logVODChunk</a> (LiveStreamPacketizerSanJoseChunk chunk)
void	<a href="#">notifyHTTPSessionCreate(IApplicationInstance appInstance, IHTTPStreamerSession httpStreamerSession)</a>
void	<a href="#">notifyHTTPSessionDestroy(IApplicationInstance appInstance, IHTTPStreamerSession httpStreamerSession)</a>
void	<a href="#">shutdown</a> ()
void	<a href="#">updateLoggingValues</a> ()

#### 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, isHTTPOrigin, isPlayLogged, isRedirectSession, isTimeout, isTimeoutSession, isValidated, isValidStreamDomainStr, lockRepeaterStreams, putFileInfo, putStreamNameParts, redirectSession, redirectSession, rejectSession, removeStreamDomainStr, setAcceptSession, setActive, setAppInstance, setCookieStr, setDvrSessionInfo, setHTTPOrigin, 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](#), [isHTTPOrigin](#), [isPlayLogged](#), [isRedirectSession](#), [isTimeout](#), [isTimeoutSession](#), [isValidated](#), [isValidStreamDomainStr](#), [lockRepeaterStreams](#), [putFileInfo](#), [putStreamNameParts](#), [redirectSession](#), [redirectSession](#), [rejectSession](#), [removeStreamDomainStr](#), [setAcceptSession](#), [setActive](#), [setAppInstance](#), [setCookieStr](#), [setDvrSessionInfo](#), [setHTTPOrigin](#), [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](#)

#### Methods inherited from interface [com.wowza.wms.httpstreamer.model.IHTTPSessionNotify](#)

[notifyHTTPSessionCreate](#), [notifyHTTPSessionDestroy](#)

## Constructors

### HTTPStreamerSessionSanJose

```
public HTTPStreamerSessionSanJose()
```

## Methods

### shutdown

```
public void shutdown()
```

(continued from last page)

---

## containsIndex

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

---

## 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)
```

---

(continued from last page)

## logVODChunk

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

---

## isFirstABSTRequest

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

---

## addFirstABSTRequest

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

---

## notifyHTTPSessionCreate

```
public void notifyHTTPSessionCreate(IApplicationInstance appInstance,  
    IHTTPStreamerSession httpStreamerSession)
```

---

## notifyHTTPSessionDestroy

```
public void notifyHTTPSessionDestroy(IApplicationInstance appInstance,  
    IHTTPStreamerSession httpStreamerSession)
```

---

---

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.HTTPStreamerSessionSmoothStream  
er

All Implemented Interfaces:

[IHTTPSessionNotify](#), [IHTTPStreamerSession](#)

```
public class HTTPStreamerSessionSmoothStreamer
extends HTTPStreamerSessionBase
implements IHTTPStreamerSession, IHTTPSessionNotify
```

### Fields inherited from class com.wowza.wms.httpstreamer.model.HTTPStreamerSessionBase

acceptSession, appInstance, connectionHolder, cookieStr, DATEFORMAT, elapsedTime, fastDateFormat, fileInfoMap, firstCheck, httpHeaders, httpStreamerAdapter, ioPerformanceCounter, ipAddress, isActive, isHTTPOrigin, isPlayLogged, 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	<a href="#">HTTPStreamerSessionSmoothStreamer()</a>
--------	---

## Method Summary

void	<a href="#">clearLoggingValues()</a>
------	--------------------------------------

boolean	<a href="#">containsIndex</a> (String streamName)
---------	---

IHTTPStreamerSmoothSt reamerIndex	<a href="#">getIndex</a> ( <a href="#">IHTTPStreamerSession</a> httpStreamerSession, <a href="#">IHTTPStreamerApplicationContext</a> appContext, String streamExt, String streamName, long playStart, long playDuration)
--------------------------------------	--

SmoothStreamingLivePlaylist	<a href="#">getLivePlaylist()</a>
void	<a href="#">logLiveFragment</a> (SmoothStreamerFragmentId fragmentId, PacketFragmentList fragmentData)
void	<a href="#">logVODFragment</a> (SmoothStreamerFragmentId fragmentId, PacketFragmentList fragmentData)
void	<a href="#">notifyHTTPSessionCreate</a> ( <a href="#">IApplicationInstance</a> appInstance, <a href="#">IHTTPStreamerSession</a> httpStreamerSession)
void	<a href="#">notifyHTTPSessionDestroy</a> ( <a href="#">IApplicationInstance</a> appInstance, <a href="#">IHTTPStreamerSession</a> httpStreamerSession)
void	<a href="#">setLivePlaylist</a> (SmoothStreamingLivePlaylist livePlaylist)
void	<a href="#">shutdown</a> ()
void	<a href="#">updateLoggingValues</a> ()

#### 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, isHTTPOrigin, isPlayLogged, isRedirectSession, isTimeout, isTimeoutSession, isValidated, isValidStreamDomainStr, lockRepeaterStreams, putFileInfo, putStreamNameParts, redirectSession, redirectSession, rejectSession, removeStreamDomainStr, setAcceptSession, setActive, setAppInstance, setCookieStr, setDvrSessionInfo, setHTTPOrigin, 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](#), [isHTTPOrigin](#), [isPlayLogged](#),  
[isRedirectSession](#), [isTimeout](#), [isTimeoutSession](#), [isValidated](#), [isValidStreamDomainStr](#),  
[lockRepeaterStreams](#), [putFileInfo](#), [putStreamNameParts](#), [redirectSession](#),  
[redirectSession](#), [rejectSession](#), [removeStreamDomainStr](#), [setAcceptSession](#), [setActive](#),  
[setAppInstance](#), [setCookieStr](#), [setDvrSessionInfo](#), [setHTTPOrigin](#),  
[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](#)

**Methods inherited from interface** [com.wowza.wms.httpstreamer.model.IHTTPSessionNotify](#)

[notifyHTTPSessionCreate](#), [notifyHTTPSessionDestroy](#)

## Constructors

### HTTPStreamerSessionSmoothStreamer

```
public HTTPStreamerSessionSmoothStreamer()
```

## Methods

### shutdown

```
public void shutdown()
```

### containsIndex

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



(continued from last page)

---

## getIndex

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

---

## updateLoggingValues

```
public void updateLoggingValues()
```

---

## clearLoggingValues

```
public void clearLoggingValues()
```

---

## getLivePlaylist

```
public SmoothStreamingLivePlaylist getLivePlaylist()
```

---

## setLivePlaylist

```
public void setLivePlaylist(SmoothStreamingLivePlaylist livePlaylist)
```

---

## logLiveFragment

```
public void logLiveFragment(SmoothStreamerFragmentId fragmentId,
PacketFragmentList fragmentData)
```

---

## logVODFragment

```
public void logVODFragment(SmoothStreamerFragmentId fragmentId,
PacketFragmentList fragmentData)
```

---

## notifyHTTPSessionCreate

```
public void notifyHTTPSessionCreate(IApplicationInstance appInstance,
IHTTPStreamerSession httpStreamerSession)
```

---

(continued from last page)

**notifyHTTPSessionDestroy**

```
public void notifyHTTPSessionDestroy(IApplicationInstance appInstance,  
    IHTTPStreamerSession httpStreamerSession)
```

---

Package

**com.wowza.wms.httpstreamer.webmstreami  
ng.httpstreamer**

## com.wowza.wms.httpstreamer.webmstreaming.httpstreamer Class HTTPStreamerSessionWebM

java.lang.Object

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

└-com.wowza.wms.httpstreamer.webmstreaming.httpstreamer.HTTPStreamerSessionWebM

All Implemented Interfaces:

[IHTTPStreamerSession](#)

public class **HTTPStreamerSessionWebM**  
extends HTTPStreamerSessionBase

### Fields inherited from class com.wowza.wms.httpstreamer.model.HTTPStreamerSessionBase

acceptSession, appInstance, connectionHolder, cookieStr, DATEFORMAT, elapsedTime, fastDateFormat, fileInfoMap, firstCheck, httpHeaders, httpStreamerAdapter, ioPerformanceCounter, ipAddress, isActive, isHTTPOrigin, isPlayLogged, 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,
isHTTPOrigin, isPlayLogged, isRedirectSession, isTimeout, isTimeoutSession,
isValidated, isValidStreamDomainStr, lockRepeaterStreams, putFileInfo,
putStreamNameParts, redirectSession, redirectSession, rejectSession,
removeStreamDomainStr, setAcceptSession, setActive, setAppInstance, setCookieStr,
setDvrSessionInfo, setHTTPOrigin, 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](#), [isHTTPOrigin](#), [isPlayLogged](#),  
[isRedirectSession](#), [isTimeout](#), [isTimeoutSession](#), [isValidated](#), [isValidStreamDomainStr](#),  
[lockRepeaterStreams](#), [putFileInfo](#), [putStreamNameParts](#), [redirectSession](#),  
[redirectSession](#), [rejectSession](#), [removeStreamDomainStr](#), [setAcceptSession](#), [setActive](#),  
[setAppInstance](#), [setCookieStr](#), [setDvrSessionInfo](#), [setHTTPOrigin](#),  
[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	<a href="#"><u>ABSOLUTE_FORMAT</u></a> ABSOLUTE string literal. Value: <b>ABSOLUTE</b>
public static final	<a href="#"><u>ABSOLUTE_TIME_PATTERN</u></a> SimpleTimePattern for ABSOLUTE. Value: <b>HH:mm:ss,SSS</b>
public static final	<a href="#"><u>APPLICATION_KEY</u></a> application string literal. Value: <b>application</b>
public static final	<a href="#"><u>CODES_HREF</u></a> Codes URL string literal. Value: <b>http://logging.apache.org/log4j/docs/codes.html</b>
public static final	<a href="#"><u>CONFIGURATOR_CLASS_KEY</u></a> log4j.configuratorClass string literal. Value: <b>log4j.configuratorClass</b>
public static final	<a href="#"><u>DATE_AND_TIME_FORMAT</u></a> DATE string literal. Value: <b>DATE</b>
public static final	<a href="#"><u>DATE_AND_TIME_PATTERN</u></a> SimpleTimePattern for DATE. Value: <b>dd MMM yyyy HH:mm:ss,SSS</b>
public static final	<a href="#"><u>DEFAULT_CONFIGURATION_FILE</u></a> The default property file name for automatic configuration. Value: <b>log4j.properties</b>
public static final	<a href="#"><u>DEFAULT_CONFIGURATION_KEY</u></a> log4j.configuration string literal. Value: <b>log4j.configuration</b>
public static final	<a href="#"><u>DEFAULT_REPOSITORY_NAME</u></a> The name of the default repository is "default" (without the quotes). Value: <b>default</b>
public static final	<a href="#"><u>DEFAULT_XML_CONFIGURATION_FILE</u></a> The default XML configuration file name for automatic configuration. Value: <b>log4j.xml</b>



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

## 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, <a href="#">IMediaStream</a> 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	<a href="#">FD_cs_bytes_inc</a> Value: <b>cs-bytes-inc</b>
public static final	<a href="#">FD_cs_stream_bytes_inc</a> Value: <b>cs-stream-bytes-inc</b>
public static final	<a href="#">FD_sc_bytes_inc</a> Value: <b>sc-bytes-inc</b>
public static final	<a href="#">FD_sc_stream_bytes_inc</a> Value: <b>sc-stream-bytes-inc</b>
public static final	<a href="#">FD_x_duration_inc</a> Value: <b>x-duration-inc</b>

### Constructor Summary



public	<a href="#">LogNotifyCalculateIncremental()</a>
--------	---

## Method Summary

void	<a href="#">onLog</a> (org.apache.log4j.Level level, String comment, <a href="#">IMediaStream</a> 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	<a href="#">DEFAULT_PORT</a> The default port number for the UDP packets, 9991. Value: <b>9991</b>
---------------------	--

## Fields inherited from class org.apache.log4j.AppenderSkeleton

closed, errorHandler, headFilter, layout, name, tailFilter, threshold

## Constructor Summary

public	<a href="#">UDPAppender</a> ()
public	<a href="#">UDPAppender</a> (java.net.InetAddress address, int port) Sends UDP packets to the address and port.
public	<a href="#">UDPAppender</a> (String host, int port) Sends UDP packets to the address and port.

## Method Summary

void	<a href="#">activateOptions</a> () Open the UDP sender for the <b>RemoteHost</b> and <b>Port</b> .
void	<a href="#">append</a> (org.apache.log4j.spi.LoggingEvent event)

void	<a href="#"><code>cleanUp()</code></a> Close the UDP Socket and release the underlying connector thread if it has been created
void	<a href="#"><code>close()</code></a> Close this appender.
String	<a href="#"><code>getApplication()</code></a> Returns value of the <b>App</b> option.
String	<a href="#"><code>getEncoding()</code></a> Returns value of the <b>Encoding</b> option.
int	<a href="#"><code>getPort()</code></a> Returns value of the <b>Port</b> option.
String	<a href="#"><code>getRemoteHost()</code></a> Returns value of the <b>RemoteHost</b> option.
boolean	<a href="#"><code>isActive()</code></a>
boolean	<a href="#"><code>requiresLayout()</code></a> The UDPAppender uses layouts.
void	<a href="#"><code>setApplication(String app)</code></a> The <b>App</b> option takes a string value which should be the name of the application getting logged.
void	<a href="#"><code>setEncoding(String encoding)</code></a> The <b>Encoding</b> option specifies how the bytes are encoded.
void	<a href="#"><code>setPort(int port)</code></a> The <b>Port</b> option takes a positive integer representing the port where UDP packets will be sent.
void	<a href="#"><code>setRemoteHost(String host)</code></a> The <b>RemoteHost</b> 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	<a href="#">WMSLogger</a> (String name, org.apache.log4j.Logger tmpLogger)
public	<a href="#">WMSLogger</a> (String name)

## Method Summary

void	<a href="#">debug</a> (String comment)
void	<a href="#">debug</a> (String comment, <a href="#">IMediaStream</a> stream)
void	<a href="#">debug</a> (String comment, <a href="#">IMediaStream</a> stream, String category, String event, int status, String context)
void	<a href="#">debug</a> (String comment, String category, String event)
void	<a href="#">debug</a> (String comment, String category, String event, int status, String context)
void	<a href="#">error</a> (String comment)
void	<a href="#">error</a> (String comment, <a href="#">IMediaStream</a> stream)
void	<a href="#">error</a> (String comment, <a href="#">IMediaStream</a> stream, String category, String event, int status, String context)
void	<a href="#">error</a> (String comment, String category, String event)



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

void	<a href="#">log</a> (org.apache.log4j.Level level, String comment, String category, String event)
void	<a href="#">warn</a> (String comment)
void	<a href="#">warn</a> (String comment, <a href="#">IMediaStream</a> stream)
void	<a href="#">warn</a> (String comment, <a href="#">IMediaStream</a> stream, String category, String event, int status, String context)
void	<a href="#">warn</a> (String comment, String category, String event)
void	<a href="#">warn</a> (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	<a href="#">LOGGERNAME_SERVER</a> Value: <b>_Server_</b>
---------------------	---

### Constructor Summary

public	<a href="#">WMSLoggerFactory()</a>
--------	------------------------------------

### Method Summary

<a href="#">WMSLogger</a>	<a href="#">forceNewLoggerInstance</a> (String name, org.apache.log4j.Logger tmpLogger)
static Object	<a href="#">getGlobalLogValue</a> (String key)
static <a href="#">WMSLoggerFactory</a>	<a href="#">getInstance</a> ()
static <a href="#">WMSLogger</a>	<a href="#">getLogger</a> (Class classObj)
static <a href="#">WMSLogger</a>	<a href="#">getLoggerObj</a> ( <a href="#">IApplication</a> application)
static <a href="#">WMSLogger</a>	<a href="#">getLoggerObj</a> ( <a href="#">IApplicationInstance</a> appInstance)
static <a href="#">WMSLogger</a>	<a href="#">getLoggerObj</a> ( <a href="#">IVHost</a> vhost)
<a href="#">WMSLogger</a>	<a href="#">getLoggerObj</a> (String name)
static void	<a href="#">initContextLogging</a> ( <a href="#">IApplication</a> application)
static void	<a href="#">initContextLogging</a> ( <a href="#">IApplicationInstance</a> appInstance)
static void	<a href="#">initContextLogging</a> ( <a href="#">IVHost</a> vhost)

static <a href="#">WMSLogger</a>	<a href="#">initializeLogging</a> (String loggingConfigURL)
static <a href="#">WMSLogger</a>	<a href="#">initializeLogging</a> (String loggingConfigURL, <a href="#">IVHost</a> vhost)
static boolean	<a href="#">isGlobalLogValueSet</a> (String key)
org.apache.log4j.Logger	<a href="#">makeNewLoggerInstance</a> (String name)
static void	<a href="#">putGlobalLogValue</a> (String key, Object obj)
static void	<a href="#">removeGlobalLogValue</a> (String key)
static void	<a href="#">removeGlobalLogValues</a> ()
static void	<a href="#">resetMDC</a> ()

#### 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	<a href="#">CAT_ALL</a>
public static final	<a href="#">CAT_application</a> Value: <b>application</b>
public static final	<a href="#">CAT_cupertino</a> Value: <b>cupertino</b>
public static final	<a href="#">CAT_dvrchunk</a> Value: <b>dvrchunk</b>
public static final	<a href="#">CAT_mpegdash</a> Value: <b>mpegdash</b>
public static final	<a href="#">CAT_rtsp</a> Value: <b>rtsp</b>
public static final	<a href="#">CAT_sanjose</a> Value: <b>sanjose</b>
public static final	<a href="#">CAT_server</a> Value: <b>server</b>
public static final	<a href="#">CAT_session</a> Value: <b>session</b>
public static final	<a href="#">CAT_smoothstreaming</a> Value: <b>smoothstreaming</b>
public static final	<a href="#">CAT_stream</a> Value: <b>stream</b>
public static final	<a href="#">CAT_transcoder</a> Value: <b>transcoder</b>

public static final	<a href="#">CAT_vhost</a> Value: <b>vhost</b>
public static final	<a href="#">CAT_webm</a> Value: <b>webm</b>
public static final	<a href="#">CTRL_playlist_node</a> Value: <b>CTRL-playlist-node</b>
public static final	<a href="#">EVT_ALL</a>
public static final	<a href="#">EVT_announce</a> Value: <b>announce</b>
public static final	<a href="#">EVT_app_start</a> Value: <b>app-start</b>
public static final	<a href="#">EVT_app_stop</a> Value: <b>app-stop</b>
public static final	<a href="#">EVT_comment</a> Value: <b>comment</b>
public static final	<a href="#">EVT_connect</a> Value: <b>connect</b>
public static final	<a href="#">EVT_connect_burst</a> Value: <b>connect-burst</b>
public static final	<a href="#">EVT_connect_pending</a> Value: <b>connect-pending</b>
public static final	<a href="#">EVT_create</a> Value: <b>create</b>
public static final	<a href="#">EVT_decoderaudiostart</a> Value: <b>decoder-audio-start</b>
public static final	<a href="#">EVT_decoderaudiostop</a> Value: <b>decoder-audio-stop</b>
public static final	<a href="#">EVT_decodervideostart</a> Value: <b>decoder-video-start</b>
public static final	<a href="#">EVT_decodervideostop</a> Value: <b>decoder-video-stop</b>

public static final	<a href="#">EVT_describe</a> Value: <b>describe</b>
public static final	<a href="#">EVT_destroy</a> Value: <b>destroy</b>
public static final	<a href="#">EVT_disconnect</a> Value: <b>disconnect</b>
public static final	<a href="#">EVT_encoderaudiostart</a> Value: <b>encoder-audio-start</b>
public static final	<a href="#">EVT_encoderaudiostop</a> Value: <b>encoder-audio-stop</b>
public static final	<a href="#">EVT_encodervideostart</a> Value: <b>encoder-video-start</b>
public static final	<a href="#">EVT_encodervideostop</a> Value: <b>encoder-video-stop</b>
public static final	<a href="#">EVT_pause</a> Value: <b>pause</b>
public static final	<a href="#">EVT_play</a> Value: <b>play</b>
public static final	<a href="#">EVT_publish</a> Value: <b>publish</b>
public static final	<a href="#">EVT_record</a> Value: <b>record</b>
public static final	<a href="#">EVT_recordstop</a> Value: <b>recordstop</b>
public static final	<a href="#">EVT_seek</a> Value: <b>seek</b>
public static final	<a href="#">EVT_server_start</a> Value: <b>server-start</b>
public static final	<a href="#">EVT_server_stop</a> Value: <b>server-stop</b>
public static final	<a href="#">EVT_setbuffertime</a> Value: <b>setbuffertime</b>

public static final	<a href="#">EVT_setstreamtype</a> Value: <b>setstreamtype</b>
public static final	<a href="#">EVT_stop</a> Value: <b>stop</b>
public static final	<a href="#">EVT_unpause</a> Value: <b>unpause</b>
public static final	<a href="#">EVT_unpublish</a> Value: <b>unpublish</b>
public static final	<a href="#">EVT_vhost_start</a> Value: <b>vhost-start</b>
public static final	<a href="#">EVT_vhost_stop</a> Value: <b>vhost-stop</b>
public static final	<a href="#">FD_ALL</a>
public static final	<a href="#">FD_c_client_id</a> Value: <b>c-client-id</b>
public static final	<a href="#">FD_c_ip</a> Value: <b>c-ip</b>
public static final	<a href="#">FD_c_proto</a> Value: <b>c-proto</b>
public static final	<a href="#">FD_c_referrer</a> Value: <b>c-referrer</b>
public static final	<a href="#">FD_c_user_agent</a> Value: <b>c-user-agent</b>
public static final	<a href="#">FD_cs_bytes</a> Value: <b>cs-bytes</b>
public static final	<a href="#">FD_cs_stream_bytes</a> Value: <b>cs-stream-bytes</b>
public static final	<a href="#">FD_cs_uri_query</a> Value: <b>cs-uri-query</b>
public static final	<a href="#">FD_cs_uri_stem</a> Value: <b>cs-uri-stem</b>



public static final	<a href="#">FD_date</a> Value: <b>date</b>
public static final	<a href="#">FD_s_ip</a> Value: <b>s-ip</b>
public static final	<a href="#">FD_s_port</a> Value: <b>s-port</b>
public static final	<a href="#">FD_s_uri</a> Value: <b>s-uri</b>
public static final	<a href="#">FD_sc_bytes</a> Value: <b>sc-bytes</b>
public static final	<a href="#">FD_sc_stream_bytes</a> Value: <b>sc-stream-bytes</b>
public static final	<a href="#">FD_time</a> Value: <b>time</b>
public static final	<a href="#">FD_tz</a> Value: <b>tz</b>
public static final	<a href="#">FD_x_app</a> Value: <b>x-app</b>
public static final	<a href="#">FD_x_appinst</a> Value: <b>x-appinst</b>
public static final	<a href="#">FD_x_category</a> Value: <b>x-category</b>
public static final	<a href="#">FD_x_comment</a> Value: <b>x-comment</b>
public static final	<a href="#">FD_x_ctx</a> Value: <b>x-ctx</b>
public static final	<a href="#">FD_x_ctx_override</a> Value: <b>x-ctx-override</b>
public static final	<a href="#">FD_x_duration</a> Value: <b>x-duration</b>
public static final	<a href="#">FD_x_event</a> Value: <b>x-event</b>

public static final	<a href="#">FD_x_file_ext</a> Value: <b>x-file-ext</b>
public static final	<a href="#">FD_x_file_length</a> Value: <b>x-file-length</b>
public static final	<a href="#">FD_x_file_name</a> Value: <b>x-file-name</b>
public static final	<a href="#">FD_x_file_size</a> Value: <b>x-file-size</b>
public static final	<a href="#">FD_x_severity</a> Value: <b>x-severity</b>
public static final	<a href="#">FD_x_sname</a> Value: <b>x-sname</b>
public static final	<a href="#">FD_x_sname_query</a> Value: <b>x-sname-query</b>
public static final	<a href="#">FD_x_spos</a> Value: <b>x-spos</b>
public static final	<a href="#">FD_x_status</a> Value: <b>x-status</b>
public static final	<a href="#">FD_x_stream_id</a> Value: <b>x-stream-id</b>
public static final	<a href="#">FD_x_suri</a> Value: <b>x-suri</b>
public static final	<a href="#">FD_x_suri_query</a> Value: <b>x-suri-query</b>
public static final	<a href="#">FD_x_suri_stem</a> Value: <b>x-suri-stem</b>
public static final	<a href="#">FD_x_vhost</a> Value: <b>x-vhost</b>
public static final	<a href="#">PROTO_HTTPCUPERTINO</a> Value: <b>http (cupertino)</b>
public static final	<a href="#">PROTO_HTTPDVRCHUNK</a> Value: <b>http (dvr)</b>

public static final	<a href="#">PROTO_HTTPSANJOSE</a> Value: <b>http (sanjose)</b>
public static final	<a href="#">PROTO_HTTPSCUPERTINO</a> Value: <b>https (cupertino)</b>
public static final	<a href="#">PROTO_HTTPSDVRCHUNK</a> Value: <b>https (dvr)</b>
public static final	<a href="#">PROTO_HTTPSMOOTH</a> Value: <b>http (smooth)</b>
public static final	<a href="#">PROTO_HTTPSSANJOSE</a> Value: <b>https (sanjose)</b>
public static final	<a href="#">PROTO_HTTPSSMOOTH</a> Value: <b>https (smooth)</b>
public static final	<a href="#">PROTO_HTTPSSTREAMER</a> Value: <b>https (streamer)</b>
public static final	<a href="#">PROTO_HTTPSTREAMER</a> Value: <b>http (streamer)</b>
public static final	<a href="#">PROTO_RTMP</a> Value: <b>rtmp</b>
public static final	<a href="#">PROTO_RTMP_E</a> Value: <b>rtmpe</b>
public static final	<a href="#">PROTO_RTMP_S</a> Value: <b>rtmps</b>
public static final	<a href="#">PROTO_RTMP_T</a> Value: <b>rtmpt (HTTP-1.1)</b>
public static final	<a href="#">PROTO_RTMP_T_E</a> Value: <b>rtmpte (HTTP-1.1)</b>
public static final	<a href="#">PROTO_RTMP_T_S</a> Value: <b>rtmpts (HTTP-1.1)</b>
public static final	<a href="#">PROTO_RTSP</a> Value: <b>rtsp</b>
public static final	<a href="#">STAT_connect_application_not_available</a> Value: <b>302</b>

public static final	<a href="#"><u>STAT_connect_application_not_found</u></a> Value: <b>404</b>
public static final	<a href="#"><u>STAT_connect_bad_gateway</u></a> Value: <b>502</b>
public static final	<a href="#"><u>STAT_connect_internal_error</u></a> Value: <b>500</b>
public static final	<a href="#"><u>STAT_connect_license_limit</u></a> Value: <b>413</b>
public static final	<a href="#"><u>STAT_connect_pending_wating</u></a> Value: <b>100</b>
public static final	<a href="#"><u>STAT_connect_redirect</u></a> Value: <b>302</b>
public static final	<a href="#"><u>STAT_connect_rejected_by_application</u></a> Value: <b>401</b>
public static final	<a href="#"><u>STAT_connect_rejected_by_module</u></a> Value: <b>403</b>
public static final	<a href="#"><u>STAT_connect_resource_limit</u></a> Value: <b>409</b>
public static final	<a href="#"><u>STAT_connect_service_unavailable</u></a> Value: <b>503</b>
public static final	<a href="#"><u>STAT_connect_successful</u></a> Value: <b>200</b>
public static final	<a href="#"><u>STAT_connect_unknown_protocol</u></a> Value: <b>400</b>
public static final	<a href="#"><u>STAT_general_internal_error</u></a> Value: <b>500</b>
public static final	<a href="#"><u>STAT_general_successful</u></a> Value: <b>200</b>
public static final	<a href="#"><u>STAT_play_bad_request</u></a> Value: <b>400</b>
public static final	<a href="#"><u>STAT_play_internal_error</u></a> Value: <b>500</b>

public static final	<a href="#">STAT_play_rejected_by_application</a> Value: <b>401</b>
public static final	<a href="#">STAT_play_rejected_by_module</a> Value: <b>403</b>
public static final	<a href="#">STAT_play_stream_not_found</a> Value: <b>404</b>
public static final	<a href="#">STAT_play_successful</a> Value: <b>200</b>
public static final	<a href="#">STAT_play_unsupported_media_type</a> Value: <b>415</b>
public static final	<a href="#">STAT_publish_bad_request</a> Value: <b>400</b>
public static final	<a href="#">STAT_publish_in_use</a> Value: <b>409</b>
public static final	<a href="#">STAT_publish_internal_error</a> Value: <b>500</b>
public static final	<a href="#">STAT_publish_rejected_by_application</a> Value: <b>401</b>
public static final	<a href="#">STAT_publish_successful</a> Value: <b>200</b>
public static final	<a href="#">STAT_publish_unsupported_media_type</a> Value: <b>415</b>
public static final	<a href="#">STAT_stop_client_disconnect</a> Value: <b>408</b>
public static final	<a href="#">STAT_stop_successful</a> Value: <b>200</b>

## Constructor Summary

public	<a href="#">WMSLoggerIDs()</a>
--------	--------------------------------

## 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	<a href="#"><u>ID3FOOTER_SIZE</u></a> Value: <b>10</b>
public static final	<a href="#"><u>ID3HEADER_SIZE</u></a> Value: <b>10</b>
public static final	<a href="#"><u>ID3HEADER_VERSION</u></a> Value: <b>1024</b>
public static final	<a href="#"><u>ID3HEADERFLAGS_DEFAULT</u></a> Value: <b>0</b>
public static final	<a href="#"><u>ID3HEADERFLAGS_EXPERIMENTAL</u></a> Value: <b>32</b>
public static final	<a href="#"><u>ID3HEADERFLAGS_EXTENDED</u></a> Value: <b>64</b>
public static final	<a href="#"><u>ID3HEADERFLAGS_FOOTERPRESENT</u></a> Value: <b>16</b>
public static final	<a href="#"><u>ID3HEADERFLAGS_UNSYNC</u></a> Value: <b>128</b>

### Constructor Summary

public	<a href="#"><u>ID3Frames</u></a> ( )
--------	--------------------------------------

### Method Summary

void	<a href="#"><u>clear</u></a> ( )
java.util.List	<a href="#"><u>getFrameMapIds</u></a> ( )



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

#### 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	<a href="#">MIMETYPES_JPEG</a> Value: <b>image/jpeg</b>
public static final	<a href="#">MIMETYPES_PNG</a> Value: <b>image/png</b>
public static final	<a href="#">MIMETYPES_URL</a> Value: <b>--&gt;</b>
public static	<a href="#">PICTUREMAXFILESIZE</a>
public static final	<a href="#">PICTURETYPE_ARTISTLOGO</a> Value: <b>19</b>
public static final	<a href="#">PICTURETYPE_COVERBACK</a> Value: <b>4</b>
public static final	<a href="#">PICTURETYPE_COVERFRONT</a> Value: <b>3</b>
public static final	<a href="#">PICTURETYPE_FILEICON</a> Value: <b>1</b>
public static final	<a href="#">PICTURETYPE_ILLUSTRATION</a> Value: <b>18</b>
public static final	<a href="#">PICTURETYPE_MOVIESCREENCAPTURE</a> Value: <b>16</b>
public static final	<a href="#">PICTURETYPE_OTHER</a> Value: <b>0</b>

public static final	<a href="#">PICTURETYPE_OTHERFILEICON</a> Value: <b>2</b>
public static final	<a href="#">PICTURETYPE_PUBLISHERLOGO</a> Value: <b>20</b>

#### 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	<a href="#">ID3V2FrameAttachedPicture</a> (String idStr, int flags)
public	<a href="#">ID3V2FrameAttachedPicture</a> ()

## Method Summary

void	<a href="#">deserializeBody</a> (byte[] buffer, int offset, int len)
int	<a href="#">getBodySize</a> ()
String	<a href="#">getDescription</a> ()
String	<a href="#">getMapIdStr</a> ()
String	<a href="#">getMimeType</a> ()
byte[]	<a href="#">getPictureData</a> ()
int	<a href="#">getPictureType</a> ()
int	<a href="#">getTextEncoding</a> ()
boolean	<a href="#">loadFile</a> (java.io.File file)
int	<a href="#">serializeBody</a> (byte[] buffer, int offset)
void	<a href="#">setDescription</a> (String description)
void	<a href="#">setMimeType</a> (String mimeType)

void	<a href="#">setPictureData</a> (byte[] pictureData)
void	<a href="#">setPictureDataAsURL</a> (String urlStr)
void	<a href="#">setPictureType</a> (int pictureType)
void	<a href="#">setTextEncoding</a> (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	<a href="#">FLAGS_DEFAULT</a> Value: <b>0</b>
public static final	<a href="#">HEADERSIZE</a> Value: <b>10</b>
public static final	<a href="#">TAG_APIC</a> Value: <b>APIC</b>
public static final	<a href="#">TAG_COMM</a> Value: <b>COMM</b>
public static final	<a href="#">TAG_LINK</a> Value: <b>LINK</b>
public static final	<a href="#">TAG_PRIV</a> Value: <b>PRIV</b>
public static final	<a href="#">TAG_RBUF</a> Value: <b>RBUF</b>
public static final	<a href="#">TAG_SYLT</a> Value: <b>SYLT</b>
public static final	<a href="#">TAG_TALB</a> Value: <b>TALB</b>

public static final	<a href="#">TAG_TBPM</a> Value: <b>TBPM</b>
public static final	<a href="#">TAG_TCOM</a> Value: <b>TCOM</b>
public static final	<a href="#">TAG_TCON</a> Value: <b>TCON</b>
public static final	<a href="#">TAG_TCOP</a> Value: <b>TCOP</b>
public static final	<a href="#">TAG_TDEN</a> Value: <b>TDEN</b>
public static final	<a href="#">TAG_TDLY</a> Value: <b>TDLY</b>
public static final	<a href="#">TAG_TDOR</a> Value: <b>TDOR</b>
public static final	<a href="#">TAG_TDRC</a> Value: <b>TDRC</b>
public static final	<a href="#">TAG_TDRL</a> Value: <b>TDRL</b>
public static final	<a href="#">TAG_TDTG</a> Value: <b>TDTG</b>
public static final	<a href="#">TAG_TENC</a> Value: <b>TENC</b>
public static final	<a href="#">TAG_TEXT</a> Value: <b>TEXT</b>
public static final	<a href="#">TAG_TFLT</a> Value: <b>TFLT</b>
public static final	<a href="#">TAG_TIPL</a> Value: <b>TIPL</b>
public static final	<a href="#">TAG_TIT1</a> Value: <b>TIT1</b>
public static final	<a href="#">TAG_TIT2</a> Value: <b>TIT2</b>

public static final	<a href="#"><u>TAG_TIT3</u></a> Value: <b>TIT3</b>
public static final	<a href="#"><u>TAG_TKEY</u></a> Value: <b>TKEY</b>
public static final	<a href="#"><u>TAG_TLAN</u></a> Value: <b>TLAN</b>
public static final	<a href="#"><u>TAG_TLEN</u></a> Value: <b>TLEN</b>
public static final	<a href="#"><u>TAG_TMCL</u></a> Value: <b>TMCL</b>
public static final	<a href="#"><u>TAG_TMED</u></a> Value: <b>TMED</b>
public static final	<a href="#"><u>TAG_TMOO</u></a> Value: <b>TMOO</b>
public static final	<a href="#"><u>TAG_TOAL</u></a> Value: <b>TOAL</b>
public static final	<a href="#"><u>TAG_TOFN</u></a> Value: <b>TOFN</b>
public static final	<a href="#"><u>TAG_TOLY</u></a> Value: <b>TOLY</b>
public static final	<a href="#"><u>TAG_TOPE</u></a> Value: <b>TOPE</b>
public static final	<a href="#"><u>TAG_TOWN</u></a> Value: <b>TOWN</b>
public static final	<a href="#"><u>TAG_TPE1</u></a> Value: <b>TPE1</b>
public static final	<a href="#"><u>TAG_TPE2</u></a> Value: <b>TPE2</b>
public static final	<a href="#"><u>TAG_TPE3</u></a> Value: <b>TPE3</b>
public static final	<a href="#"><u>TAG_TPE4</u></a> Value: <b>TPE4</b>

public static final	<a href="#">TAG_TPOS</a> Value: <b>TPOS</b>
public static final	<a href="#">TAG_TPRO</a> Value: <b>TPRO</b>
public static final	<a href="#">TAG_TPUB</a> Value: <b>TPUB</b>
public static final	<a href="#">TAG_TRCK</a> Value: <b>TRCK</b>
public static final	<a href="#">TAG_TRSN</a> Value: <b>TRSN</b>
public static final	<a href="#">TAG_TRSO</a> Value: <b>TRSO</b>
public static final	<a href="#">TAG_TSOA</a> Value: <b>TSOA</b>
public static final	<a href="#">TAG_TSOP</a> Value: <b>TSOP</b>
public static final	<a href="#">TAG_TSOT</a> Value: <b>TSOT</b>
public static final	<a href="#">TAG_TSRC</a> Value: <b>TSRC</b>
public static final	<a href="#">TAG_TSSE</a> Value: <b>TSSE</b>
public static final	<a href="#">TAG_TSST</a> Value: <b>TSST</b>
public static final	<a href="#">TAG_TXXX</a> Value: <b>TXXX</b>
public static final	<a href="#">TAG_UNKN</a> Value: <b>UNKN</b>
public static final	<a href="#">TAG_WCOM</a> Value: <b>WCOM</b>
public static final	<a href="#">TAG_WCOP</a> Value: <b>WCOP</b>

public static final	<a href="#">TAG_WOAF</a> Value: <b>WOAF</b>
public static final	<a href="#">TAG_WOAR</a> Value: <b>WOAR</b>
public static final	<a href="#">TAG_WOAS</a> Value: <b>WOAS</b>
public static final	<a href="#">TAG_WORS</a> Value: <b>WORS</b>
public static final	<a href="#">TAG_WPAY</a> Value: <b>WPAY</b>
public static final	<a href="#">TAG_WPUB</a> Value: <b>WPUB</b>
public static final	<a href="#">TAG_WXXX</a> Value: <b>WXXX</b>
public static final	<a href="#">TAGS_TEXTINFORMATION</a>
public static final	<a href="#">TEXTENCODING_ISO_8859_1</a> Value: <b>0</b>
public static final	<a href="#">TEXTENCODING_UTF16</a> Value: <b>1</b>
public static final	<a href="#">TEXTENCODING_UTF16BE</a> Value: <b>2</b>
public static final	<a href="#">TEXTENCODING_UTF8</a> Value: <b>3</b>

## Constructor Summary

public	<a href="#">ID3V2FrameBase</a> (String idStr, int flags)
--------	--

## Method Summary

static int	<a href="#">byteStringLen</a> (String value)
static <a href="#">ID3V2FrameBase</a>	<a href="#">deserializeFrame</a> (byte[] buffer, int offset, int len)
static String	<a href="#">deserializeString</a> (byte[] buffer, int offset, int len)
int	<a href="#">getFlags</a> ()



String	<a href="#">getIdStr()</a>
String	<a href="#">getMapIdStr()</a>
int	<a href="#">getSize()</a>
int	<a href="#">serialize</a> (byte[] buffer, int offset)
static int	<a href="#">serializeString</a> (String value, byte[] buffer, int offset)
static int	<a href="#">serializeStringLen</a> (String value)
void	<a href="#">setFlags</a> (int flags)
void	<a href="#">setIdStr</a> (String idStr)
static String	<a href="#">trimTrailingZero</a> (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()
```

---

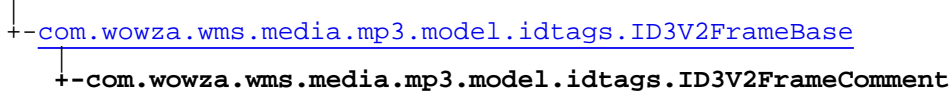
**getMapIdStr**

```
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\\_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	<a href="#">ID3V2FrameComment</a> (String idStr, int flags)
public	<a href="#">ID3V2FrameComment</a> (String idStr)
public	<a href="#">ID3V2FrameComment</a> ()

### Method Summary

void	<a href="#">deserializeBody</a> (byte[] buffer, int offset, int len)
int	<a href="#">getBodySize</a> ()
String	<a href="#">getDescription</a> ()
int	<a href="#">getTextEncoding</a> ()
String	<a href="#">getValue</a> ()
int	<a href="#">serializeBody</a> (byte[] buffer, int offset)

void	<a href="#"><u>setDescription</u></a> (String description)
void	<a href="#"><u>setTextEncoding</u></a> (int textEncoding)
void	<a href="#"><u>setValue</u></a> (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()
```

---

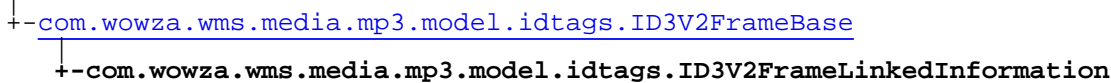
## setDescription

```
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\\_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	<a href="#">ID3V2FrameLinkedInformation</a> (String idStr, int flags)
public	<a href="#">ID3V2FrameLinkedInformation</a> ()

### Method Summary

void	<a href="#">deserializeBody</a> (byte[] buffer, int offset, int len)
int	<a href="#">getBodySize</a> ()
byte[]	<a href="#">getData</a> ()
String	<a href="#">getDescription</a> ()
long	<a href="#">getFrameIdentifier</a> ()
String	<a href="#">getURL</a> ()
int	<a href="#">serializeBody</a> (byte[] buffer, int offset)

void	<a href="#"><u>setData</u></a> (byte[] data)
void	<a href="#"><u>setDescription</u></a> (String description)
void	<a href="#"><u>setFrameIdentifier</u></a> (long frameIdentifier)
void	<a href="#"><u>setURL</u></a> (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**

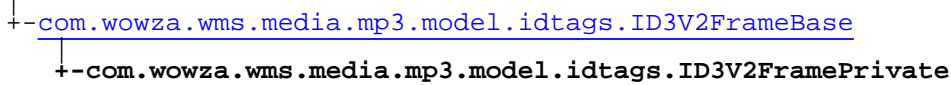
```
public void setData(byte[] data)
```

---

## 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\\_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	<a href="#">ID3V2FramePrivate</a> (String idStr, int flags)
public	<a href="#">ID3V2FramePrivate</a> ()

### Method Summary

void	<a href="#">deserializeBody</a> (byte[] buffer, int offset, int len)
int	<a href="#">getBodySize</a> ()
int	<a href="#">serializeBody</a> (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)
```

---

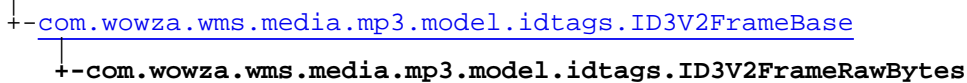
### getBodySize

```
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	<a href="#">ID3V2FrameRawBytes</a> (String idStr, int flags)
public	<a href="#">ID3V2FrameRawBytes</a> (String idStr)

### Method Summary

void	<a href="#">deserializeBody</a> (byte[] buffer, int offset, int len)
int	<a href="#">getBodySize</a> ()
int	<a href="#">serializeBody</a> (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

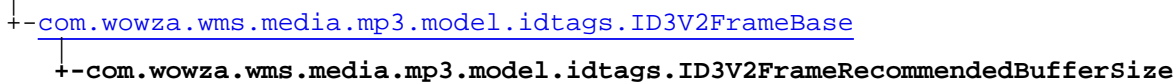
```
public int getBodySize()
```

---

## 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	<a href="#">ID3V2FrameRecommendedBufferSize</a> (String idStr, int flags)
public	<a href="#">ID3V2FrameRecommendedBufferSize</a> ()

### Method Summary

void	<a href="#">deserializeBody</a> (byte[] buffer, int offset, int len)
int	<a href="#">getBodySize</a> ()
int	<a href="#">getBufferSize</a> ()
byte	<a href="#">getEmbeddedFlag</a> ()
long	<a href="#">getOffsetToNextTag</a> ()
int	<a href="#">serializeBody</a> (byte[] buffer, int offset)
void	<a href="#">setBufferSize</a> (int bufferSize)

void	<a href="#">setEmbeddedFlag</a> (byte embeddedFlag)
void	<a href="#">setOffsetToNextTag</a> (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	<a href="#">CONTENTTYPE_CHORD</a> Value: <b>5</b>
public static final	<a href="#">CONTENTTYPE_EVENTS</a> Value: <b>4</b>
public static final	<a href="#">CONTENTTYPE_LYRICS</a> Value: <b>1</b>
public static final	<a href="#">CONTENTTYPE_MOVEMENT</a> Value: <b>3</b>
public static final	<a href="#">CONTENTTYPE_OTHER</a> Value: <b>0</b>
public static final	<a href="#">CONTENTTYPE_TRANSCRIPTION</a> Value: <b>2</b>
public static final	<a href="#">CONTENTTYPE_TRIVIA</a> Value: <b>6</b>
public static final	<a href="#">CONTENTTYPE_URLIMAGES</a> Value: <b>8</b>
public static final	<a href="#">CONTENTTYPE_URLWEBPAGES</a> Value: <b>7</b>
public static final	<a href="#">TIMESTAMPFORMAT_MILLISECONDS</a> Value: <b>2</b>

public static final	<a href="#"><u>TIMESTAMPFORMAT_MPEG</u></a> Value: <b>1</b>
---------------------	--

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	<a href="#"><u>ID3V2FrameSynchronizedText</u></a> (String idStr, int flags)
public	<a href="#"><u>ID3V2FrameSynchronizedText</u></a> ()

## Method Summary

void	<a href="#"><u>addContentDescriptor</u></a> ( <a href="#"><u>ID3V2FrameSynchronizedTextDescriptor</u></a> contentDescriptor)
void	<a href="#"><u>addContentDescriptor</u></a> (long timecode, String value)
void	<a href="#"><u>deserializeBody</u></a> (byte[] buffer, int offset, int len)
int	<a href="#"><u>getBodySize</u></a> ()
java.util.List	<a href="#"><u>getContentDescriptors</u></a> ()
byte	<a href="#"><u>getContentType</u></a> ()
String	<a href="#"><u>getLanguage</u></a> ()
int	<a href="#"><u>getTextEncoding</u></a> ()
byte	<a href="#"><u>getTimeStampFormat</u></a> ()
int	<a href="#"><u>serializeBody</u></a> (byte[] buffer, int offset)
void	<a href="#"><u>setContentType</u></a> (byte contentType)
void	<a href="#"><u>setLanguage</u></a> (String language)
void	<a href="#"><u>setTextEncoding</u></a> (int textEncoding)

void	<a href="#"><u>setTimeStampFormat</u></a> (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	<a href="#">ID3V2FrameSynchronizedTextDescriptor()</a>
public	<a href="#">ID3V2FrameSynchronizedTextDescriptor(long timecode, String value)</a>

### Method Summary

long	<a href="#">getTimecode()</a>
String	<a href="#">getValue()</a>
void	<a href="#">setTimecode(long timecode)</a>
void	<a href="#">setValue(String value)</a>

#### 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)
```

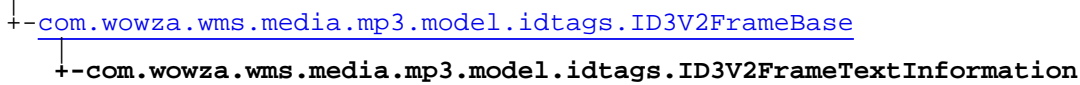
---



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

### Class ID3V2FrameTextInformation

java.lang.Object



All Implemented Interfaces:

[IID3V2Frame](#)

public class **ID3V2FrameTextInformation**  
 extends [ID3V2FrameBase](#)

#### Field Summary

protected	<a href="#">addTrailingZero</a>
-----------	---------------------------------

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	<a href="#">ID3V2FrameTextInformation</a> (String idStr, int flags)
public	<a href="#">ID3V2FrameTextInformation</a> (String idStr)

#### Method Summary

void	<a href="#">deserializeBody</a> (byte[] buffer, int offset, int len)
int	<a href="#">getBodySize</a> ()
int	<a href="#">getTextEncoding</a> ()
String	<a href="#">getValue</a> ()
int	<a href="#">serializeBody</a> (byte[] buffer, int offset)

void	<a href="#">setTextEncoding</a> (int textEncoding)
void	<a href="#">setValue</a> (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

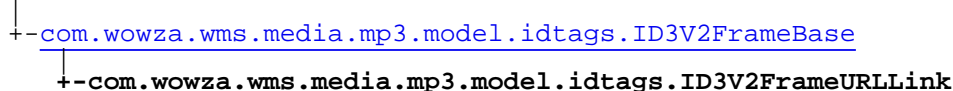
```
public void setTextEncoding(int textEncoding)
```

---

## 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	<a href="#">ID3V2FrameURLLink</a> (String idStr, int flags)
public	<a href="#">ID3V2FrameURLLink</a> (String idStr)

## Method Summary

void	<a href="#">deserializeBody</a> (byte[] buffer, int offset, int len)
int	<a href="#">getBodySize</a> ()
int	<a href="#">getTextEncoding</a> ()
String	<a href="#">getURL</a> ()
int	<a href="#">serializeBody</a> (byte[] buffer, int offset)
void	<a href="#">setTextEncoding</a> (int textEncoding)
void	<a href="#">setURL</a> (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	<a href="#">ID3V2Utils()</a>
--------	------------------------------

Method Summary

static int	<a href="#">byteArrayToIntSafeSync</a> (byte[] b, int offset, int count)
static long	<a href="#">byteArrayToLongSafeSync</a> (byte[] b, int offset, int count)
static void	<a href="#">intToByteArraySafeSync</a> (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	<a href="#">deserializeBody</a> (byte[] buffer, int offset, int len)
int	<a href="#">getBodySize</a> ()
int	<a href="#">getFlags</a> ()
String	<a href="#">getIdStr</a> ()
String	<a href="#">getMapIdStr</a> ()
int	<a href="#">getSize</a> ()
int	<a href="#">serialize</a> (byte[] buffer, int offset)
int	<a href="#">serializeBody</a> (byte[] buffer, int offset)
void	<a href="#">setFlags</a> (int flags)
void	<a href="#">setIdStr</a> (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	<a href="#"><u>MEDIACASTERTYPE_LIVEREPEATER</u></a> Value: <b>1</b>
public static final	<a href="#"><u>MEDIACASTERTYPE_RTPLIVE</u></a> Value: <b>3</b>
public static final	<a href="#"><u>MEDIACASTERTYPE_SHOUTCAST</u></a> Value: <b>2</b>
public static final	<a href="#"><u>MEDIACASTERTYPE_UNKNOWN</u></a> Value: <b>0</b>
public static final	<a href="#"><u>STREAMTIMEOUTREASON_GOOD</u></a> Value: <b>100</b>
public static final	<a href="#"><u>STREAMTIMEOUTREASON_MISSING</u></a> Value: <b>101</b>
public static final	<a href="#"><u>STREAMTIMEOUTREASON_NORTSPSESSION</u></a> Value: <b>6</b>
public static final	<a href="#"><u>STREAMTIMEOUTREASON_NOSESSION</u></a> Value: <b>2</b>
public static final	<a href="#"><u>STREAMTIMEOUTREASON_NOSTREAM</u></a> Value: <b>4</b>
public static final	<a href="#"><u>STREAMTIMEOUTREASON_NOTIMEOUT</u></a> Value: <b>1</b>
public static final	<a href="#"><u>STREAMTIMEOUTREASON_NOURL</u></a> Value: <b>3</b>
public static final	<a href="#"><u>STREAMTIMEOUTREASON_RECONNECTRUNNING</u></a> Value: <b>5</b>
public static final	<a href="#"><u>STREAMTIMEOUTREASON_UNKNOWN</u></a> Value: <b>0</b>

## Method Summary

boolean	<a href="#"><u>doWatchdog()</u></a> Idle processor
void	<a href="#"><u>forceReset()</u></a> Force a reset/reconnect of this media caster
<a href="#"><u>IApplicationInstance</u></a>	<a href="#"><u>getAppInstance()</u></a> Get the application instance this media caster is associated with
long	<a href="#"><u>getConnectLastAttempt()</u></a> Get system time in milliseconds of last connection attempt
long	<a href="#"><u>getConnectLastForceReset()</u></a> Get system time in milliseconds of last time forceReset was called
long	<a href="#"><u>getConnectLastSuccess()</u></a> Get system time in milliseconds of last connection success
int	<a href="#"><u>getIdleTimeout()</u></a> Get the idle timeout for this media caster (milliseconds)
<a href="#"><u>MediaCasterItem</u></a>	<a href="#"><u>getMediaCasterDef()</u></a> Get the media caster definition
String	<a href="#"><u>getMediaCasterId()</u></a> Get the media caster id
<a href="#"><u>MediaCasterStreamItem</u></a>	<a href="#"><u>getMediaCasterStreamItem()</u></a> Get the media caster item associated with this media caster
int	<a href="#"><u>getMediaCasterType()</u></a> Get the media caster type.
int	<a href="#"><u>getReconnectWaitTime()</u></a> Get the minimum time between reconnect attempts (milliseconds)
<a href="#"><u>IMediaStream</u></a>	<a href="#"><u>getStream()</u></a> Get the underlying stream being used by this media caster
Object	<a href="#"><u>getStreamIsRunningLock()</u></a> Get stream running lock
long	<a href="#"><u>getStreamLastSeq()</u></a> Get the AMFPacket sequence number of last watchdog processed packet
long	<a href="#"><u>getStreamMissingTime()</u></a> Get the time in milliseconds the stream has been missing
int	<a href="#"><u>getStreamTimeout()</u></a> Get the watchdog stream timeout (milliseconds)
long	<a href="#"><u>getStreamTimeoutLastReset()</u></a> Get system time in milliseconds of last time stream was reset due to stream timeout (debug)
long	<a href="#"><u>getStreamTimeoutLastTime()</u></a> Get system time in milliseconds of last time stream was considered in missing state (debug)

int	<a href="#"><u>getStreamTimeoutReason()</u></a> Get the reason the stream is in timeout condition (debug)
<a href="#"><u>IVHost</u></a>	<a href="#"><u>getVHost()</u></a> Get the virtual host associated with this media caster
void	<a href="#"><u>init(MediaCasterStreamItem mediaCasterStreamItem, MediaCasterItem mediaCasterDef, IApplicationInstance appInstance, String mediaCasterId, String streamExt)</u></a> Initialize the media caster
boolean	<a href="#"><u>isSession()</u></a> Is there current a session attached to this MediaCaster
boolean	<a href="#"><u>isStream()</u></a> Is there a stream associated with this MediaCaster
boolean	<a href="#"><u>isStreamIsRunning()</u></a> Return true if stream is currently running
void	<a href="#"><u>registerPlayer(IMediaStreamPlay player)</u></a> Register a player with this media caster
void	<a href="#"><u>sessionClosed(org.apache.mina.common.Session session)</u></a> sessionClosed callback
void	<a href="#"><u>sessionOpened(org.apache.mina.common.Session session)</u></a> sessionOpened callback
void	<a href="#"><u>setAppInstance(IApplicationInstance appInstance)</u></a> Set the application instance this media caster is associated with
void	<a href="#"><u>setMediaCasterDef(MediaCasterItem mediaCasterDef)</u></a> Set the media caster definition
void	<a href="#"><u>setMediaCasterId(String mediaCasterId)</u></a> Get the media caster id
void	<a href="#"><u>setMediaCasterType(int mediaCasterType)</u></a> Set the media caster type.
void	<a href="#"><u>setReconnectWaitTime(int reconnectWaitTime)</u></a> Set the minimum time between reconnect attempts (milliseconds)
void	<a href="#"><u>setStream(IMediaStream stream)</u></a> Set the underlying stream being used by this media caster
void	<a href="#"><u>setStreamTimeout(int streamTimeout)</u></a> Set the watchdog stream timeout (milliseconds)
void	<a href="#"><u>shutdown(boolean isAppInstanceShutdown)</u></a> Shutdown media caster
void	<a href="#"><u>unregisterPlayer(IMediaStreamPlay player)</u></a> 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	<a href="#">onData</a> (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	<a href="#">getNetConnection()</a> 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	<a href="#">onMediaCasterCreate</a> ( <a href="#">IMediaCaster</a> mediaCaster) Invoked when mediaCaster created
void	<a href="#">onMediaCasterDestroy</a> ( <a href="#">IMediaCaster</a> mediaCaster) Invoked when MediaCaster destroyed
void	<a href="#">onRegisterPlayer</a> ( <a href="#">IMediaCaster</a> mediaCaster, <a href="#">IMediaStreamPlay</a> player) Invoked when a player is added to this mediaCaster
void	<a href="#">onSetSourceStream</a> ( <a href="#">IMediaCaster</a> mediaCaster, <a href="#">IMediaStream</a> stream) Invoked when soure stream is set (can be called with stream of null)
void	<a href="#">onUnRegisterPlayer</a> ( <a href="#">IMediaCaster</a> mediaCaster, <a href="#">IMediaStreamPlay</a> 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	<a href="#">onConnectFailure</a> ( <a href="#">IMediaCaster</a> mediaCaster) Invoked when a connection or reconnection attempt fails
void	<a href="#">onConnectStart</a> ( <a href="#">IMediaCaster</a> mediaCaster) Invoked when a connection or reconnection attempt is invoked
void	<a href="#">onConnectSuccess</a> ( <a href="#">IMediaCaster</a> mediaCaster) Invoked when a connection or reconnection attempt is successful
void	<a href="#">onStreamStart</a> ( <a href="#">IMediaCaster</a> mediaCaster) Invoked when the stream starts receiving media data from the media source.
void	<a href="#">onStreamStop</a> ( <a href="#">IMediaCaster</a> 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	<a href="#">onResetMediaCaster</a> ( <a href="#">IApplicationInstance</a> appInstance, <a href="#">IMediaCaster</a> mediaCaster) Called when media caster is reset
boolean	<a href="#">onValidateMediaCaster</a> ( <a href="#">IApplicationInstance</a> appInstance, <a href="#">IMediaCaster</a> mediaCaster) Called for each media caster to validate the media caster.
void	<a href="#">onValidateMediaCastersStart</a> ( <a href="#">IApplicationInstance</a> appInstance) Called when validation for all streams of an application instance is starting
void	<a href="#">onValidateMediaCastersStop</a> ( <a href="#">IApplicationInstance</a> 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	<a href="#"><u>MediaCasterItem</u></a> (String name, String streamType, String baseClass) Media caster item constructor
--------	--

#### Method Summary

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

void	<a href="#"><code>setBaseClass</code></a> (String baseClass) Set base class
void	<a href="#"><code>setConnectionTimeout</code></a> (int connectionTimeout) Set connection timeout (milliseconds)
void	<a href="#"><code>setDescription</code></a> (String description) Set description
void	<a href="#"><code>setKeepAliveTime</code></a> (int keepAliveTime) Set keep alive time (milliseconds)
void	<a href="#"><code>setName</code></a> (String name) Set name
void	<a href="#"><code>setProperty</code></a> (String name, String value) Set a property
void	<a href="#"><code>setStreamType</code></a> (String streamType) Set stream type
String	<a href="#"><code>toString</code></a> () 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	<a href="#"><u>MediaCasterList()</u></a> Constructor
--------	---

### Method Summary

<a href="#"><u>MediaCasterItem</u></a>	<a href="#"><u>getMediaCasterDef()</u></a> (String name) Get media caster definition by name
java.util.Map	<a href="#"><u>getMediaCasterDefs()</u></a> Get map of media caster items
java.util.List	<a href="#"><u>getMediaCasterNames()</u></a> 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	<a href="#">MediaCasterSettings()</a>
--------	---------------------------------------

## Method Summary

HostPortConfig	<a href="#">getMediaCasterHostPortConfig()</a> Get media caster host port config
int	<a href="#">getMediaCasterProcessorCount()</a> Get the thread count use for this host port
void	<a href="#">setMediaCasterHostPortConfig</a> (HostPortConfig mediaCasterHostPortConfig) Set media caster host port config
void	<a href="#">setMediaCasterProcessorCount</a> (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	<a href="#"><u>MediaCasterStreamItem</u></a> (long uniqueId) Media caster item constructor
--------	---

### Method Summary

void	<a href="#"><u>acquire</u></a> ( ) Increment acquire lock count for this media caster item
void	<a href="#"><u>acquireAndRelease</u></a> ( ) Increment then decrement acquire lock count for this media caster item
boolean	<a href="#"><u>doWatchdog</u></a> ( ) Do idle processing
String	<a href="#"><u>getLiveStreamPacketizer</u></a> ( ) Get the live stream packetizer for this media caster stream item
String	<a href="#"><u>getLiveStreamRepeater</u></a> ( ) Get the live stream repeater for this media caster stream item
Object	<a href="#"><u>getLock</u></a> ( )
int	<a href="#"><u>getLockCount</u></a> ( ) Get the current number of acquire locks on this media caster item
<a href="#"><u>IMediaCaster</u></a>	<a href="#"><u>getMediaCaster</u></a> ( ) Get the underlying IMediaCaster interface for this MediaCaster
String	<a href="#"><u>getMediaCasterId</u></a> ( ) Get this media caster item id
int	<a href="#"><u>getPlayerCount</u></a> ( ) Get the current number of players associated with this media caster item
String	<a href="#"><u>getStreamExt</u></a> ( )
long	<a href="#"><u>getUniqueId</u></a> ( )

void	<a href="#"><code>init</code></a> (String mediaCasterId, String streamExt, <a href="#"><code>MediaCasterItem</code></a> mediaCasterDef, <a href="#"><code>MediaCasterStreamMap</code></a> parent, String liveStreamPacketizer, String liveStreamRepeater) Initialize the media caster item (internal use)
boolean	<a href="#"><code>isShutdownOnRelease</code></a> ( ) On last release shutdown the stream even if clients are connected
boolean	<a href="#"><code>isValid</code></a> ( )
void	<a href="#"><code>registerPlayer</code></a> ( <a href="#"><code>IMediaStreamPlay</code></a> player) Register a player with a media caster item (internal use)
void	<a href="#"><code>release</code></a> ( ) Decrement acquire lock count for this media caster item
void	<a href="#"><code>reset</code></a> ( ) Force a reconnect or reset for this media caster item
void	<a href="#"><code>setLiveStreamPacketizer</code></a> (String liveStreamPacketizer) Set the live stream packetizer for this media caster stream item
void	<a href="#"><code>setLiveStreamRepeater</code></a> (String liveStreamRepeater) Set the live stream repeater for this media caster stream item
void	<a href="#"><code>setShutdownOnRelease</code></a> (boolean shutdownOnRelease) On last release shutdown the stream even if clients are connected
void	<a href="#"><code>setStreamExt</code></a> (String streamExt)
void	<a href="#"><code>setValid</code></a> (boolean isValid)
void	<a href="#"><code>shutdown</code></a> (boolean isAppInstanceShutdown) Shutdown this media caster item
void	<a href="#"><code>unregisterPlayer</code></a> ( <a href="#"><code>IMediaStreamPlay</code></a> 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	<a href="#">MediaCasterStreamManager</a> ( <a href="#">MediaCasterStreamMap</a> mediaCasterStreamMap)
--------	---

### Method Summary

String[]	<a href="#">getStreamArray</a> () Get a list of active streams
java.util.List	<a href="#">getStreamList</a> () Get a list of active streams
boolean	<a href="#">startStream</a> (String streamName, String mediaCasterType) Start a stream by name
boolean	<a href="#">stopStream</a> (String streamName) Stop a stream by name
boolean	<a href="#">streamExists</a> (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	<a href="#">MediaCasterStreamMap</a> ( <a href="#">IApplicationInstance</a> appInstance) Create a new mediacaster map
--------	--

### Method Summary

<a href="#">MediaCasterStreamItem</a>	<a href="#">acquire</a> (String streamName) Increment the lock count for a media caster item (so that it remains loaded even if all listeners go away).
<a href="#">MediaCasterStreamItem</a>	<a href="#">acquire</a> (String inStreamName, <a href="#">MediaCasterItem</a> mediaCasterDef) Increment the lock count for a media caster item (so that it remains loaded even if all listeners go away).
<a href="#">MediaCasterStreamItem</a>	<a href="#">acquire</a> (String inStreamName, String streamType) Increment the lock count for a media caster item (so that it remains loaded even if all listeners go away).
<a href="#">MediaCasterStreamItem</a>	<a href="#">acquire</a> (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).
<a href="#">MediaCasterStreamItem</a>	<a href="#">acquire</a> (String inStreamName, String liveStreamPacketizer, String liveStreamRepeater, <a href="#">MediaCasterItem</a> mediaCasterDef) Increment the lock count for a media caster item (so that it remains loaded even if all listeners go away).
<a href="#">MediaCasterStreamItem</a>	<a href="#">acquire</a> (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).
<a href="#">MediaCasterStreamItem</a>	<a href="#">acquireAndRelease</a> (String inStreamName, String liveStreamPacketizer, String liveStreamRepeater, <a href="#">MediaCasterItem</a> 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	<a href="#">addStreamSrcToMediaCaster</a> (long streamSrc, String mediaCasterId)
void	<a href="#">clearStreamSrcToMediaCaster</a> (long streamSrc)
void	<a href="#">doWatchdog</a> () Do periodic idle time processing

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

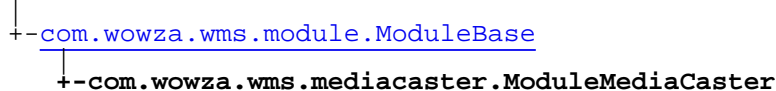
**Parameters:**

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	<a href="#">ModuleMediaCaster</a> ()
--------	--------------------------------------

### Method Summary

void	<a href="#">acquireMediaCaster</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params) Increment the lock count of a media caster stream.
void	<a href="#">getLockCount</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params) Get the current lock count for a stream
void	<a href="#">getPlayerCount</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params) Get the numbers of players associated with a particular media caster
void	<a href="#">getStreamNames</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params) Get an array of stream names that are media casters associate with this application instance (returned as AMFDataArray)
void	<a href="#">onAppStart</a> ( <a href="#">IApplicationInstance</a> appInstance) onAppStart



void	<a href="#">onAppStop</a> ( <a href="#">IApplicationInstance</a> appInstance) onAppStop
void	<a href="#">releaseMediaCaster</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params) Decrement the lock count of a media caster stream
void	<a href="#">resetStream</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params) Reset a media caster stream
void	<a href="#">shutdownStream</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> 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	<a href="#"><code>doIdle</code></a> (MediaCaster mediaCaster)
RTPSessionDescription Data	<a href="#"><code>getSessionDescriptionData</code></a> ( <a href="#">IApplicationInstance</a> appInstance, String streamName, int retryCount, <a href="#">IRTPSessionDescriptionSessionHandler</a> handler, <a href="#">IMediaCaster</a> mediaCaster)
void	<a href="#"><code>init</code></a> (MediaCaster mediaCaster)
void	<a href="#"><code>sessionStart</code></a> ( <a href="#">RTPSession</a> rtpSession)
void	<a href="#"><code>sessionStop</code></a> ( <a href="#">RTPSession</a> 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	<a href="#">onDisconnect()</a>
------	--------------------------------

---

### 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	<a href="#"><code>onCodecConfigAAC</code></a> (com.wowza.wms.media.aac.AACFrame frame, byte[] buffer, long offset)
void	<a href="#"><code>onFrameAAC</code></a> (com.wowza.wms.media.aac.AACFrame frame, byte[] buffer, long offset)
void	<a href="#"><code>onFrameMP3</code></a> (int frequency, int samplesPerFrame, int channels, byte[] syncHeader, byte[] packetHeader, byte[] frameData)
void	<a href="#"><code>onHeaderData</code></a> (java.util.Map headerMap)
void	<a href="#"><code>onMetaData</code></a> (java.util.Map metaMap)
void	<a href="#"><code>onTrim</code></a> ()

### 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	<a href="#">addMetaDataListener</a> ( <a href="#">IShoutCastMetaDataNotify</a> listener)
boolean	<a href="#">removeMetaDataListener</a> ( <a href="#">IShoutCastMetaDataNotify</a> 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	<a href="#">onAACEncodeInfo</a> ( <a href="#">IMediaCaster</a> mediaCaster, int frequency, int channels, int samplesPerFrame)
void	<a href="#">onHeaderData</a> ( <a href="#">IMediaCaster</a> mediaCaster, java.util.Map headerMap)
void	<a href="#">onMetaData</a> ( <a href="#">IMediaCaster</a> mediaCaster, java.util.Map metaMap)
void	<a href="#">onMP3EncodeInfo</a> ( <a href="#">IMediaCaster</a> 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	<a href="#">lock</a>
protected	<a href="#">name</a>
protected	<a href="#">properties</a>
protected	<a href="#">segments</a>

#### Constructor Summary

public	<a href="#">MediaList()</a>
--------	-----------------------------

#### Method Summary

void	<a href="#">addSegment</a> (int index, <a href="#">MediaListSegment</a> mediaListSegment)
void	<a href="#">addSegment</a> ( <a href="#">MediaListSegment</a> mediaListSegment)
void	<a href="#">clearSegments</a> ()
<a href="#">MediaListSegment</a>	<a href="#">getFirstSegment</a> ()
Object	<a href="#">getLock</a> ()
String	<a href="#">getName</a> ()
<a href="#">WMSProperties</a>	<a href="#">getProperties</a> ()
<a href="#">WMSProperties</a>	<a href="#">getProperties</a> (boolean write)
java.util.List	<a href="#">getSegment</a> ()
<a href="#">MediaListSegment</a>	<a href="#">getSegment</a> (int index)

<a href="#">MediaListSegment</a>	<a href="#">removeSegment</a> (int index)
void	<a href="#">removeSegment</a> ( <a href="#">MediaListSegment</a> mediaListSegment)
void	<a href="#">reset</a> ()
void	<a href="#">setName</a> (String name)
String	<a href="#">toSMILString</a> ()
String	<a href="#">toString</a> ()

**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	<a href="#">audioCodecId</a>
protected	<a href="#">bitrateAudio</a>
protected	<a href="#">bitrateTotal</a>
protected	<a href="#">bitrateVideo</a>
protected	<a href="#">height</a>
protected	<a href="#">lock</a>
protected	<a href="#">mediaListSegment</a>
protected	<a href="#">name</a>
protected	<a href="#">playDuration</a>
protected	<a href="#">playStart</a>
protected	<a href="#">properties</a>
protected	<a href="#">type</a>
protected	<a href="#">videoCodecId</a>
protected	<a href="#">width</a>
protected	<a href="#">wowzaAudioOnly</a>

### Constructor Summary

public	<a href="#">MediaListRendition()</a>
--------	--------------------------------------

## Method Summary

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

void	<a href="#">setHeight</a> (int height)
void	<a href="#">setLock</a> (Object lock)
void	<a href="#">setMediaListSegment</a> ( <a href="#">MediaListSegment</a> mediaListSegment)
void	<a href="#">setName</a> (String name)
void	<a href="#">setPlayDuration</a> (long playDuration)
void	<a href="#">setPlayStart</a> (long playStart)
void	<a href="#">setSize</a> (int width, int height)
void	<a href="#">setType</a> (int type)
void	<a href="#">setVideoCodecId</a> (String videoCodecId)
void	<a href="#">setWidth</a> (int width)
void	<a href="#">setWowzaAudioOnly</a> (boolean wowzaAudioOnly)
String	<a href="#">toSMILString</a> ()
String	<a href="#">toString</a> ()

#### 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	<a href="#">lock</a>
protected	<a href="#">mediaList</a>
protected	<a href="#">properties</a>
protected	<a href="#">renditions</a>

#### Constructor Summary

public	<a href="#">MediaListSegment()</a>
--------	------------------------------------

#### Method Summary

void	<a href="#">addRendition</a> (int index, <a href="#">MediaListRendition</a> mediaListRendition)
void	<a href="#">addRendition</a> ( <a href="#">MediaListRendition</a> mediaListRendition)
void	<a href="#">clearSegments</a> ()
<a href="#">MediaListRendition</a>	<a href="#">getFirstRendition</a> ()
Object	<a href="#">getLock</a> ()
<a href="#">MediaList</a>	<a href="#">getMediaList</a> ()
<a href="#">WMSProperties</a>	<a href="#">getProperties</a> ()
<a href="#">WMSProperties</a>	<a href="#">getProperties</a> (boolean write)
<a href="#">MediaListRendition</a>	<a href="#">getRendition</a> (int index)
java.util.List	<a href="#">getRenditions</a> ()

<a href="#">MediaListRendition</a>	<a href="#">removeRendition</a> (int index)
void	<a href="#">removeRendition</a> ( <a href="#">MediaListRendition</a> mediaListRendition)
void	<a href="#">setLock</a> (Object lock)
void	<a href="#">setMediaList</a> ( <a href="#">MediaList</a> mediaList)
String	<a href="#">toSMILString</a> ()
String	<a href="#">toString</a> ()

**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(<a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> 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	<a href="#"><u>onModuleLoad</u></a> (ModuleItem item) Called when module loaded
void	<a href="#"><u>onModuleUnload</u></a> (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	<a href="#">onAppStart</a> ( <a href="#">IApplicationInstance</a> appInstance) Invoked when an application instance is started.
void	<a href="#">onAppStop</a> ( <a href="#">IApplicationInstance</a> 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	<a href="#"><code>onConnect</code></a> ( <a href="#"><code>IClient</code></a> client, com.wowza.wms.request.RequestFunction function, <a href="#"><code>AMFDataList</code></a> params) Invoked when a client connection is initiated.
void	<a href="#"><code>onConnectAccept</code></a> ( <a href="#"><code>IClient</code></a> client) Invoked when a client connection is accepted.
void	<a href="#"><code>onConnectReject</code></a> ( <a href="#"><code>IClient</code></a> client) Invoked when a client connection is rejected.
void	<a href="#"><code>onDisconnect</code></a> ( <a href="#"><code>IClient</code></a> 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	<a href="#"><u>KEYDATA_MODE_ENCRYPT</u></a> Value: <b>3</b>
public static final	<a href="#"><u>KEYDATA_MODE_INIT</u></a> Value: <b>1</b>
public static final	<a href="#"><u>KEYDATA_MODE_PLAYLIST</u></a> Value: <b>2</b>

### Method Summary

void	<a href="#"><u>onHTTPCupertinoEncryptionKeyCreateLive</u></a> ( <a href="#"><u>IApplicationInstance</u></a> appInstance, String streamName, byte[] encKey) Called when live stream key is requested (per-published stream)
void	<a href="#"><u>onHTTPCupertinoEncryptionKeyCreateVOD</u></a> ( <a href="#"><u>HTTPStreamerSessionCupertino</u></a> httpSession, byte[] encKey) Called when video on demand key is requested (per-session)
void	<a href="#"><u>onHTTPCupertinoEncryptionKeyData</u></a> ( <a href="#"><u>HTTPStreamerSessionCupertino</u></a> httpSession, <a href="#"><u>IHTTPRequest</u></a> req, <a href="#"><u>IHTTPResponse</u></a> resp, byte[] encKeyData) Called when a key data is requested.
void	<a href="#"><u>onHTTPCupertinoEncryptionKeyLiveChunk</u></a> ( <a href="#"><u>ILiveStreamPacketizer</u></a> liveStreamPacketizer, String streamName, <a href="#"><u>CupertinoEncInfo</u></a> encInfo, long chunkId, int mode) Called when live stream key is requested (per-published stream, per-chunk - for rotating keys)
void	<a href="#"><u>onHTTPCupertinoEncryptionKeyRequest</u></a> ( <a href="#"><u>HTTPStreamerSessionCupertino</u></a> httpSession, <a href="#"><u>IHTTPRequest</u></a> req, <a href="#"><u>IHTTPResponse</u></a> resp) Called when a key is requested.
void	<a href="#"><u>onHTTPCupertinoEncryptionKeyVODChunk</u></a> ( <a href="#"><u>HTTPStreamerSessionCupertino</u></a> httpSession, <a href="#"><u>IHTTPStreamerCupertinoIndex</u></a> index, <a href="#"><u>CupertinoEncInfo</u></a> 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	<a href="#"><code>onHTTPCupertinoStreamingSessionCreate</code></a> ( <a href="#"><code>HTTPStreamerSessionCupertino</code></a> <a href="#"><code>httpCupertinoStreamingSession</code></a> ) Invoked when an HTTP Cupertino Streaming session is created.
void	<a href="#"><code>onHTTPCupertinoStreamingSessionDestroy</code></a> ( <a href="#"><code>HTTPStreamerSessionCupertino</code></a> <a href="#"><code>httpCupertinoStreamingSession</code></a> ) 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	<a href="#">onHTTPSanJoseStreamingSessionCreate(HTTPStreamerSessionSanJose httpSanJoseStreamingSession)</a> Invoked when an HTTP SanJose Streaming session is created.
void	<a href="#">onHTTPSanJoseStreamingSessionDestroy(HTTPStreamerSessionSanJose httpSanJoseStreamingSession)</a> 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	<a href="#">onHTTPSessionCreate</a> ( <a href="#">IHTTPStreamerSession</a> httpSession) Invoked when an HTTP Session is created (both Smooth and Cupertino sessions).
void	<a href="#">onHTTPSessionDestroy</a> ( <a href="#">IHTTPStreamerSession</a> 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	<a href="#">onHTTPSmoothStreamingPlayReadyCreateLive</a> ( <a href="#">IApplicationInstance</a> appInstance, String streamName, com.wowza.wms.drm.playready.PlayReadyKeyInfo playReadyKeyInfo)
void	<a href="#">onHTTPSmoothStreamingPlayReadyCreateVOD</a> ( <a href="#">HTTPStreamerSessionSmoothStreamer</a> 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	<a href="#">onHTTPSmoothStreamingSessionCreate(HTTPStreamerSessionSmoothStreamer httpSmoothStreamingSession)</a> Invoked when an HTTP Smooth Streaming session is created.
void	<a href="#">onHTTPSmoothStreamingSessionDestroy(HTTPStreamerSessionSmoothStreamer httpSmoothStreamingSession)</a> 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	<a href="#">onRTPSessionCreate(RTPSession rtpSession)</a> Invoked when an RTP Session is created.
void	<a href="#">onRTPSessionDestroy(RTPSession rtpSession)</a> 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	<a href="#">onStreamCreate</a> ( <a href="#">IMediaStream</a> stream) Invoked when a stream is created.
void	<a href="#">onStreamDestroy</a> ( <a href="#">IMediaStream</a> 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	<a href="#"><code>onResult(IClient client, long pingTime, int pingId, boolean result)</code></a> 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	<a href="#">CALLBACK_PARAM1</a> Callback param: param1 Value: <b>1</b>
public static final	<a href="#">CALLBACK_PARAM10</a> Callback param: param10 Value: <b>10</b>
public static final	<a href="#">CALLBACK_PARAM2</a> Callback param: param2 Value: <b>2</b>
public static final	<a href="#">CALLBACK_PARAM3</a> Callback param: param3 Value: <b>3</b>
public static final	<a href="#">CALLBACK_PARAM4</a> Callback param: param4 Value: <b>4</b>
public static final	<a href="#">CALLBACK_PARAM5</a> Callback param: param5 Value: <b>5</b>
public static final	<a href="#">CALLBACK_PARAM6</a> Callback param: param6 Value: <b>6</b>
public static final	<a href="#">CALLBACK_PARAM7</a> Callback param: param7 Value: <b>7</b>

public static final	<a href="#">CALLBACK_PARAM8</a> Callback param: param8 Value: <b>8</b>
public static final	<a href="#">CALLBACK_PARAM9</a> Callback param: param9 Value: <b>9</b>
public static final	<a href="#">PARAM1</a> Method param: param1 Value: <b>3</b>
public static final	<a href="#">PARAM10</a> Method param: param10 Value: <b>12</b>
public static final	<a href="#">PARAM2</a> Method param: param2 Value: <b>4</b>
public static final	<a href="#">PARAM3</a> Method param: param3 Value: <b>5</b>
public static final	<a href="#">PARAM4</a> Method param: param4 Value: <b>6</b>
public static final	<a href="#">PARAM5</a> Method param: param5 Value: <b>7</b>
public static final	<a href="#">PARAM6</a> Method param: param6 Value: <b>8</b>
public static final	<a href="#">PARAM7</a> Method param: param7 Value: <b>9</b>
public static final	<a href="#">PARAM8</a> Method param: param8 Value: <b>10</b>
public static final	<a href="#">PARAM9</a> Method param: param9 Value: <b>11</b>
public static final	<a href="#">PARAMMETHODNAME</a> Method: method name Value: <b>0</b>
public static final	<a href="#">PLAYTRANSITION_APPEND</a> Value: <b>0</b>

public static final	<a href="#"><u>PLAYTRANSITION_APPEND_IMMEDIATE</u></a> Value: <b>2</b>
public static final	<a href="#"><u>PLAYTRANSITION_RESET</u></a> Value: <b>1</b>
public static final	<a href="#"><u>PLAYTRANSITION_RESET_IMMEDIATE</u></a> Value: <b>3</b>
public static final	<a href="#"><u>PLAYTRANSITION_STOP</u></a> Value: <b>10</b>
public static final	<a href="#"><u>PLAYTRANSITION_SWAP</u></a> Value: <b>12</b>
public static final	<a href="#"><u>PLAYTRANSITION_SWITCH</u></a> Value: <b>13</b>
public static final	<a href="#"><u>PLAYTRANSITION_UNKNOWN</u></a> Value: <b>14</b>
public static final	<a href="#"><u>PLAYTRANSITIONSTR_APPEND</u></a> Play2 transition: APPEND Value: <b>append</b>
public static final	<a href="#"><u>PLAYTRANSITIONSTR_RESET</u></a> Play2 transition: RESET Value: <b>reset</b>
public static final	<a href="#"><u>PLAYTRANSITIONSTR_STOP</u></a> Play2 transition: STOP Value: <b>stop</b>
public static final	<a href="#"><u>PLAYTRANSITIONSTR_SWAP</u></a> Play2 transition: SWAP Value: <b>swap</b>
public static final	<a href="#"><u>PLAYTRANSITIONSTR_SWITCH</u></a> Play2 transition: SWITCH Value: <b>switch</b>
public static final	<a href="#"><u>PLAYTRANSITIONSTR_UNKNOWN</u></a> Play2 transition: UNKNOWN Value: <b>unknown</b>

## Constructor Summary

public	<a href="#"><u>ModuleBase</u></a> ( )
--------	---------------------------------------

## Method Summary

static <a href="#">IApplicationInstance</a>	<a href="#">getAppInstance</a> ( <a href="#">IClient</a> client) Get applicationInstace of a client.
static <a href="#">IApplication</a>	<a href="#">getApplication</a> ( <a href="#">IClient</a> client) Get application of a client.
static int	<a href="#">getCallbackParamCount</a> ( <a href="#">AMFDataList</a> params) Get the total number of parameters passed to callback.
static <a href="#">WMSLogger</a>	<a href="#">getLogger</a> () Get the logging interface.
static <a href="#">AMFData</a>	<a href="#">getParam</a> ( <a href="#">AMFDataList</a> params, int index) Get parameter by index.
static boolean	<a href="#">getParamBoolean</a> ( <a href="#">AMFDataList</a> params, int index) Get parameter by index, Return as boolean.
static boolean	<a href="#">getParamBoolean</a> ( <a href="#">AMFDataList</a> params, int index, boolean defaultVal) Get parameter by index, Return as boolean.
static int	<a href="#">getParamCount</a> ( <a href="#">AMFDataList</a> params) Get the total number of parameters passed to method.
static java.util.Date	<a href="#">getParamDate</a> ( <a href="#">AMFDataList</a> params, int index) Get parameter by index, Return as Date.
static double	<a href="#">getParamDouble</a> ( <a href="#">AMFDataList</a> params, int index) Get parameter by index, Return as double.
static double	<a href="#">getParamDouble</a> ( <a href="#">AMFDataList</a> params, int index, double defaultVal) Get parameter by index, Return as double.
static int	<a href="#">getParamInt</a> ( <a href="#">AMFDataList</a> params, int index) Get parameter by index, Return as int.
static int	<a href="#">getParamInt</a> ( <a href="#">AMFDataList</a> params, int index, int defaultVal) Get parameter by index, Return as int.
static long	<a href="#">getParamLong</a> ( <a href="#">AMFDataList</a> params, int index) Get parameter by index, Return as long.
static long	<a href="#">getParamLong</a> ( <a href="#">AMFDataList</a> params, int index, long defaultVal) Get parameter by index, Return as long.
static <a href="#">AMFDataMixedArray</a>	<a href="#">getParamMixedArray</a> ( <a href="#">AMFDataList</a> params, int index) Get parameter by index, Return as AMFDataMixedArray.
static <a href="#">AMFDataObj</a>	<a href="#">getParamObj</a> ( <a href="#">AMFDataList</a> params, int index) Get parameter by index, Return as Object.
static String	<a href="#">getParamString</a> ( <a href="#">AMFDataList</a> params, int index) Get parameter by index, Return as String.
static String	<a href="#">getParamString</a> ( <a href="#">AMFDataList</a> params, int index, String defaultVal) Get parameter by index, Return as String.
static int	<a href="#">getParamType</a> ( <a href="#">AMFDataList</a> params, int index) Get parameter type.

static <a href="#">IMediaStream</a>	<a href="#">getStream</a> ( <a href="#">IClient</a> 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 <a href="#">IVHost</a>	<a href="#">getVHost</a> ( <a href="#">IClient</a> client) Get vHost of a client.
void	<a href="#">invokePrevious</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params) Wowza Pro will determine command priority based on the order of the modules in the module list.
static void	<a href="#">invokePrevious</a> (Object instance, <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params) Wowza Pro will determine command priority based on the order of the modules in the module list.
static boolean	<a href="#">isSendResult</a> ( <a href="#">AMFDataList</a> params) Is this method call expecting sendResult to be called.
static void	<a href="#">sendClientOnStatusError</a> ( <a href="#">IClient</a> client, String code, String description) Send an error message to the client-side client.onStatus handler
static boolean	<a href="#">sendResult</a> ( <a href="#">IClient</a> client, <a href="#">AMFDataList</a> params, <a href="#">AMFData</a> data) Send a result to client method call as a AMFData object.
static boolean	<a href="#">sendResult</a> ( <a href="#">IClient</a> client, <a href="#">AMFDataList</a> params, boolean value) Send a result to client method call as a single boolean value.
static boolean	<a href="#">sendResult</a> ( <a href="#">IClient</a> client, <a href="#">AMFDataList</a> params, double value) Send a result to client method call as a single double value.
static boolean	<a href="#">sendResult</a> ( <a href="#">IClient</a> client, <a href="#">AMFDataList</a> params, int value) Send a result to client method call as a single int value.
static boolean	<a href="#">sendResult</a> ( <a href="#">IClient</a> client, <a href="#">AMFDataList</a> params, String value) Send a result to client method call as a single String value.
static void	<a href="#">sendStreamOnStatusError</a> ( <a href="#">IMediaStream</a> 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

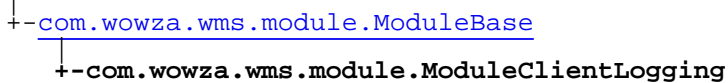
### Parameters:

(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	<a href="#">ModuleClientLogging()</a>
--------	---------------------------------------

### Method Summary

static void	<a href="#">logDebug</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params) Send a debug message to the logging system NetConnection.call("logDebug", null, message);
static void	<a href="#">logError</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params) Send a error message to the logging system NetConnection.call("logError", null, message);
static void	<a href="#">logInfo</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params) Send a info message to the logging system NetConnection.call("logInfo", null, message);
static void	<a href="#">logWarn</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> 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	<a href="#">ModuleCore()</a>
--------	------------------------------

## Method Summary

static void	<a href="#">closeStream</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params) Server side implementation of NetStream.close();
static void	<a href="#">createStream</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params) Create new server side NetStream object (internal to Flash workings).
static void	<a href="#">deleteStream</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params) Delete server side stream object (internal to Flash workings).
static void	<a href="#">FCPublish</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params) FCPublish method called by FME 2.5
static void	<a href="#">FCSubscribe</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params) FCSubscribe subscribes to a live stream (if origin edge will start the stream from the origin to the edge)
static void	<a href="#">FCUnpublish</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params)
static void	<a href="#">FCUnPublish</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params) FCUnpublish method called by FME 2.5
static void	<a href="#">FCUnsubscribe</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params) FCUnSubscribe to a live stream
static void	<a href="#">FCUnSubscribe</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params)
static void	<a href="#">getClientID</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params) Get the clientId for a client connection NetConnection.call("getClientID", resultObj);
static void	<a href="#">getLastStreamId</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params) Get the id for the last created stream NetConnection.call("getLastStreamId", resultObj);
static void	<a href="#">getLiveStreamPacketizer</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params) Get the live stream packetizer list for a client connection NetConnection.call("getLiveStreamPacketizer", resultObj);
static void	<a href="#">getPageUrl</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params) getPageUrl returns the pageUrl from the onConnect metadata
static void	<a href="#">getReferrer</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params) getReferrer returns the referrer from the onConnect metadata



static void	<a href="#">getRepeaterOriginUrl</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params) Get the Repeater Origin URL for this client
void	<a href="#">getStreamBitrate</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params) Get stream bitrate in bits-per-second for a video on demand stream (will not work for live).
static void	<a href="#">getStreamLength</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> 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	<a href="#">getStreamType</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params) Get the default stream type for a client connection NetConnection.call("getStreamType", resultObj);
static void	<a href="#">getVersion</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params) Get the Wowza Pro server version and build number NetConnection.call("getVersion", resultObj);
static void	<a href="#">initLiveStreamRepeating</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params) Initialize a stream for live stream repeating
static void	<a href="#">initStream</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params) Create new server side NetStream object (internal to Flash workings).
static void	<a href="#">pause</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params) Server side implementation of NetStream.pause([ flag : Boolean]);
static void	<a href="#">pauseRaw</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params) pauseRaw method introduced in Flash player 10
static void	<a href="#">play</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params) Server side implementation of NetStream.play(name : Object [,start : Number[, len : Number[, reset : Object]]]);
static void	<a href="#">play2</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params) Server side implementation of NetStream.play(playOptions : NetStreamPlayOptions);
static void	<a href="#">publish</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params) Server side implementation of NetStream.publish(name : String [, howToPublish : String]);
static void	<a href="#">receiveAudio</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params) Server side implementation of NetStream.receiveAudio(receive : Boolean);
static void	<a href="#">receiveVideo</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params) Server side implementation of NetStream.receiveVideo(receive : Boolean   FPS : Number); FPS does not work the same as FMS.
static void	<a href="#">releaseStream</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params)

static void	<a href="#">seek</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params) Server side implementation of NetStream.seek(offset : Number);
static void	<a href="#">setBandwidthLimit</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params)
static void	<a href="#">setBufferTime</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params) Server side implementation of NetStream.setBufferTime(bufferTime : Number);
static void	<a href="#">setLiveStreamPacketizer</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params) Set the live stream packetizer for a stream
static void	<a href="#">setRepeaterOriginUrl</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params) Set the Repeater Origin URL for this client
static void	<a href="#">setStreamType</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> 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)
```



(continued from last page)

## ModuleFastPlay

```
public ModuleFastPlay()
```

## Methods

### setFastPlay

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

Turn on fast play `NetStream.call("setFastPlay", null, multiplier, frames-per-second, direction);`

#### Parameters:

client - client

function - function

params - params (multiplier, frames-per-second, direction)

## com.wowza.wms.module Class ModuleMediaCasterStreamMonitorAdvanced

```
java.lang.Object
|
+-com.wowza.wms.module.ModuleBase
|
+-com.wowza.wms.module.ModuleMediaCasterStreamMonitorAdvanced
```

### All Implemented Interfaces:

[IMediaCasterValidateMediaCaster](#)

---

```
public class ModuleMediaCasterStreamMonitorAdvanced
extends ModuleBase
implements IMediaCasterValidateMediaCaster
```

ModuleMediaCasterStreamMonitorAdvanced: Advanced stream monitoring module.

Add this <Module> definition to the end of the <Modules> list in [install-dir]/conf/[application]/Application.xml:

```
<Module>
  <Name>ModuleMediaCasterStreamMonitorAdvanced</Name>
  <Description>ModuleMediaCasterStreamMonitorAdvanced</Description>
  <Class>com.wowza.wms.module.ModuleMediaCasterStreamMonitorAdvanced</Class>
</Module>
```

Add these properties to the application level <Properties> container at the bottom of [install-dir]/conf/[application]/Application.xml (be sure to get the correct <Properties> container - there are several in the Application.xml file). Carefully read the instructions above each set of properties and set accordingly:

```

<!--
Monitor incoming packets (stream, audio, video) to be sure packets continue to flow from
encoder to stream. The
streamMonitor[type]StartTimeout controls the timeout (milliseconds) for the first packet. The
streamMonitor[type]Timeout
controls the timeout (milliseconds) for packets after the first packet. The stream type
refers to a catch all of any packet of
any type (audio, video, data). If any of these values are set to zero, the test is turned
off.
-->
<Property>
    <Name>streamMonitorStreamStartTimeout</Name>
    <Value>20000</Value>
    <Type>Integer</Type>
</Property>
<Property>
    <Name>streamMonitorStreamTimeout</Name>
    <Value>12000</Value>
    <Type>Integer</Type>
</Property>
<Property>
    <Name>streamMonitorVideoStartTimeout</Name>
    <Value>0</Value>
    <Type>Integer</Type>
</Property>
<Property>
    <Name>streamMonitorVideoTimeout</Name>
    <Value>0</Value>
    <Type>Integer</Type>
</Property>
<Property>
    <Name>streamMonitorAudioStartTimeout</Name>
    <Value>0</Value>
    <Type>Integer</Type>
</Property>
<Property>
    <Name>streamMonitorAudioTimeout</Name>
    <Value>0</Value>
    <Type>Integer</Type>
</Property>

<!--
Monitor the incoming packet timecodes (audio, video or data) to be sure packets do not arrive
out of order or late. The
streamMonitor[type]TCPosTolerance and streamMonitor[type]TCNegTolerance (milliseconds) values
define a sliding window
based on the timecode of the previous packet. For example if these values are set to -500 and
3000 respectively then the
timecode difference between the current arriving packet and the previous packet of that type
must fall within
-500 and 3000 milliseconds. If not and streamMonitor[type]TCToleranceEnable is set to true

```

then stream will be considered  
unhealthy and will be reset.

-->

```
<Property>
    <Name>streamMonitorVideoTCToleranceEnable</Name>
    <Value>>false</Value>
    <Type>Boolean</Type>
</Property>
<Property>
    <Name>streamMonitorVideoTCPosTolerance</Name>
    <Value>3000</Value>
    <Type>Integer</Type>
</Property>
<Property>
    <Name>streamMonitorVideoTCNegTolerance</Name>
    <Value>-500</Value>
    <Type>Integer</Type>
</Property>
<Property>
    <Name>streamMonitorAudioTCToleranceEnable</Name>
    <Value>>false</Value>
    <Type>Boolean</Type>
</Property>
<Property>
    <Name>streamMonitorAudioTCPosTolerance</Name>
    <Value>3000</Value>
    <Type>Integer</Type>
</Property>
<Property>
    <Name>streamMonitorAudioTCNegTolerance</Name>
    <Value>-500</Value>
    <Type>Integer</Type>
</Property>
<Property>
    <Name>streamMonitorDataTCToleranceEnable</Name>
    <Value>>false</Value>
    <Type>Boolean</Type>
</Property>
<Property>
    <Name>streamMonitorDataTCPosTolerance</Name>
    <Value>3000</Value>
    <Type>Integer</Type>
</Property>
<Property>
    <Name>streamMonitorDataTCNegTolerance</Name>
    <Value>-500</Value>
    <Type>Integer</Type>
</Property>
```

<!--

Monitors the time difference between the audio and video channel of a stream. If the difference between the currently arriving video packet and the previous audio packet (or vice-versa) is greater than streamMonitorAVSyncTolerance and streamMonitorAVSyncToleranceEnable is set to true, then the stream will be considered unhealthy and will be reset.

-->

<Property>

<Name>streamMonitorAVSyncToleranceEnable</Name>

<Value>>false</Value>

<Type>Boolean</Type>

</Property>

<Property>

<Name>streamMonitorAVSyncTolerance</Name>

<Value>1500</Value>

<Type>Integer</Type>

</Property>

<!--

If set to true, then when a stream is reset and it belong to a MediaStreamNameGroup all streams in the group will be reset. If false only the unhealthy stream will be reset.

-->

<Property>

<Name>streamMonitorResetNameGroups</Name>

<Value>>true</Value>

<Type>Boolean</Type>

</Property>

<!--

Turns on debug logging of the monitoring.

-->

<Property>

<Name>streamMonitorDebug</Name>

<Value>>false</Value>

<Type>Boolean</Type>

</Property>

## Field Summary

protected	<a href="#">appInstance</a>
protected	<a href="#">badStreams</a>
protected	<a href="#">monitors</a>
protected	<a href="#">streamMonitorAudioStartTimeout</a>



protected	<a href="#">streamMonitorAudioTCNegTolerance</a>
protected	<a href="#">streamMonitorAudioTCPosTolerance</a>
protected	<a href="#">streamMonitorAudioTCToleranceEnable</a>
protected	<a href="#">streamMonitorAudioTimeout</a>
protected	<a href="#">streamMonitorAVSyncTolerance</a>
protected	<a href="#">streamMonitorAVSyncToleranceEnable</a>
protected	<a href="#">streamMonitorDataTCNegTolerance</a>
protected	<a href="#">streamMonitorDataTCPosTolerance</a>
protected	<a href="#">streamMonitorDataTCToleranceEnable</a>
protected	<a href="#">streamMonitorDebug</a>
protected	<a href="#">streamMonitorResetNameGroups</a>
protected	<a href="#">streamMonitorStreamStartTimeout</a>
protected	<a href="#">streamMonitorStreamTimeout</a>
protected	<a href="#">streamMonitorVideoStartTimeout</a>
protected	<a href="#">streamMonitorVideoTCNegTolerance</a>
protected	<a href="#">streamMonitorVideoTCPosTolerance</a>
protected	<a href="#">streamMonitorVideoTCToleranceEnable</a>
protected	<a href="#">streamMonitorVideoTimeout</a>

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	<a href="#">ModuleMediaCasterStreamMonitorAdvanced()</a>
--------	--

## Method Summary

void	<a href="#">onAppStart</a> ( <a href="#">IApplicationInstance</a> appInstance)
void	<a href="#">onAppStop</a> ( <a href="#">IApplicationInstance</a> appInstance)
boolean	<a href="#">onResetMediaCaster</a> ( <a href="#">IApplicationInstance</a> appInstance, <a href="#">IMediaCaster</a> mediaCaster)
void	<a href="#">onStreamCreate</a> ( <a href="#">IMediaStream</a> stream)
void	<a href="#">onStreamDestroy</a> ( <a href="#">IMediaStream</a> stream)
boolean	<a href="#">onValidateMediaCaster</a> ( <a href="#">IApplicationInstance</a> appInstance, <a href="#">IMediaCaster</a> mediaCaster)
void	<a href="#">onValidateMediaCastersStart</a> ( <a href="#">IApplicationInstance</a> appInstance)
void	<a href="#">onValidateMediaCastersStop</a> ( <a href="#">IApplicationInstance</a> 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)

---

**streamMonitorAVSyncToleranceEnable**protected boolean **streamMonitorAVSyncToleranceEnable**

---

---

**streamMonitorAVSyncTolerance**protected int **streamMonitorAVSyncTolerance**

---

---

**streamMonitorVideoStartTimeout**protected int **streamMonitorVideoStartTimeout**

---

---

**streamMonitorVideoTimeout**protected int **streamMonitorVideoTimeout**

---

---

**streamMonitorAudioStartTimeout**protected int **streamMonitorAudioStartTimeout**

---

---

**streamMonitorAudioTimeout**protected int **streamMonitorAudioTimeout**

---

---

**streamMonitorResetNameGroups**protected boolean **streamMonitorResetNameGroups**

---

---

**streamMonitorDebug**protected boolean **streamMonitorDebug**

---

---

**appInstance**protected 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)
```

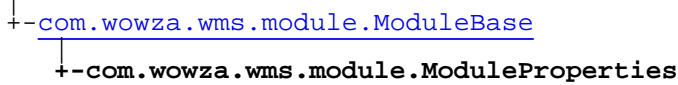
---

## 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	<a href="#">ModuleProperties</a> ()
--------	-------------------------------------

### Method Summary

static void	<a href="#">getAppInstanceProperty</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params) Get application instance level property value NetConnection.call("getAppInstanceProperty", resultObj, name);
static void	<a href="#">getApplicationProperty</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params) Get application level property value NetConnection.call("getApplicationProperty", resultObj, name);
static void	<a href="#">getClientProperty</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params) Get client level property value NetConnection.call("getClientProperty", resultObj, name);
static void	<a href="#">getStreamProperty</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params) Get stream level property value NetConnection.call("getStreamProperty", resultObj, streamId, name);
static void	<a href="#">setAppInstanceProperty</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params) Set application instance level property NetConnection.call("setAppInstanceProperty", null, name, value);
static void	<a href="#">setApplicationProperty</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params) Set application level property NetConnection.call("setApplicationProperty", null, name, value);

static void	<a href="#">setClientProperty</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params) Set client level property NetConnection.call("setClientProperty", null, name, value);
static void	<a href="#">setStreamProperty</a> ( <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> 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	<a href="#">doBreak</a>
--------	-------------------------

### Constructor Summary

public	<a href="#">ResponseFunction</a> ( <a href="#">IClient</a> client) Create an empty ResponseFunction
public	<a href="#">ResponseFunction</a> (com.wowza.wms.netconnection.INetConnection netConnection) Create an empty ResponseFunction
public	<a href="#">ResponseFunction</a> (com.wowza.wms.netconnection.INetConnection netConnection, <a href="#">AMFObj</a> amfObj) Create an empty ResponseFunction
public	<a href="#">ResponseFunction</a> ( <a href="#">IClient</a> client, <a href="#">AMFObj</a> amfObj) Create an empty ResponseFunction
public	<a href="#">ResponseFunction</a> ( <a href="#">IMediaStream</a> stream, <a href="#">AMFObj</a> amfObj) Create an empty ResponseFunction

### Method Summary

void	<a href="#">addBody</a> ( <a href="#">AMFData</a> body) Add AMFData to the body of the message.
void	<a href="#">addBytes</a> (byte[] inbytes) Add raw AMFData bytes[] to message body
void	<a href="#">createBroadcastMessage</a> (java.nio.ByteBuffer messageBytes) Create broadcast message.
void	<a href="#">createConnectMessage</a> (String inName, double inResultNum) Create net connection connect message.
void	<a href="#">createDefaultMessage</a> (String inName, double inResultNum) Create a default message (onStatus type messages).
void	<a href="#">createEnhancedSeekMessage</a> (byte[] messageBuffer, int src, int tc)

void	<a href="#"><u>createPlayStatusMessage</u></a> (String inName) Create onPlayStatus message.
void	<a href="#"><u>createSeekMessage</u></a> (String inName) Create a seek result message.
void	<a href="#"><u>createSOMessage</u></a> (byte[] messageBuffer, int objectEncoding) Create shared object message.
long	<a href="#"><u>getTimecode</u></a> () Get function timecode (milliseconds).
int	<a href="#"><u>getType</u></a> () Get message type.
boolean	<a href="#"><u>isForceAMF0</u></a> ()
void	<a href="#"><u>setBody</u></a> (int index, <a href="#"><u>AMFData</u></a> body) Add AMFData to the body of the message.
void	<a href="#"><u>setForceAMF0</u></a> (boolean forceAMF0)
void	<a href="#"><u>setMessageBytes</u></a> (byte[] messageBytes)
void	<a href="#"><u>setRetAMFNumber</u></a> (int innum) Set the return channel id
void	<a href="#"><u>setSrc</u></a> (int src) Set the src (stream id) for the message.
void	<a href="#"><u>setTimecode</u></a> (long timecode) Set function timecode (milliseconds).
void	<a href="#"><u>setType</u></a> (int type) Set message type.
int	<a href="#"><u>write</u></a> (java.io.OutputStream out, boolean isAbsTimecode, int chunkSize) Write message directly to OutputStream
int	<a href="#"><u>write</u></a> (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	<a href="#">ResponseFunctions()</a>
--------	-------------------------------------

### Method Summary

void	<a href="#">add(ResponseFunction wmsResponseFunction)</a> Add a function.
void	<a href="#">clear()</a>
boolean	<a href="#">isPending()</a> Are there any pending items in the list.
int	<a href="#">output(java.io.OutputStream out, int sendChunkSize)</a> 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	<a href="#">handleMessage</a> (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[]	<a href="#">getMetadataPacket</a> ( <a href="#">RTPStream</a> 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	<a href="#">onRTPSessionCreate</a> ( <a href="#">RTPSession</a> rtpSession) Invoked when RTP session is created
void	<a href="#">onRTPSessionDestroy</a> ( <a href="#">RTPSession</a> 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	<a href="#">onAnnounce(RTPSession rtspSession, com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp)</a> Invoked by ANNOUNCE command
void	<a href="#">onDescribe(RTPSession rtspSession, com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp)</a> Invoked by DESCRIBE command
void	<a href="#">onGetParameter(RTPSession rtspSession, com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp)</a> Invoked by GET_PARAMETER command
void	<a href="#">onOptions(RTPSession rtspSession, com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp)</a> Invoked by OPTIONS command
void	<a href="#">onPause(RTPSession rtspSession, com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp)</a> Invoked by PAUSE command
void	<a href="#">onPlay(RTPSession rtspSession, com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp)</a> Invoked by PLAY command
void	<a href="#">onRecord(RTPSession rtspSession, com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp)</a> Invoked by RECORD command
void	<a href="#">onRedirect(RTPSession rtspSession, com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp)</a> Invoked by REDIRECT command
void	<a href="#">onSetParameter(RTPSession rtspSession, com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp)</a> Invoked by SET_PARAMETER command
void	<a href="#">onSetup(RTPSession rtspSession, com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp)</a> Invoked by SETUP command
void	<a href="#">onTeardown(RTPSession rtspSession, com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp)</a> 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	<a href="#">RTPContext</a> ( <a href="#">IVHost</a> vhost) Constructor
--------	---

#### Method Summary

boolean	<a href="#">acquireSocketAddress</a> (java.net.SocketAddress socketAddress) Acquire UDP socket address
int	<a href="#">acquireUDPPortPair</a> () Acquire next available UDP port pair
int	<a href="#">acquireUDPPortPair</a> (int port) Acquire UDP port pair
void	<a href="#">cacheRTPStream</a> ( <a href="#">RTPStream</a> stream) Cache an RTP stream, Internal use.
void	<a href="#">doWatchdog</a> () Idle events for cleanup
boolean	<a href="#">existsRTSPTunnelingSession</a> (String sessionId) Return true if session id is valid RTSP/RTP tunneling session id
int[]	<a href="#">expandToPortPair</a> (int port) Expand a single port to a pair.
RTPPacketizerItem	<a href="#">getAudioPacketizerItem</a> ( <a href="#">IApplicationInstance</a> appInstance, int codecId) Get audio packetizer for a given codec id.
com.wowza.wms.rtp.dep acketizer.RTPDePacket izerList	<a href="#">getDePacketizerList</a> () Get a list of the available depacketizers
String	<a href="#">getDePacketizerName</a> (RTPTrack rtpTrack) Get a depacketizer by name
Object	<a href="#">getLock</a> () Get the UDP port manager lock
Object	<a href="#">getRTSPTunnelingLock</a> () Get the RTSP/RTP tunneling lock

com.wowza.wms.rtsp.RTSP TunnelingSession	<a href="#"><u>getRTSPTunnelingSession</u></a> (String sessionId) Get RTSP/RTP tunneling session by session id
<a href="#"><u>RTPSessions</u></a>	<a href="#"><u>getSessions</u></a> ( ) Get RTP sessions
RTPPacketizerItem	<a href="#"><u>getStreamPacketizerItem</u></a> ( <a href="#"><u>IApplicationInstance</u></a> appInstance, int codecId) Get stream packetizer for a given codec id.
com.wowza.wms.rtp.transport.UDPTransportManager	<a href="#"><u>getUDPTransportManager</u></a> ( ) Get the UDP transport manager
<a href="#"><u>IVHost</u></a>	<a href="#"><u>getVHost</u></a> ( ) Get vhost
RTPPacketizerItem	<a href="#"><u>getVideoPacketizerItem</u></a> ( <a href="#"><u>IApplicationInstance</u></a> appInstance, int codecId) Get video packetizer for a given codec id.
void	<a href="#"><u>init</u></a> ( ) Initialize
void	<a href="#"><u>putAudioPacketizerItem</u></a> (int codecId, RTPPacketizerItem rtpPacketizerInfo) Set the audio packetizer for a given codec id
void	<a href="#"><u>putRTSPTunnelingSession</u></a> (String sessionId, com.wowza.wms.rtsp.RTSP TunnelingSession rtspTunnelingSession) Remove RTSP/RTP tunneling session by session id
void	<a href="#"><u>putStreamPacketizerItem</u></a> (int codecId, RTPPacketizerItem rtpPacketizerInfo) Set the stream packetizer for a given codec id
void	<a href="#"><u>putVideoPacketizerItem</u></a> (int codecId, RTPPacketizerItem rtpPacketizerInfo) Set the video packetizer for a given codec id
boolean	<a href="#"><u>releaseSocketAddress</u></a> ( java.net.SocketAddress socketAddress) Release UDP socket address
void	<a href="#"><u>releaseUDPPortPair</u></a> (int port) Release port pair
com.wowza.wms.rtsp.RTSP TunnelingSession	<a href="#"><u>removeRTSPTunnelingSession</u></a> (String sessionId) Remove RTSP/RTP tunneling session by session id
void	<a href="#"><u>shutdown</u></a> ( ) Shutdown
void	<a href="#"><u>shutdownRTPSession</u></a> ( <a href="#"><u>RTPSession</u></a> rtpSession) Gracefully and forcefully shutdown and RTP session
<a href="#"><u>RTPStream</u></a>	<a href="#"><u>uncacheRTPStream</u></a> (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	<a href="#">RTPDestination()</a>
--------	----------------------------------

#### Method Summary

String	<a href="#">getAudioHost()</a> Get audio host
int	<a href="#">getAudioPort()</a> Get audio port
String	<a href="#">getHost()</a> Get the host
String	<a href="#">getHostType()</a> Get host type (default IP4)
String	<a href="#">getName()</a> Get name
int	<a href="#">getStreamPort()</a> Get stream port
int	<a href="#">getTTL()</a> Get time to live
String	<a href="#">getVideoHost()</a> Get video host
int	<a href="#">getVideoPort()</a> Get video port
boolean	<a href="#">isMulticast()</a> Return true if multicast destination
boolean	<a href="#">isRTPWrapped()</a> Is stream wrapped in RTP (MPEG-TS in RTP)
boolean	<a href="#">isStream()</a> Return true if stream destination (and not native RTP destination)

void	<a href="#"><code>setAudioHost</code></a> (String audioHost) Set audio host
void	<a href="#"><code>setAudioPort</code></a> (int audioPort) Set audio port
void	<a href="#"><code>setHost</code></a> (String host) Set host
void	<a href="#"><code>setHostType</code></a> (String hostType) Set host type (default IP4)
void	<a href="#"><code>setName</code></a> (String name) Set name
void	<a href="#"><code>setRTPWrapped</code></a> (boolean isRTPWrapped) Is stream wrapped in RTP (MPEG-TS in RTP)
void	<a href="#"><code>setStreamPort</code></a> (int streamPort) Set stream port
void	<a href="#"><code>setTTL</code></a> (int ttl) Set time to live
void	<a href="#"><code>setVideoHost</code></a> (String videoHost) Set video host
void	<a href="#"><code>setVideoPort</code></a> (int videoPort) Set video port
String	<a href="#"><code>toString</code></a> ()

#### 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	<a href="#">RTPPort</a> (String inIpAddress, int inPort, String outIpAddress, int outPort, boolean isMulticast) Constructor
--------	--

### Method Summary

void	<a href="#">bind</a> (int direction) Bind to port
int	<a href="#">getAddressCount</a> () Get address count
<a href="#">I RTPMessageHandler</a>	<a href="#">getHandler</a> () Get the message handler
String	<a href="#">getInIpAddress</a> () Get in IP address
int	<a href="#">getInPort</a> () Get in port
String	<a href="#">getOutIpAddress</a> () Get out IP address
int	<a href="#">getOutPort</a> () Get out port
RTPTrack	<a href="#">getTrack</a> () Get RTP track
int	<a href="#">getTTL</a> () Get time to live (milliseconds)
com.wowza.wms.rtp.transport.IUDPTransport	<a href="#">getUDPTransport</a> () Get the UDP transport
void	<a href="#">handleMessage</a> (java.net.SocketAddress socketAddr, Object message)

boolean	<a href="#"><u>isBlockUDPOut</u></a> ( )
boolean	<a href="#"><u>isInMulticast</u></a> ( ) Is in stream multicast
boolean	<a href="#"><u>isMulticast</u></a> ( ) Is multicast
boolean	<a href="#"><u>isOutMulticast</u></a> ( ) Is out stream multicast
void	<a href="#"><u>sendMessage</u></a> (byte[] message, int offset, int len) Send a message out
void	<a href="#"><u>sendResponse</u></a> (byte[] message) Send a response
void	<a href="#"><u>sendResponse</u></a> (byte[] message, int offset, int len) Send response
void	<a href="#"><u>sendResponse</u></a> (byte[] message, int offset, int len, java.net.SocketAddress destination) Send response to destination
void	<a href="#"><u>sessionClosed</u></a> (com.wowza.wms.rtp.transport.IUDPTransportSession session)
void	<a href="#"><u>sessionOpened</u></a> (com.wowza.wms.rtp.transport.IUDPTransportSession session)
void	<a href="#"><u>setAddressCount</u></a> (int addressCount) Set address count
void	<a href="#"><u>setBlockUDPOut</u></a> (boolean blockUDPOut)
void	<a href="#"><u>setHandler</u></a> ( <a href="#"><u>IRTPMessageHandler</u></a> handler) Set the message handler
void	<a href="#"><u>setTrack</u></a> (RTPTrack track) Set RTP track
void	<a href="#"><u>setTTL</u></a> (int ttl) Set time to live (milliseconds)
void	<a href="#"><u>shutdown</u></a> ( ) Shutdown port
void	<a href="#"><u>unbind</u></a> ( ) 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	<a href="#">RTPPushPublishSession()</a>
--------	---

### Method Summary

<a href="#">RTPSession</a>	<a href="#">getRTPSession()</a> Get RTP session
String	<a href="#">getSDPData()</a> Get SDP data
void	<a href="#">setRTPSession(<a href="#">RTPSession</a> rtpSession)</a> Set RTP session
void	<a href="#">setSDPData(String sdpData)</a> 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	<a href="#">actionListeners</a>
protected	<a href="#">appInstance</a>
protected	<a href="#">appInstanceShutdown</a>
protected	<a href="#">audioPacketizers</a>
protected	<a href="#">authenticatePlayHandler</a>
protected	<a href="#">authenticatePublishHandler</a>
public static final	<a href="#">AUTHMETHOD_PLAY</a> Value: <b>1</b>
public static final	<a href="#">AUTHMETHOD_PUBLISH</a> Value: <b>2</b>
public static final	<a href="#">AUTHMETHOD_UNKNOWN</a> Value: <b>0</b>
protected	<a href="#">connectionHolder</a>
protected	<a href="#">cookieStr</a>
protected	<a href="#">debugRTSPSession</a>
protected	<a href="#">elapsedTime</a>
protected	<a href="#">idleFrequency</a>
protected	<a href="#">idleHandler</a>

protected	<a href="#"><u>ioPerformanceCounter</u></a>
protected	<a href="#"><u>ioSession</u></a>
protected	<a href="#"><u>ip</u></a>
protected	<a href="#"><u>isAnnounce</u></a>
protected	<a href="#"><u>isConnected</u></a>
protected	<a href="#"><u>isDescribe</u></a>
protected	<a href="#"><u>isSessionValid</u></a>
protected	<a href="#"><u>lastAuthenticateMethod</u></a>
protected	<a href="#"><u>lock</u></a>
protected	<a href="#"><u>loggedConnect</u></a>
protected	<a href="#"><u>properties</u></a>
protected	<a href="#"><u>queryStr</u></a>
protected	<a href="#"><u>redirectSession</u></a>
protected	<a href="#"><u>redirectSessionCode</u></a>
protected	<a href="#"><u>redirectSessionMessage</u></a>
protected	<a href="#"><u>redirectSessionURL</u></a>
protected	<a href="#"><u>referrer</u></a>
protected	<a href="#"><u>rtpWriteListener</u></a>
protected	<a href="#"><u>rtspPlayRangeStart</u></a>
protected	<a href="#"><u>rtspPlayRangeStop</u></a>
protected	<a href="#"><u>rtspTunnelingSessionId</u></a>
protected	<a href="#"><u>serverIp</u></a>
protected	<a href="#"><u>serverPort</u></a>
protected	<a href="#"><u>sessionId</u></a>

protected	<a href="#">shutdownClient</a>
protected	<a href="#">streamPacketizers</a>
protected	<a href="#">streams</a>
protected	<a href="#">streamsOrder</a>
protected	<a href="#">timeCreated</a>
protected	<a href="#">totalIOPerformance2Last</a>
protected	<a href="#">totalIOPerformanceLast</a>
protected	<a href="#">uri</a>
protected	<a href="#">userAgent</a>
protected	<a href="#">vhost</a>
protected	<a href="#">videoPacketizers</a>

## Constructor Summary

public	<a href="#">RTPSession</a> (String sessionId) Constructor
--------	--

## Method Summary

void	<a href="#">acceptSession</a> () Accept this session
void	<a href="#">addActionListener</a> ( <a href="#">IRTSPActionNotify</a> actionListener) Add action listener
void	<a href="#">addIOPerformance</a> ( <a href="#">IOPerformanceCounter</a> totalIOPerformanceResult) Internal use
void	<a href="#">addIOPerformance2</a> ( <a href="#">IOPerformanceCounter</a> totalIOPerformanceResult) Internal use
void	<a href="#">addRTSPStream</a> ( <a href="#">RTPStream</a> stream) Add RTP stream
void	<a href="#">clearLoggingValues</a> () Clear logging values, Internal use.
void	<a href="#">doIdle</a> ()
<a href="#">IApplicationInstance</a>	<a href="#">getAppInstance</a> () Get application instance

RTPPacketizerItem	<a href="#">getAudioPacketizerItem(RTPContext rtpContext, IApplicationInstance appInstance, int codecId)</a> Get audio packetizer for a given codec id.
<a href="#">IAuthenticateRTSP</a>	<a href="#">getAuthenticatePlayHandler()</a> Get the authentication play handler
<a href="#">IAuthenticateRTSP</a>	<a href="#">getAuthenticatePublishHandler()</a> Get the RTP authentication handler
ConnectionHolder	<a href="#">getConnectionHolder()</a> Get connection holder, Internal use.
String	<a href="#">getCookieStr()</a> Get cookie string
<a href="#">ElapsedTimer</a>	<a href="#">getElapsedTime()</a> Get the elapsed timer for this RTP session
int	<a href="#">getIdleFrequency()</a> Get idle frequency (milliseconds)
RTPIdleHandler	<a href="#">getIdleHandler()</a> Get idle handler
<a href="#">IOPerformanceCounter</a>	<a href="#">getIOPerformanceCounter()</a> Get IO performance counter
org.apache.mina.common.support.BaseIoSession	<a href="#">getIoSession()</a>
String	<a href="#">getIp()</a> Get remote IP address
int	<a href="#">getLastAuthenticateMethod()</a> Get the last method received
<a href="#">WMSProperties</a>	<a href="#">getProperties()</a> Get properties
String	<a href="#">getQueryStr()</a> Get query string
int	<a href="#">getRedirectSessionCode()</a>
String	<a href="#">getRedirectSessionMessage()</a>
String	<a href="#">getRedirectSessionURL()</a>
String	<a href="#">getReferrer()</a> Get referrer
RTPWriteListener	<a href="#">getRTPWriteListener()</a> Get the RTP write listener for this session
double	<a href="#">getRTSPPlayRangeStart()</a> Get play start range, Internal use.

double	<a href="#"><u>getRTSPPlayRangeStop()</u></a> Get play stop range, Internal use.
<a href="#"><u>RTPStream</u></a>	<a href="#"><u>getRTSPStream()</u></a> Get the default RTP Stream (all RTP sessions have a single RTP Stream)
<a href="#"><u>RTPStream</u></a>	<a href="#"><u>getRTSPStream(String streamId)</u></a> Get RTP Stream
String	<a href="#"><u>getRTSPTunnelingSessionId()</u></a> Get the RTSP/RTP tunneling session id
String	<a href="#"><u>getServerIp()</u></a> Get the server IP address
int	<a href="#"><u>getServerPort()</u></a> Get server port
String	<a href="#"><u>getSessionId()</u></a> Get session id
RTPPacketizerItem	<a href="#"><u>getStreamPacketizerItem(RTPContext rtpContext, IApplicationInstance appInstance, int codecId)</u></a> Get stream packetizer for a given codec id.
String	<a href="#"><u>getTimeRunning()</u></a> Get the time running for this RTP session
double	<a href="#"><u>getTimeRunningSeconds()</u></a> Get the number of second this RTP session has been running
String	<a href="#"><u>getUri()</u></a> Get URI
String	<a href="#"><u>getUserAgent()</u></a> Get user agent
<a href="#"><u>IVHost</u></a>	<a href="#"><u>getVHost()</u></a> Get vhost
RTPPacketizerItem	<a href="#"><u>getVideoPacketizerItem(RTPContext rtpContext, IApplicationInstance appInstance, int codecId)</u></a> Get video packetizer for a given codec id.
boolean	<a href="#"><u>isAnnounce()</u></a> Has ANNOUNCE command been called on this session
boolean	<a href="#"><u>isAnnounceOrDescribe()</u></a> Has announce or described been called
boolean	<a href="#"><u>isConnected()</u></a> Is session connection
boolean	<a href="#"><u>isDebugRTSPSession()</u></a> True if debugging RTSP session
boolean	<a href="#"><u>isDescribe()</u></a> Has DESCRIBE command been called on this session



boolean	<a href="#"><u>isLoggedConnect</u></a> ( ) Is connect logged
boolean	<a href="#"><u>isRedirectSession</u></a> ( )
boolean	<a href="#"><u>isSessionValid</u></a> ( ) Is this session valid
boolean	<a href="#"><u>isShutdownClient</u></a> ( ) Is RTP session shutdown
void	<a href="#"><u>onAnnounce</u></a> (com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp) Invoked on ANNOUNCE command
void	<a href="#"><u>onDescribe</u></a> (com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp) Invoked on DESCRIBE command
void	<a href="#"><u>onGetParameter</u></a> (com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp) Invoked on GET_PARAMETER command
void	<a href="#"><u>onOptions</u></a> (com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp) Invoked on OPTIONS command
void	<a href="#"><u>onPause</u></a> (com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp) Invoked on PAUSE command
void	<a href="#"><u>onPlay</u></a> (com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp) Invoked on PLAY command
void	<a href="#"><u>onRecord</u></a> (com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp) Invoked on RECORD command
void	<a href="#"><u>onRedirect</u></a> (com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp) Invoked on REDIRECT command
void	<a href="#"><u>onSetParameter</u></a> (com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp) Invoked on SET_PARAMETER command
void	<a href="#"><u>onSetup</u></a> (com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp) Invoked on SETUP command
void	<a href="#"><u>onTeardown</u></a> (com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp) Invoked on TEARDOWN command
void	<a href="#"><u>putAudioPacketizerItem</u></a> (int codecId, String classPath) Set the audio packetizer for a given codec id
void	<a href="#"><u>putStreamPacketizerItem</u></a> (int codecId, String classPath) Set the stream packetizer for a given codec id

void	<a href="#"><u>putVideoPacketizerItem</u></a> (int codecId, String classPath) Set the video packetizer for a given codec id
void	<a href="#"><u>redirectSession</u></a> (String redirectSessionURL)
void	<a href="#"><u>redirectSession</u></a> (String redirectSessionURL, int redirectSessionCode)
void	<a href="#"><u>rejectSession</u></a> () Reject this session
void	<a href="#"><u>removeActionListener</u></a> ( <a href="#"><u>IRTSPActionNotify</u></a> actionListener) Remove action listener
<a href="#"><u>RTPStream</u></a>	<a href="#"><u>removeRTSPStream</u></a> (String streamId) Remove RTP stream
void	<a href="#"><u>setAnnounce</u></a> (boolean isAnnounce) Set ANNOUNCE command been called on this session
void	<a href="#"><u>setAppInstance</u></a> ( <a href="#"><u>IApplicationInstance</u></a> appInstance) Set application instance
void	<a href="#"><u>setAuthenticatePlayHandler</u></a> ( <a href="#"><u>IAuthenticateRTSP</u></a> authenticatePlayHandler) Set the authentication play handler
void	<a href="#"><u>setAuthenticatePublishHandler</u></a> ( <a href="#"><u>IAuthenticateRTSP</u></a> authenticatePublishHandler) Set the RTP authentication handler
void	<a href="#"><u>setConnected</u></a> (boolean isConnected) Set session is connected
void	<a href="#"><u>setCookieStr</u></a> (String cookieStr) Set cooking string
void	<a href="#"><u>setDebugRTSPSession</u></a> (boolean debugRTSPSession) Set debugging RTSP session
void	<a href="#"><u>setDescribe</u></a> (boolean isDescribe) Set DESCRIBE command been called on this session
void	<a href="#"><u>setIdleFrequency</u></a> (int idleFrequency) Set idle frequency (milliseconds)
void	<a href="#"><u>setIdleHandler</u></a> ( <a href="#"><u>RTPIdleHandler</u></a> idleHandler) Set idle handler
void	<a href="#"><u>setIOPerformanceCounter</u></a> ( <a href="#"><u>IOPerformanceCounter</u></a> ioPerformanceCounter) Set IO performance counter
void	<a href="#"><u>setIoSession</u></a> (org.apache.mina.common.support.BaseIoSession ioSession)
void	<a href="#"><u>setIp</u></a> (String ip) Set remote IP address
void	<a href="#"><u>setLastAuthenticateMethod</u></a> (int lastAuthenticateMethod) Set last method received

void	<a href="#"><u>setLoggedConnect</u></a> (boolean loggedConnect) Set connect logged
void	<a href="#"><u>setQueryStr</u></a> (String queryStr) Set query string
void	<a href="#"><u>setRedirectSession</u></a> (boolean redirectSession)
void	<a href="#"><u>setRedirectSessionCode</u></a> (int redirectSessionCode)
void	<a href="#"><u>setRedirectSessionMessage</u></a> (String redirectSessionMessage)
void	<a href="#"><u>setRedirectSessionURL</u></a> (String redirectSessionURL)
void	<a href="#"><u>setReferrer</u></a> (String referrer) Set referrer
void	<a href="#"><u>setRTSPPlayRangeStart</u></a> (double rtspPlayRangeStart) Set play start range, Internal use.
void	<a href="#"><u>setRTSPPlayRangeStop</u></a> (double rtspPlayRangeStop) Set play stop range, Internal use.
void	<a href="#"><u>setRTSPTunnelingSessionId</u></a> (String rtspTunnelingSessionId) Set the RTSP/RTP tunneling session id
void	<a href="#"><u>setServerIp</u></a> (String serverIp) Set the server IP address
void	<a href="#"><u>setServerPort</u></a> (int serverPort) Set server port
void	<a href="#"><u>setSessionId</u></a> (String sessionId) Set session id
void	<a href="#"><u>setSessionValid</u></a> (boolean isSessionValid) Set session valid
void	<a href="#"><u>setShutdownClient</u></a> (boolean shutdownClient) Set RTP session shutdown
void	<a href="#"><u>setUri</u></a> (String uri) Set URI
void	<a href="#"><u>setUserAgent</u></a> (String userAgent) Set user agent
void	<a href="#"><u>setVHost</u></a> ( <a href="#"><u>IVHost</u></a> vhost) Set vhost
void	<a href="#"><u>shutdown</u></a> ( ) shutdown RTP session, Internal use.
void	<a href="#"><u>shutdown</u></a> (RTPRequestStatus status) shutdown RTP session, Internal use.
void	<a href="#"><u>touch</u></a> ( ) Touch the stream so it doesn't timeout

void	<a href="#">updateLoggingValues()</a> 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	<a href="#">RTPSessions</a> ( <a href="#">IVHost</a> vhost) Constructor
--------	--

#### Method Summary

<a href="#">RTPSession</a>	<a href="#">addSession</a> ( <a href="#">RTPSession</a> session) Add RTP session
void	<a href="#">addSessionListener</a> ( <a href="#">IRTPSessionNotify</a> listener) Add a RTP session listener
String	<a href="#">getNextSessionId</a> () Get next RTP session id for new session
<a href="#">RTPSession</a>	<a href="#">getSession</a> (String sessionId) Get RTP session by session id
java.util.List	<a href="#">getSessionIds</a> () Get list of current RTP session ids
<a href="#">IVHost</a>	<a href="#">getVHost</a> () Get vhost
void	<a href="#">notifySessionCreate</a> ( <a href="#">IApplicationInstance</a> appInstance, <a href="#">RTPSession</a> rtpSession) Notify session create
void	<a href="#">notifySessionCreate</a> ( <a href="#">RTPSession</a> rtpSession) Notify session create
void	<a href="#">notifySessionDestroy</a> ( <a href="#">IApplicationInstance</a> appInstance, <a href="#">RTPSession</a> rtpSession) Notify session destroy
void	<a href="#">notifySessionDestroy</a> ( <a href="#">RTPSession</a> rtpSession) Notify session destroy
void	<a href="#">releaseSessionId</a> (String sessionIdStr) Release an RTP session id

<a href="#">RTPSession</a>	<a href="#">removeSession</a> ( <a href="#">RTPSession</a> session) Remove RTP session by object
<a href="#">RTPSession</a>	<a href="#">removeSession</a> (String sessionId) Remove RTP session by session id
void	<a href="#">removeSessionListener</a> ( <a href="#">IRTPSessionNotify</a> 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	<a href="#">AVSYNCMETHODS_RTPTIMECODE</a> Value: <b>3</b>
public static final	<a href="#">AVSYNCMETHODS_SENDERREPORT</a> Value: <b>1</b>
public static final	<a href="#">AVSYNCMETHODS_SYSTEMCLOCK</a> Value: <b>2</b>
public static final	<a href="#">AVSYNCMETHODS_UNKNOWN</a> Value: <b>0</b>
public static final	<a href="#">MODE_PLAY</a> Value: <b>1</b>
public static final	<a href="#">MODE_PUBLISH</a> Value: <b>2</b>
public static final	<a href="#">MODE_UNKNOWN</a> Value: <b>0</b>
public static final	<a href="#">SDPLOCATION_AUDIO</a> Value: <b>1</b>
public static final	<a href="#">SDPLOCATION_STREAM</a> Value: <b>0</b>
public static final	<a href="#">SDPLOCATION_VIDEO</a> Value: <b>2</b>
public static final	<a href="#">STREAMINFO_SESSIONATTRIBUTES</a> Value: <b>attributes</b>

public static final	<a href="#">STREAMINFO_SESSIONBANDWIDTH</a> Value: <b>bandwidth</b>
public static final	<a href="#">STREAMINFO_SESSIONCONNECTIONDATA</a> Value: <b>connectiondata</b>
public static final	<a href="#">STREAMINFO_SESSIONEMAILADDRESS</a> Value: <b>emailaddress</b>
public static final	<a href="#">STREAMINFO_SESSIONINFORMATION</a> Value: <b>information</b>
public static final	<a href="#">STREAMINFO_SESSIONNAME</a> Value: <b>name</b>
public static final	<a href="#">STREAMINFO_SESSIONPHONENUMBER</a> Value: <b>phonenummer</b>
public static final	<a href="#">STREAMINFO_SESSIONPORIGIN</a> Value: <b>origin</b>
public static final	<a href="#">STREAMINFO_SESSIONPROTOCOLVERSION</a> Value: <b>protocolversion</b>
public static final	<a href="#">STREAMINFO_SESSIONREPEATTIMES</a> Value: <b>repeattimes</b>
public static final	<a href="#">STREAMINFO_SESSIONTIMEZONES</a> Value: <b>timezones</b>
public static final	<a href="#">STREAMINFO_SESSIONTIMING</a> Value: <b>timing</b>
public static final	<a href="#">STREAMINFO_SESSIONURI</a> Value: <b>uri</b>

## Constructor Summary

public	<a href="#">RTPStream</a> ( <a href="#">RTPContext</a> context, String streamId) Constructor
--------	---

## Method Summary

void	<a href="#">addStreamAttribute</a> (String key, String value) Add a name value pair to the stream attribute collection
void	<a href="#">addStreamInfo</a> (String key, String value) Add a name value pair to the stream info collection



void	<a href="#"><u>addTrack</u></a> (RTPTrack track) Add a track
void	<a href="#"><u>addTrackId</u></a> (String seq, String trackId) Add track
void	<a href="#"><u>addTrackInternal</u></a> (RTPTrack track) Add RTP track
void	<a href="#"><u>announce</u></a> (RTPRequestStatus status) Execute announce command
void	<a href="#"><u>attachToWMSSession</u></a> (RtmpSessionInfo wmsSessionInfo) Attach to WMS session, Internal use
static int	<a href="#"><u>avSyncNameToId</u></a> (String avSyncName) Get audio/video sync id from name
void	<a href="#"><u>checkSendMetadata</u></a> (long adjTimecode, RTPTrack rtpTrack) Check to see if we have sent onMetadata event, if not send
void	<a href="#"><u>clearRTSPSessionExtraLines</u></a> () Clear SDP extra lines
void	<a href="#"><u>clearTracks</u></a> () Clear all tracks
boolean	<a href="#"><u>createStream</u></a> (RTPRequestStatus status) Create IMediaStream, Internal use.
String	<a href="#"><u>describe</u></a> (RTPSession rtspSession, int isStreamPacketizer, RTPRequestStatus status) Execute describe command
String	<a href="#"><u>describe</u></a> (RTPSession rtspSession, RTPRequestStatus status) Execute describe command
void	<a href="#"><u>detachFromWMSSession</u></a> (RtmpSessionInfo wmsSessionInfo) Detach from WMS session, Internal use.
void	<a href="#"><u>extractCodecConfigFromTrackInfo</u></a> () Extract codec config information from SDP data
String	<a href="#"><u>formatRTPInfo</u></a> (long timecode, int videoSeq, int audioSeq) Format RTP info
<a href="#"><u>IApplicationInstance</u></a>	<a href="#"><u>getAppInstance</u></a> () Get application instance
String	<a href="#"><u>getAppInstanceName</u></a> () Get the application instance name
String	<a href="#"><u>getAppName</u></a> () Get the application name
RTPTrack	<a href="#"><u>getAudioTrack</u></a> () Get the most likely audio track
int	<a href="#"><u>getAutoAllocateInterleavePorts</u></a> ()

int	<a href="#"><u>getAVSyncMethod()</u></a> Get the audio/video sync method.
double	<a href="#"><u>getDuration()</u></a> Get the duration of the stream if video on demand
String	<a href="#"><u>getHost()</u></a> Get host
<a href="#"><u>AMFPacket[]</u></a>	<a href="#"><u>getLastPacketsByType()</u></a> ( <a href="#"><u>IMediaReader</u></a> localReader, double startTime) Analyzes stream to get information, Internal use.
<a href="#"><u>AMFPacket[]</u></a>	<a href="#"><u>getLastPacketsByType()</u></a> ( <a href="#"><u>IMediaStream</u></a> localStream) Analyzes stream to get information, Internal use.
int	<a href="#"><u>getMaxRTCPWaitTime()</u></a> Get max time to wait for RTCP sender reports (milliseconds)
String	<a href="#"><u>getMediaCasterType()</u></a> Get the media caster stream type for this stream
<a href="#"><u>IMediaReader</u></a>	<a href="#"><u>getMediaReader()</u></a> Get media reader if video on demand stream
byte[]	<a href="#"><u>getMetadataPacket()</u></a> Get the onMetadata packet for this RTP stream
int	<a href="#"><u>getMode()</u></a> Get the current play/publish mode
int	<a href="#"><u>getMPEGTSAudioBitrate()</u></a>
String	<a href="#"><u>getMPEGTSAudioLanguage()</u></a>
int	<a href="#"><u>getMPEGTSAudioPID()</u></a> Get the audio PID id if MPEG-TS stream
int	<a href="#"><u>getMPEGTSProgramID()</u></a>
int	<a href="#"><u>getMPEGTSVideoBitrate()</u></a>
int	<a href="#"><u>getMPEGTSVideoPID()</u></a> Get the video PID id if MPEG-TS stream
long	<a href="#"><u>getNormalizedNTPTimecode()</u></a> (long timecode) Turn a millisecond timcode into an NTP timecode
String	<a href="#"><u>getOutHost()</u></a> Get the out host
<a href="#"><u>RTPContext</u></a>	<a href="#"><u>getRTPContext()</u></a> Get the RTP context
<a href="#"><u>RTPDestination</u></a>	<a href="#"><u>getRTPDestination()</u></a> Get RTP destination
RTPStream.RTPInfo	<a href="#"><u>getRTPInfo()</u></a> (double startTime, int videoSeq, int audioSeq) Get the RTP info

String	<a href="#"><u>getRTSPBindIpAddress()</u></a> Get the bind RTSP bind IP address
String	<a href="#"><u>getRTSPConnectionAddressType()</u></a> Get the connection address type
String	<a href="#"><u>getRTSPConnectionIpAddress()</u></a> Get the connection IP address
int	<a href="#"><u>getRTSPMaximumPendingWriteBytes()</u></a> Get the maximum number of waiting bytes allow for this RTSP session
String	<a href="#"><u>getRTSPOriginAddressType()</u></a> Get the origin address type
String	<a href="#"><u>getRTSPOriginIpAddress()</u></a> Get the origin IP address
String	<a href="#"><u>getRTSPSessionDescription()</u></a> Get session description
java.util.List	<a href="#"><u>getRTSPSessionExtraLines()</u></a> Get extra SDP lines
String	<a href="#"><u>getRTSPSessionName()</u></a> Get session name
int	<a href="#"><u>getRTSPSessionTimeout()</u></a> Get RTP session timeout (milliseconds)
String	<a href="#"><u>getSDPLang()</u></a> Get the SDP language
<a href="#"><u>RTPSession</u></a>	<a href="#"><u>getSession()</u></a> Get the RTP session
<a href="#"><u>IMediaStream</u></a>	<a href="#"><u>getStream()</u></a> Get the IMediaStream
java.util.Map	<a href="#"><u>getStreamAttributes()</u></a> Get all name/value pairs in the stream attributes collection
String	<a href="#"><u>getStreamExt()</u></a> Get the stream extension
String	<a href="#"><u>getStreamId()</u></a> Get the stream id
java.util.Map	<a href="#"><u>getStreamInfo()</u></a> Get all name/value pairs in the stream info collection
String	<a href="#"><u>getStreamInfo(String key)</u></a> Get stream info by name
Object	<a href="#"><u>getStreamLock()</u></a> Get the synchronization lock for this stream
String	<a href="#"><u>getStreamName()</u></a> Get stream name

String	<a href="#"><u>getStreamNameLogging()</u></a> Get the stream name used for logging
String	<a href="#"><u>getStreamQueryStr()</u></a> Get the stream query string
long	<a href="#"><u>getStreamSessionId()</u></a> Get the stream session id
String	<a href="#"><u>getStreamSessionIp()</u></a> Get the stream session ip
long	<a href="#"><u>getStreamSessionVersion()</u></a> Get the stream session version
RTPTrack	<a href="#"><u>getStreamTrack()</u></a> Get the most likely stream track
String	<a href="#"><u>getStreamType()</u></a> Get the stream type
RTPTrack	<a href="#"><u>getTrack()</u></a> (String trackId) Get track by id
String	<a href="#"><u>getTrackId()</u></a> (String seq) Get track by sequence number
java.util.List	<a href="#"><u>getTrackNames()</u></a> Get a list of track ids
String	<a href="#"><u>getTransportMode()</u></a> Get the transport mode
int	<a href="#"><u>getUDPManagedDeliveryCount()</u></a>
int	<a href="#"><u>getUDPManagedDeliveryDelay()</u></a>
com.wowza.wms.rtp.transport.IUDPTransport	<a href="#"><u>getUDPTransport()</u></a> (boolean isMulticast) Get the UDP transport for this stream
<a href="#"><u>IVHost</u></a>	<a href="#"><u>getVHost()</u></a> Get vhost
RTPTrack	<a href="#"><u>getVideoTrack()</u></a> Get the most likely video track
long	<a href="#"><u>getVODLastTimeTC()</u></a> Get the last timecode (milliseconds) sent for video on demand
long	<a href="#"><u>getVODPlayLen()</u></a> Get the video on demand play duration (milliseconds)
long	<a href="#"><u>getVODStartTimeTC()</u></a> Get the video on demand start time (milliseconds)
void	<a href="#"><u>idle()</u></a> (org.apache.mina.common.IoSession session, RtmpResponseMessage resp) Process idle event

void	<a href="#"><u>incrementMediaInBytes</u></a> (long bytes) Increment the media bytes in, Internal use.
boolean	<a href="#"><u>isAVSyncNonSR</u></a> ( ) Is sync method based on RTCP packets (sender report)
boolean	<a href="#"><u>isBlockUDPOut</u></a> ( )
boolean	<a href="#"><u>isCheckIpAddr</u></a> ( ) Are we checking the ip address of each incoming RTP packet
boolean	<a href="#"><u>isCheckSSRC</u></a> ( ) Are we checking the ssrc values of each incoming RTP packet
boolean	<a href="#"><u>isForceMPEGTSOut</u></a> ( )
boolean	<a href="#"><u>isForceRTSPInterleaved</u></a> ( ) True if forcing RTSP interleaved
boolean	<a href="#"><u>isLive</u></a> ( ) Is live stream
boolean	<a href="#"><u>isModePlay</u></a> ( ) Is this a play stream
boolean	<a href="#"><u>isModePublish</u></a> ( ) Is this a publish stream
boolean	<a href="#"><u>isModeUnknown</u></a> ( ) Is the stream mode unknown (publish vs play)
boolean	<a href="#"><u>isMPEGTSOut</u></a> ( ) Is MPEG-TS out
boolean	<a href="#"><u>isPaused</u></a> ( ) Is stream paused
boolean	<a href="#"><u>isPublishStreamReady</u></a> ( ) See if a publishing stream has enough data to start playback
boolean	<a href="#"><u>isResetPlayStream</u></a> ( ) Is reset stream trigger, Internal use.
boolean	<a href="#"><u>isResyncAudioVideoOnSR</u></a> ( ) Reset audio/video sync on new RTCP packets (not just first packet)
boolean	<a href="#"><u>isRTPIgnoreProfileLevelId</u></a> ( )
boolean	<a href="#"><u>isRTSP</u></a> ( ) Is this RTP Stream managed by RTSP session
boolean	<a href="#"><u>isRTSPAlwaysUseSDPPorts</u></a> ( ) Force RTSP to use ports in SDP data
boolean	<a href="#"><u>isRTSPPull</u></a> ( ) Is this RTP Stream managed by RTSP session

boolean	<a href="#"><u>isSendSDESEvents</u></a> ( ) Send RTCP SDES events
boolean	<a href="#"><u>isStreamStarted</u></a> ( ) Is stream started
boolean	<a href="#"><u>isTimeout</u></a> (long currTime, int timeout) Is the stream timeout out
void	<a href="#"><u>lockRepeaterStreams</u></a> ( java.util.List streamNames) Lock a list of live repeater stream names, Internal use.
void	<a href="#"><u>pause</u></a> (RTPRequestStatus status) Execute pause
RTPStreamPlayResult	<a href="#"><u>play</u></a> (RTPRequestStatus status) Execute play
RTPStreamPlayResult	<a href="#"><u>play</u></a> (RTPRequestStatus status, double startTime, double stopTime) Execute play
void	<a href="#"><u>putRTSPSessionExtraLine</u></a> (int location, String line) Add an extra line to the SDP data
RTPStreamPlayResult	<a href="#"><u>record</u></a> (RTPRequestStatus status) Execute record
RTPStreamPlayResult	<a href="#"><u>record</u></a> (RTPRequestStatus status, double startTime, double stopTime) Execute record
RTPTrack	<a href="#"><u>removeTrack</u></a> (String trackId) Remove a track by id
RTPTrack	<a href="#"><u>removeTrackInternal</u></a> (String trackId) Remove track by track id
void	<a href="#"><u>resetSentMetadataFlag</u></a> ( ) Reset sendMetadata flag
void	<a href="#"><u>setAppInstanceName</u></a> (String appInstanceName) Set the application instance name
void	<a href="#"><u>setAppName</u></a> (String appName) Get the application name
void	<a href="#"><u>setAVSyncMethod</u></a> (int avSyncMethod) Set the audio/video sync method.
void	<a href="#"><u>setBlockUDPOut</u></a> (boolean blockUDPOut)
void	<a href="#"><u>setCheckIpAddr</u></a> (boolean checkIpAddr) Are we checking the ip address of each incoming RTP packet
void	<a href="#"><u>setCheckSSRC</u></a> (boolean checkSSRC) Are we checking the ssrc values of each incoming RTP packet
void	<a href="#"><u>setForceMPEGTSOut</u></a> (boolean isForceMPEGTSOut)

void	<a href="#"><u>setForceRTSPInterleaved</u></a> (boolean isForceRTSPInterleaved) True if forcing RTSP interleaved
void	<a href="#"><u>setHost</u></a> (String host) Set host
void	<a href="#"><u>setLive</u></a> (boolean isLive) Is live stream
void	<a href="#"><u>setMaxRTCPWaitTime</u></a> (int maxRTCPWaitTime) Set max time to wait for RTCP sender reports (milliseconds)
void	<a href="#"><u>setMode</u></a> (int mode) Set the play/publish mode
void	<a href="#"><u>setMPEGTSAudioBitrate</u></a> (int mpegtsAudioBitrate)
void	<a href="#"><u>setMPEGTSAudioLanguage</u></a> (String mpegtsAudioLanguage)
void	<a href="#"><u>setMPEGTSAudioPID</u></a> (int mpegtsAudioPID) Set the audio PID id if MPEG-TS stream
void	<a href="#"><u>setMPEGTSOut</u></a> (boolean isMPEGTSOut) Is MPEG-TS out
void	<a href="#"><u>setMPEGTSProgramID</u></a> (int mpegtsProgramId)
void	<a href="#"><u>setMPEGTSVideoBitrate</u></a> (int mpegtsVideoBitrate)
void	<a href="#"><u>setMPEGTSVideoPID</u></a> (int mpegtsVideoPID) Set the video PID id if MPEG-TS stream
void	<a href="#"><u>setOutHost</u></a> (String outHost) Set out host
void	<a href="#"><u>setResetPlayStream</u></a> (boolean doResetPlayStream) Set reset stream trigger
void	<a href="#"><u>setResyncAudioVideoOnSR</u></a> (boolean resyncAudioVideoOnSR) Reset audio/video sync on new RTCP packets (not just first packet)
void	<a href="#"><u>setRTPDestination</u></a> ( <a href="#"><u>RTPDestination</u></a> rtpDestination) Set RTP destination
void	<a href="#"><u>setRTPIgnoreProfileLevelId</u></a> (boolean rtpIgnoreProfileLevelId)
void	<a href="#"><u>setRTSP</u></a> (boolean isRTSP) Is this RTP Stream managed by RTSP session
void	<a href="#"><u>setRTSPAlwaysUseSDPPorts</u></a> (boolean rtspAlwaysUseSDPPorts) Force RTSP to use ports in SDP data
void	<a href="#"><u>setRTSPBindIpAddress</u></a> (String rtspBindIpAddress) Set the bind RTSP bind IP address
void	<a href="#"><u>setRTSPConnectionAddressType</u></a> (String rtspConnectionAddressType) Set the connection address type

void	<a href="#"><u>setRTSPConnectionIpAddress</u></a> (String rtspConnectionIpAddress) Set the connection IP address
void	<a href="#"><u>setRTSPMaximumPendingWriteBytes</u></a> (int rtspMaximumPendingWriteBytes) Set the maximum number of waiting bytes allow for this RTSP session
void	<a href="#"><u>setRTSPOriginAddressType</u></a> (String rtspOriginAddressType) Set the origin address type
void	<a href="#"><u>setRTSPOriginIpAddress</u></a> (String rtspOriginIpAddress) Set the origin IP address
void	<a href="#"><u>setRTSPPull</u></a> (boolean isRTSPPull) Is this RTP Stream managed by RTSP session
void	<a href="#"><u>setRTSPSessionDescription</u></a> (String rtspSessionDescription) Get session description
void	<a href="#"><u>setRTSPSessionName</u></a> (String rtspSessionName) Set session name
void	<a href="#"><u>setRTSPSessionTimeout</u></a> (int rtspSessionTimeout) Set RTP session timeout (milliseconds)
void	<a href="#"><u>setSDPLang</u></a> (String sdpLang) Set the SDP language
void	<a href="#"><u>setSendSDESEvents</u></a> (boolean sendSDESEvents) Send RTCP SDES events
void	<a href="#"><u>setSession</u></a> ( <a href="#"><u>RTPSession</u></a> session) Set the RTP session
void	<a href="#"><u>setStreamExt</u></a> (String streamExt) Set the stream extension
void	<a href="#"><u>setStreamName</u></a> (String streamName) Set stream name
void	<a href="#"><u>setStreamNameLogging</u></a> (String streamNameLogging) Set the stream name used for logging
void	<a href="#"><u>setStreamQueryStr</u></a> (String streamQueryStr) Set the stream query string
void	<a href="#"><u>setStreamSessionId</u></a> (long streamSessionId) Set the stream session id
void	<a href="#"><u>setStreamSessionIp</u></a> (String streamSessionIp) Set the stream session ip
void	<a href="#"><u>setStreamSessionVersion</u></a> (long streamSessionVersion) Set the stream session version
void	<a href="#"><u>setStreamType</u></a> (String streamType) Set the stream type
void	<a href="#"><u>setTransportMode</u></a> (String transportMode) Set the transport mode



void	<a href="#"><u>setUDPManagedDeliveryCount</u></a> (int udpManagedDeliveryCount)
void	<a href="#"><u>setUDPManagedDeliveryDelay</u></a> (int udpManagedDeliveryDelay)
void	<a href="#"><u>setVODLastTimeTC</u></a> (long vodLastTimeTC) Set the last timecode (milliseconds) sent for video on demand
void	<a href="#"><u>setVODPlayLen</u></a> (long vodPlayLen) Set the video on demand play duration (milliseconds)
void	<a href="#"><u>setVODStartTimeTC</u></a> (long vodStartTimeTC) Set the video on demand start time (milliseconds)
void	<a href="#"><u>shutdown</u></a> (RTPRequestStatus status) shutdown RTP stream, Internal use.
RTPTrack	<a href="#"><u>sloppyGetTrack</u></a> (String trackId) Sloppy method for finding track by name, Internal use.
boolean	<a href="#"><u>streamExists</u></a> () Return true if RTP stream contains a IMediaStream
void	<a href="#"><u>switchSetupToMPEGTS</u></a> () Switch a stream to MPEG-TS, Internal use.
void	<a href="#"><u>touch</u></a> () Touch the stream so that it does not timeout
String	<a href="#"><u>transportFindBestMatch</u></a> (String transport) Based on a trasport string from SETUP command find best match
void	<a href="#"><u>unlockRepeaterStreams</u></a> () 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	<a href="#"><u>canHandle</u></a> (CommandInterfaceRequestMessage req)
void	<a href="#"><u>invoke</u></a> (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	<a href="#"><u>onResponseWriteStart</u></a> (RtmpResponseMessage response)
void	<a href="#"><u>onResponseWriteStop</u></a> (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	<a href="#">addServerListener</a> ( <a href="#">IServerNotify</a> serverListener) Add server listener
java.util.List	<a href="#">getAdminInterfaceObjectList</a> () Get the list of objects exposed through JMX interface
RandomIdGenerator	<a href="#">getClientIdGenerator</a> () Get the client id generator for the server
CommandInterfaceCommandHandler	<a href="#">getCommandInterfaceCommandHandler</a> () Get the command interface command handler
<a href="#">HostPort</a>	<a href="#">getCommandInterfaceHostPort</a> () Get the definition of the command interface
<a href="#">ConnectionCounter</a>	<a href="#">getConnectionCounter</a> () Get the server connection counter.
ConnectionCounterSimple	<a href="#">getConnectionCounter</a> (int counterIndex) Get the server connection counter for a specific technology (see IVHost.COUNTER_*)
int	<a href="#">getCoreHandlerPoolSize</a> () Get the handler core thread pool size.
int	<a href="#">getCoreTransportPoolSize</a> () Get the transport core thread pool size.
String	<a href="#">getDateStarted</a> () Get the date and time the server was started.
java.util.Properties	<a href="#">getDynamicLogProperties</a> () Get the dynamic log properties defined at the server level in conf/log4j.properties
<a href="#">ThreadPool</a>	<a href="#">getHandlerThreadPool</a> () Get the server handler thread pool.
<a href="#">IOPerformanceCounter</a>	<a href="#">getIoPerformanceCounter</a> () Get the server performance counter.
<a href="#">IOPerformanceCounter</a>	<a href="#">getIoPerformanceCounter</a> (int counterIndex) Get the server performance counter for a specific technology (see IVHost.COUNTER_*)

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

void	<a href="#"><code>setUserAgents</code></a> (String[] userAgents) Set a pipe " " delimited list of user agents that the server recognizes as RTMPT client.
void	<a href="#"><code>startCommandInterface</code></a> () Start the command interface as defined in Server.xml.
void	<a href="#"><code>startVHost</code></a> (String vhostName) Start a vHost by name.
void	<a href="#"><code>startVHosts</code></a> () Start all vHosts
void	<a href="#"><code>stopAdminAgent</code></a> () Stop the JMX interface
void	<a href="#"><code>stopCommandInterface</code></a> () Stop the command interface as defined in Server.xml.
void	<a href="#"><code>stopVHost</code></a> (String vhostName) Stop a vHost by name.
void	<a href="#"><code>stopVHosts</code></a> () Stop all vHosts
void	<a href="#"><code>suspendAllVHosts</code></a> () Suspend all virtual hosts (Calls IVHost.suspendAllHostPorts for each vhost)
void	<a href="#"><code>suspendServer</code></a> () Suspend all virtual hosts and the command interface
void	<a href="#"><code>unbindAllVHosts</code></a> () Unbind all virtual hosts (Calls IVHost.unbindAllHostPorts for each vhost)
boolean	<a href="#"><code>writeConfig</code></a> (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	<a href="#">onServerCreate</a> ( <a href="#">IServer</a> server) Triggered when server object is first created.
void	<a href="#">onServerInit</a> ( <a href="#">IServer</a> server) Triggered when server initialization is complete and all VHosts have been started
void	<a href="#">onServerShutdownComplete</a> ( <a href="#">IServer</a> server) Triggered at the end of server shutdown
void	<a href="#">onServerShutdownStart</a> ( <a href="#">IServer</a> 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	<a href="#">onServerConfigLoaded</a> ( <a href="#">IServer</a> 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	<a href="#">Server.TranscoderPollingTracker</a> Server.TranscoderPollingTracker
-------	--

### Field Summary

public static	<a href="#">logNotifier</a>
---------------	-----------------------------

### Constructor Summary

public	<a href="#">Server()</a>
--------	--------------------------

### Method Summary

void	<a href="#">addServerListener</a> ( <a href="#">IServerNotify</a> serverListener)
static void	<a href="#">decodeS</a> (String[] ins) <b>Deprecated.</b>
static boolean	<a href="#">decodeSS</a> (String in) <b>Deprecated.</b>
String	<a href="#">decodeStorageDir</a> ( <a href="#">IVHost</a> vhost, String storageDir)
void	<a href="#">doWatchdog</a> ()
static String	<a href="#">fs</a> (byte[] kiIn, String p) <b>Deprecated.</b>
com.wowza.wms.admin.AdminAgent	<a href="#">getAdminAgent</a> ()
String	<a href="#">getAdminGUID</a> ()
java.util.List	<a href="#">getAdminInterfaceObjectList</a> ()

RandomIdGenerator	<a href="#">getClientIdGenerator()</a>
int	<a href="#">getClientIdGeneratorRecycleDelaySize()</a>
int	<a href="#">getClientIdGeneratorRecycleSize()</a>
long	<a href="#">getClientIdGeneratorTimeout()</a>
Object	<a href="#">getCommandInterface()</a>
CommandInterfaceCommandHandler	<a href="#">getCommandInterfaceCommandHandler()</a>
<a href="#">HostPort</a>	<a href="#">getCommandInterfaceHostPort()</a>
long	<a href="#">getCommittedVirtualMemory()</a>
<a href="#">ConnectionCounter</a>	<a href="#">getConnectionCounter()</a>
ConnectionCounterSimple	<a href="#">getConnectionCounter(int counterIndex)</a>
<a href="#">IConnectionValidator</a>	<a href="#">getConnectionValidator()</a>
int	<a href="#">getCoreHandlerPoolSize()</a>
int	<a href="#">getCoreTransportPoolSize()</a>
int	<a href="#">getCryptoPoolActiveCount()</a>
int	<a href="#">getCryptoPoolMaxSize()</a>
long	<a href="#">getCurrentHeapSize()</a>
String	<a href="#">getDateStarted()</a>
java.util.Properties	<a href="#">getDynamicLogProperties()</a>
String	<a href="#">getGUID()</a>
<a href="#">ThreadPool</a>	<a href="#">getHandlerThreadPool()</a>
static <a href="#">Server</a>	<a href="#">getInstance()</a>
<a href="#">IOPerformanceCounter</a>	<a href="#">getIoPerformanceCounter()</a>
<a href="#">IOPerformanceCounter</a>	<a href="#">getIoPerformanceCounter(int counterIndex)</a>
JMXRemoteConfig	<a href="#">getJmxRemoteConfig()</a>

LicenseCounter	<a href="#">getLicenseCounter</a> (int index)
Server.LicenseSessionTracker	<a href="#">getLicenseTracker</a> (int index)
<a href="#">ILicenseValidator</a>	<a href="#">getLicenseValidator</a> ()
long	<a href="#">getLiveThreads</a> ()
long	<a href="#">getMaxHeapSize</a> ()
long	<a href="#">getPeakThreads</a> ()
<a href="#">WMSProperties</a>	<a href="#">getProperties</a> ()
boolean[]	<a href="#">getProtocolUsage</a> ()
Server.ProtocolUsageSessionTracker	<a href="#">getProtocolUsageTracker</a> ()
String	<a href="#">getServerGUID</a> ()
String	<a href="#">getSessionGUID</a> ()
<a href="#">ThreadPool</a>	<a href="#">getThreadPool</a> ()
String	<a href="#">getTimeRunning</a> ()
double	<a href="#">getTimeRunningSeconds</a> ()
<a href="#">ThreadPool</a>	<a href="#">getTransportThreadPool</a> ()
com.wowza.wms.transport.udp.UDPPortManager	<a href="#">getUDPPortManager</a> ()
com.wowza.wms.transport.udp.UDPPortSharingManager	<a href="#">getUDPPortSharingManager</a> ()
String[]	<a href="#">getUserAgents</a> ()
String	<a href="#">getVersion</a> ()
<a href="#">VHostList</a>	<a href="#">getVHostList</a> ()
boolean	<a href="#">isDynamicLogContextLoaded</a> (String logContext)
boolean	<a href="#">isSuspended</a> ()
boolean	<a href="#">isVHostRunning</a> (String name)

static void	<a href="#">main</a> (String[] args)
void	<a href="#">onNewVHost</a> ( <a href="#">IVHost</a> vhost)
String	<a href="#">readConfig</a> (String sName)
static String	<a href="#">readXMLConfig</a> (String sPath)
void	<a href="#">registerLiveStreamTranscoder</a> ( <a href="#">ILiveStreamTranscoder</a> liveStreamTranscoder, byte[] license)
void	<a href="#">reloadVHostConfig</a> ()
void	<a href="#">removeServerListener</a> ( <a href="#">IServerNotify</a> serverListener)
void	<a href="#">setCommandInterface</a> (Object commandInterface)
void	<a href="#">setCommandInterfaceHostPort</a> ( <a href="#">HostPort</a> commandInterfaceHostPort)
void	<a href="#">setCoreHandlerPoolSize</a> (int corePoolSize)
void	<a href="#">setCoreTransportPoolSize</a> (int corePoolSize)
void	<a href="#">setDynamicLogProperties</a> (java.util.Properties dynamicLogProperties)
void	<a href="#">setIoPerformanceCounter</a> ( <a href="#">IOPerformanceCounter</a> ioPerformanceCounter)
void	<a href="#">setUserAgents</a> (String[] userAgents)
static void	<a href="#">start</a> ()
void	<a href="#">startCommandInterface</a> ()
void	<a href="#">startServer</a> ()
void	<a href="#">startVHost</a> (String vhostName)
void	<a href="#">startVHosts</a> ()
void	<a href="#">stopAdminAgent</a> ()
void	<a href="#">stopCommandInterface</a> ()
void	<a href="#">stopServer</a> ()
void	<a href="#">stopVHost</a> (String vhostName)
void	<a href="#">stopVHosts</a> ()

void	<a href="#"><u>suspendAllVHosts()</u></a>
void	<a href="#"><u>suspendCommandInterface()</u></a>
void	<a href="#"><u>suspendServer()</u></a>
void	<a href="#"><u>unbindAllVHosts()</u></a>
void	<a href="#"><u>unregisterLiveStreamTranscoder(ILiveStreamTranscoder liveStreamTranscoder, byte[] license)</u></a>
<a href="#"><u>IOPerformanceCounter</u></a>	<a href="#"><u>updateIOPerformance()</u></a>
void	<a href="#"><u>updateLoggingDuration()</u></a>
boolean	<a href="#"><u>writeConfig</u></a> (String sName, String data)
static boolean	<a href="#"><u>writeXMLConfig</u></a> (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	<a href="#">FILEEXTENSION</a> Value: <b>rso</b>
public static final	<a href="#">SHARED_OBJECT_CMD_CONNECT</a> shared object command: connect Value: <b>1</b>
public static final	<a href="#">SHARED_OBJECT_CMD_CONNECTSUCCESS</a> shared object command: clear Value: <b>11</b>
public static final	<a href="#">SHARED_OBJECT_CMD_DELETE</a> shared object command: delete Value: <b>10</b>
public static final	<a href="#">SHARED_OBJECT_CMD_DISCONNECT</a> shared object command: disconnect Value: <b>2</b>
public static final	<a href="#">SHARED_OBJECT_CMD_ERROR</a> shared object command: error Value: <b>7</b>
public static final	<a href="#">SHARED_OBJECT_CMD_SEND</a> shared object command: send Value: <b>6</b>
public static final	<a href="#">SHARED_OBJECT_CMD_SETVALUE</a> shared object command: setvalue Value: <b>3</b>
public static final	<a href="#">SHARED_OBJECT_STATUS_CHANGE</a> shared object status: change Value: <b>4</b>
public static final	<a href="#">SHARED_OBJECT_STATUS_CLEAR</a> shared object status: clear Value: <b>8</b>

public static final	<a href="#"><u>SHARED_OBJECT_STATUS_DELETE</u></a> shared object status: delete Value: <b>9</b>
public static final	<a href="#"><u>SHARED_OBJECT_STATUS_SUCCESS</u></a> shared object status: success Value: <b>5</b>

## Method Summary

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

<a href="#"><u>ISharedObjectSlot</u></a>	<a href="#"><u>getSlot</u></a> (String name) Get ISharedObjectSlot interface to a slot (property) by name
java.util.List	<a href="#"><u>getSlotNames</u></a> ( ) Get a list of slot (property) names
java.util.List	<a href="#"><u>getSlots</u></a> ( ) Get a list of active slots
String	<a href="#"><u>getStorageDir</u></a> ( ) Get path used to store shared object.
int	<a href="#"><u>getVersion</u></a> ( ) Get the interval version number.
boolean	<a href="#"><u>isClient</u></a> ( <a href="#"><u>IClient</u></a> client) Is this client connected to shared object
boolean	<a href="#"><u>isPersistent</u></a> ( ) Is this shared object being persisted.
void	<a href="#"><u>lock</u></a> ( ) Lock a shared object for write access
int	<a href="#"><u>purge</u></a> (int version) Purge all deleted properties older than the version number
void	<a href="#"><u>putSlot</u></a> (String name, <a href="#"><u>ISharedObjectSlot</u></a> slot) Add a new slot (property) to a shared object.
void	<a href="#"><u>release</u></a> ( ) Decrement the reference count to this shared object.
void	<a href="#"><u>removeClient</u></a> ( <a href="#"><u>IClient</u></a> client) Remove a client from this shared object.
void	<a href="#"><u>removeSlotListener</u></a> ( <a href="#"><u>ISharedObjectSlotNotify</u></a> slotListener) Remove slot listener
void	<a href="#"><u>send</u></a> (String handlerName) Call client side handler attached to shared object (no parameters).
void	<a href="#"><u>send</u></a> (String handlerName, Object[] params) Call client side handler attached to shared object.
void	<a href="#"><u>setName</u></a> (String name) Set shared object name
void	<a href="#"><u>setPersistent</u></a> (boolean isPersistent) Set is shared object persisted.
void	<a href="#"><u>setProperty</u></a> (String slotName, <a href="#"><u>AMFData</u></a> data) Set slot (property) value as AMFData object.
void	<a href="#"><u>setProperty</u></a> (String slotName, boolean value) Set slot (property) value as a boolean value (will be wrapped in an AMFDataItem object)
void	<a href="#"><u>setProperty</u></a> (String slotName, java.util.Date value) Set slot (property) value as a date value (will be wrapped in an AMFDataItem object)

void	<a href="#"><code>setProperty</code></a> (String slotName, double value) Set slot (property) value as a double value (will be wrapped in an AMFDataItem object)
void	<a href="#"><code>setProperty</code></a> (String slotName, int value) Set slot (property) value as a int value (will be wrapped in an AMFDataItem object)
void	<a href="#"><code>setProperty</code></a> (String slotName, long value) Set slot (property) value as a long value (will be wrapped in an AMFDataItem object)
void	<a href="#"><code>setProperty</code></a> (String slotName, String value) Set slot (property) value as a string value (will be wrapped in an AMFDataItem object)
void	<a href="#"><code>setStorageDir</code></a> (String storageDir) Set path used to store shared object.
void	<a href="#"><code>setVersion</code></a> (int version) Set the internal version number.
int	<a href="#"><code>size</code></a> () Get the number of active slot (properties).
void	<a href="#"><code>unlock</code></a> () Unlock a shared object for write access
void	<a href="#"><code>writeDeleteError</code></a> (IClient client, String soName, boolean isPersistent, String slotName, String errorMsg) Write an delete error message back to the client
void	<a href="#"><code>writeSetValueError</code></a> (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	<a href="#">onSharedObjectConnect</a> ( <a href="#">ISharedObject</a> sharedObject, <a href="#">IClient</a> client) Triggered when client connects to sharedObject
void	<a href="#">onSharedObjectCreate</a> ( <a href="#">ISharedObject</a> sharedObject) Triggered when sharedObject created
void	<a href="#">onSharedObjectDestroy</a> ( <a href="#">ISharedObject</a> sharedObject) Triggered when sharedObject destroyed
void	<a href="#">onSharedObjectDisconnect</a> ( <a href="#">ISharedObject</a> sharedObject, <a href="#">IClient</a> 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	<a href="#"><u>addSharedObjectListener</u></a> ( <a href="#"><u>ISharedObjectNotify</u></a> sharedObjectListener)
	Add a shared object listener.
void	<a href="#"><u>disconnect</u></a> ( <a href="#"><u>IClient</u></a> client)
	Disconnect client from all shared objects in list.
boolean	<a href="#"><u>exists</u></a> ( <a href="#"><u>ISharedObject</u></a> sharedObject)
	Is sharedObject in this list (by shared object reference).
boolean	<a href="#"><u>exists</u></a> (String objectName)
	Is sharedObject in this list (by name).
void	<a href="#"><u>flush</u></a> ()
	Flush all persistent shared objects to disk.
<a href="#"><u>ISharedObject</u></a>	<a href="#"><u>get</u></a> (String name)
	Get shared object by name.
java.util.List	<a href="#"><u>getObjectNames</u></a> ()
	Get a list of shared object names.
<a href="#"><u>ISharedObject</u></a>	<a href="#"><u>getOrCreate</u></a> (String name)
	Get shared object by name if it does not exist create a new shared object with the given name.
String	<a href="#"><u>getStorageDir</u></a> ()
	Get the storage directory for all shared objects in list.
boolean	<a href="#"><u>isPersistent</u></a> ()
	Are shared objects in list persistent.
void	<a href="#"><u>load</u></a> ()
	Load persistent shared objects from file system.
void	<a href="#"><u>put</u></a> (String objectName, <a href="#"><u>ISharedObject</u></a> sharedObject)
	Add or replace a shared object.
void	<a href="#"><u>remove</u></a> (String objectName)
	Remove a shared object.
void	<a href="#"><u>removeClient</u></a> ( <a href="#"><u>IClient</u></a> client)
	Remove a client from any shared object that it is connected to in this list.
void	<a href="#"><u>removeSharedObjectListener</u></a> ( <a href="#"><u>ISharedObjectNotify</u></a> sharedObjectListener)
	Remove a shared object listener.

void	<a href="#"><code>setPersistent</code></a> (boolean isPersistent) Set is shared object in list persistent
void	<a href="#"><code>setStorageDir</code></a> (String storageDir) Set the storage directory for all shared objects in list.
int	<a href="#"><code>size</code></a> () 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 if shared object in list persistent

(continued from last page)

**Parameters:**

isPersistent - true is shared objects in list are persistent

---

**removeClient**

```
public void removeClient(IClient client)
```

Remove a client from any shared object that it is connected to in this list.

**Parameters:**

client - client

---

**flush**

```
public void flush()
```

Flush all persistent shared objects to disk.

---

**disconnect**

```
public void disconnect(IClient client)
```

Disconnect client from all shared objects in list.

**Parameters:**

client - client

---

**addSharedObjectListener**

```
public void addSharedObjectListener(ISharedObjectNotify sharedObjectListener)
```

Add a shared object listener. The listener will receive the following events: onSharedObjectCreate, onSharedObjectDestroy, onSharedObjectConnect, onSharedObjectDisconnect.

**Parameters:**

sharedObjectListener

---

**removeSharedObjectListener**

```
public void removeSharedObjectListener(ISharedObjectNotify sharedObjectListener)
```

Remove a shared object listener.

**Parameters:**

sharedObjectListener

---

**getStorageDir**

```
public String getStorageDir()
```

Get the storage directory for all shared objects in list.

**Returns:**

storage dir

---

**setStorageDir**

```
public void setStorageDir(String storageDir)
```

(continued from last page)

Set the storage directory for all shared objects in list.

**Parameters:**

storageDir

---

## load

```
public void load()
```

Load persistent shared objects from file system.

## com.wowza.wms.sharedobject Interface ISharedObjectSlot

public interface **ISharedObjectSlot**  
extends

ISharedObjectSlot: public interface to SharedObjectSlot class.

### Method Summary

<a href="#">AMFData</a>	<a href="#">getData()</a> Get slot data as AMFData object.
int	<a href="#">getLastClientId()</a> Get client id of client that performed last operation on slot.
int	<a href="#">getLastOperation()</a> Get last slot (property) operation.
String	<a href="#">getName()</a> Get slot (property) name.
int	<a href="#">getSlotVersion()</a> Get slot version
int	<a href="#">getSoVersion()</a> Get parent shared object version
void	<a href="#">incSlotVersion()</a> Increment slot version by 1.
void	<a href="#">init</a> (String name, <a href="#">AMFData</a> data, int slotVersion) Initialize shared object slot
void	<a href="#">setData</a> ( <a href="#">AMFData</a> data) Set slot data as AMFData object.
void	<a href="#">setData</a> (byte[] data) Set slot data as byte[].
void	<a href="#">setData</a> (byte[] data, <a href="#">AMFDataContextDeserialize</a> context) Set slot data as byte[].
void	<a href="#">setLastClientId</a> (int lastClientId) Set client id of client that performed last operation on slot.
void	<a href="#">setLastOperation</a> (int lastOperation) Set last slot (property) operation.
void	<a href="#">setName</a> (String name) Set slot (property) name
void	<a href="#">setSlotVersion</a> (int slotVersion) Set slot version

void	<a href="#">setSoVersion</a> (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	<a href="#">onSlotDelete</a> ( <a href="#">ISharedObject</a> sharedObject, <a href="#">ISharedObjectSlot</a> slot) Triggered when sharedObject slot value deleted
void	<a href="#">onSlotSetValue</a> ( <a href="#">ISharedObject</a> sharedObject, <a href="#">ISharedObjectSlot</a> 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	<a href="#">READACCESS</a> Value: <b>0</b>
public static final	<a href="#">WRITEACCESS</a> Value: <b>1</b>

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	<a href="#">SharedObject</a> (String name) Create new shared object
public	<a href="#">SharedObject</a> (String name, boolean isPersistent, String storageDir) Create new shared object

### Method Summary

void	<a href="#">acquire</a> ()
void	<a href="#">addClient</a> ( <a href="#">IClient</a> client)
void	<a href="#">addSlotListener</a> ( <a href="#">ISharedObjectSlotNotify</a> slotListener)
void	<a href="#">clear</a> ()
void	<a href="#">close</a> ()

boolean	<a href="#"><u>containsProperty</u></a> (String slotName)
boolean	<a href="#"><u>containsSlot</u></a> (String slotName)
void	<a href="#"><u>deleteSlot</u></a> ( <a href="#"><u>IClient</u></a> client, String slotName)
void	<a href="#"><u>deleteSlot</u></a> (String slotName)
void	<a href="#"><u>disconnect</u></a> ( <a href="#"><u>IClient</u></a> client)
void	<a href="#"><u>flush</u></a> ()
static boolean[]	<a href="#"><u>getAccess</u></a> ( <a href="#"><u>IClient</u></a> client, String soName)
java.util.List	<a href="#"><u>getClients</u></a> ()
void	<a href="#"><u>getClientUpdates</u></a> ( <a href="#"><u>IClient</u></a> client)
String	<a href="#"><u>getName</u></a> ()
<a href="#"><u>ISharedObjects</u></a>	<a href="#"><u>getParent</u></a> ()
<a href="#"><u>AMFData</u></a>	<a href="#"><u>getProperty</u></a> (String slotName)
int	<a href="#"><u>getRefCount</u></a> ()
<a href="#"><u>ISharedObjectSlot</u></a>	<a href="#"><u>getSlot</u></a> (String name)
java.util.List	<a href="#"><u>getSlotNames</u></a> ()
java.util.List	<a href="#"><u>getSlots</u></a> ()
String	<a href="#"><u>getStorageDir</u></a> ()
int	<a href="#"><u>getVersion</u></a> ()
boolean	<a href="#"><u>isClient</u></a> ( <a href="#"><u>IClient</u></a> client)
boolean	<a href="#"><u>isPersistent</u></a> ()
void	<a href="#"><u>load</u></a> ()
void	<a href="#"><u>lock</u></a> ()
void	<a href="#"><u>notifySlotDelete</u></a> ( <a href="#"><u>ISharedObjectSlot</u></a> slot)
void	<a href="#"><u>notifySlotSetValue</u></a> ( <a href="#"><u>ISharedObjectSlot</u></a> slot)

int	<a href="#">purge</a> (int version)
void	<a href="#">putSlot</a> (String name, <a href="#">ISharedObjectSlot</a> slot)
void	<a href="#">release</a> ()
void	<a href="#">removeClient</a> ( <a href="#">IClient</a> client)
void	<a href="#">removeSlotListener</a> ( <a href="#">ISharedObjectSlotNotify</a> slotListener)
void	<a href="#">send</a> (String handlerName)
void	<a href="#">send</a> (String handlerName, Object[] params)
void	<a href="#">sendInternal</a> ( <a href="#">IClient</a> client, String handlerName, byte[] msgBytes)
void	<a href="#">sendInternal</a> ( <a href="#">IClient</a> client, String handlerName, byte[] msgBytes, <a href="#">AMFDataContextDeserialize</a> context)
void	<a href="#">sendInternal</a> (String handlerName, <a href="#">AMFData[]</a> params)
void	<a href="#">setName</a> (String name)
void	<a href="#">setParent</a> ( <a href="#">ISharedObjects</a> parent)
void	<a href="#">setPersistent</a> (boolean isPersistent)
void	<a href="#">setProperty</a> (String slotName, <a href="#">AMFData</a> data)
void	<a href="#">setProperty</a> (String slotName, boolean value)
void	<a href="#">setProperty</a> (String slotName, java.util.Date value)
void	<a href="#">setProperty</a> (String slotName, double value)
void	<a href="#">setProperty</a> (String slotName, int value)
void	<a href="#">setProperty</a> (String slotName, long value)
void	<a href="#">setProperty</a> (String slotName, String value)
void	<a href="#">setSlotValue</a> ( <a href="#">IClient</a> client, String slotName, <a href="#">AMFData</a> amfData)
void	<a href="#">setSlotValue</a> ( <a href="#">IClient</a> client, String slotName, byte[] byteData)
void	<a href="#">setSlotValue</a> ( <a href="#">IClient</a> client, String slotName, byte[] byteData, <a href="#">AMFData</a> amfData)

void	<a href="#">setSlotValue(IClient client, String slotName, byte[] byteData, <a href="#">AMFData</a> amfData, <a href="#">AMFDataContextDeserialize</a> context)</a>
void	<a href="#">setSlotValue(IClient client, String slotName, byte[] byteData, <a href="#">AMFDataContextDeserialize</a> context)</a>
void	<a href="#">setStorageDir</a> (String storageDir)
void	<a href="#">setVersion</a> (int version)
int	<a href="#">size</a> ()
void	<a href="#">unlock</a> ()
void	<a href="#">writeDeleteError(IClient client, String soName, boolean isPersistent, String slotName, String errorMsg)</a>
static void	<a href="#">writeError(IClient client, String soName, boolean isPersistent, String errorMsg, boolean isConnect)</a>
void	<a href="#">writeSetValueError(IClient client, String soName, boolean isPersistent, String slotName, String errorMsg)</a>

#### 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	<a href="#"><u>FastPlaySettings()</u></a> Create empty object
public	<a href="#"><u>FastPlaySettings(double multiplier, int fps, int direction)</u></a> Create object

#### Method Summary

int	<a href="#"><u>getDirection()</u></a> Get direction (1 forward, -1 reverse)
int	<a href="#"><u>getFps()</u></a> Get frames per second
double	<a href="#"><u>getMultiplier()</u></a> Get speed of fast play
long	<a href="#"><u>getStartTC()</u></a> Get the timecode (milliseconds) where this fast play started
long	<a href="#"><u>getStartTCOffset()</u></a> Get the timecode (milliseconds) where this fast play started (not sure why we have both values)
void	<a href="#"><u>setDirection(int direction)</u></a> Set direction (1 forward, -1 reverse)
void	<a href="#"><u>setFps(int fps)</u></a> Set frames per second
void	<a href="#"><u>setMultiplier(double multiplier)</u></a> Set speed of fast play
void	<a href="#"><u>setStartTC(long startTC)</u></a> Set the timecode (milliseconds) where this fast play started
void	<a href="#"><u>setStartTCOffset(long startTCOffset)</u></a> 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	<a href="#">MISSING</a> media file is missing Value: <b>**missing**</b>
---------------------	---

### Method Summary

<a href="#">IMediaIndexItem</a>	<a href="#">clone()</a> Make a clone copy of media index
String	<a href="#">getChecksum()</a> Get checksum for media index
int	<a href="#">getExtent()</a> Get the duration (milliseconds) of media index
long	<a href="#">getLastAccessed()</a> Get last time media index was accessed (milliseconds)
boolean	<a href="#">isHitEnd()</a> Is media index complete
void	<a href="#">setChecksum(IRandomAccessReader file)</a> Set media index checksum.
void	<a href="#">setChecksum(String checksum)</a> Set media index checksum
void	<a href="#">setLastAccessed(long time)</a> Set last time media index accessed (milliseconds)
void	<a href="#">touch()</a> 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

<a href="#">MediaList</a>	<a href="#">resolveMediaList</a> ( <a href="#">IMediaListReader</a> mediaListReader, <a href="#">IMediaStream</a> 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	<a href="#"><code>close()</code></a> close file
<a href="#"><code>IHTTPStreamerSession</code></a>	<a href="#"><code>getHTTPStreamerSession()</code></a> Get the HTTPStreamerSession associated with this media list reader
String	<a href="#"><code>getMediaExtension()</code></a> Get media extension
<a href="#"><code>MediaList</code></a>	<a href="#"><code>getMediaList()</code></a> Get the MediaList object
String	<a href="#"><code>getPath()</code></a> Get abstract path to the media item
void	<a href="#"><code>init(IApplicationInstance appInstance, IMediaStream stream, String mediaReadType, String basePath, String mediaName, IHTTPStreamerSession httpStreamerSession)</code></a> Initialize mediaReader
boolean	<a href="#"><code>isOpen()</code></a> is file open
void	<a href="#"><code>open(String basePath, String name)</code></a> Open the file for reading
void	<a href="#"><code>setHTTPStreamerSession(IHTTPStreamerSession httpStreamerSession)</code></a> Set the HTTPStreamerSession associated with this media list reader
void	<a href="#"><code>setMediaReaderItem(MediaReaderItem mediaReaderItem)</code></a> Set the mediaReader item definition
void	<a href="#"><code>setProperties(WMSProperties properties)</code></a> 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	<a href="#"><u>CONTENTTYPE_MEDIA</u></a> Value: <b>1</b>
public static final	<a href="#"><u>CONTENTTYPE_MEDIALIST</u></a> Value: <b>2</b>
public static final	<a href="#"><u>DEFAULT_RANDOMACCESSREADER</u></a> Value: <b>com.wowza.io.DirectRandomAccessReader</b>
public static final	<a href="#"><u>PLAYEVENT_AFTERBUFFERFILL</u></a> Value: <b>5</b>
public static final	<a href="#"><u>PLAYEVENT_AFTERMETADATA</u></a> Value: <b>3</b>
public static final	<a href="#"><u>PLAYEVENT_BEFOREBUFFERFILL</u></a> Value: <b>4</b>
public static final	<a href="#"><u>PLAYEVENT_BEFOREMETADATA</u></a> Value: <b>2</b>
public static final	<a href="#"><u>PLAYEVENT_STARTPLAYBACK</u></a> Value: <b>1</b>
public static final	<a href="#"><u>SEEK_EXACT</u></a> Seek direction: closest frame (audio, video) (key, no-key) Value: <b>4</b>
public static final	<a href="#"><u>SEEK_KEYCLOSE</u></a> Seek direction: closest key frame Value: <b>3</b>
public static final	<a href="#"><u>SEEK_KEYDOWN</u></a> Seek direction: down to closets key frame Value: <b>2</b>

public static final	<a href="#">SEEK_KEYUP</a> Seek direction: up to closets key frame Value: <b>1</b>
public static final	<a href="#">SEEKTARGET_AUDIO</a> Value: <b>3</b>
public static final	<a href="#">SEEKTARGET_ENHANCED</a> Value: <b>4</b>
public static final	<a href="#">SEEKTARGET_VIDEOKEYFRAME</a> Value: <b>1</b>

## Method Summary

void	<a href="#">close()</a> close file
long	<a href="#">getDuration()</a> Get duration or time (milliseconds) of the media file
long	<a href="#">getLength()</a> Get the stream length in bytes
String	<a href="#">getMediaExtension()</a> Get media extension
java.util.List	<a href="#">getMetadata()</a> Get a collection of metadata packets in ByteBuffers for this file.
String	<a href="#">getPath()</a> Get abstract path to the media item
<a href="#">IMediaReaderStreamPosition</a>	<a href="#">getStreamPosition()</a> Get a reference to the current stream position
void	<a href="#">init()</a> ( <a href="#">IApplicationInstance</a> appInstance, <a href="#">IMediaStream</a> stream, String mediaReadType, String basePath, String mediaName) Initialize mediaReader
boolean	<a href="#">isOpen()</a> is file open
void	<a href="#">open()</a> (String basePath, String name) Open the file for reading
void	<a href="#">rewind()</a> rewind file to start
PlaylistSeekResult	<a href="#">seek()</a> (long timecode, int seektype) seek to timecode in file
int	<a href="#">sendZeroLengthPacket()</a> (int type, int timecode, boolean isAbsolute, <a href="#">IMediaStream</a> stream, java.io.OutputStream out, <a href="#">AMFObj</a> wmsObj, PlaylistCursor flvCursor, PlaylistWriteControl control, PlaylistReaderWriteResults results, long[] sizes, boolean isForceTCZero)

void	<a href="#"><code>setMediaReaderItem</code></a> ( <a href="#"><code>MediaReaderItem</code></a> mediaReaderItem) Set the mediaReader item definition
void	<a href="#"><code>setProperties</code></a> ( <a href="#"><code>WMSProperties</code></a> properties) Set the properties for this media reader
void	<a href="#"><code>setStreamPosition</code></a> ( <a href="#"><code>IMediaReaderStreamPosition</code></a> pos) Set the file position within the media file
void	<a href="#"><code>startPlayback</code></a> () Called each time the player being playback (before the buffer it filled)
int	<a href="#"><code>writeGeneratedKeyFrame</code></a> ( <a href="#"><code>IMediaStream</code></a> stream, java.io.OutputStream out, <a href="#"><code>AMFObj</code></a> wmsObjAudio, <a href="#"><code>AMFObj</code></a> wmsObjVideo, <a href="#"><code>AMFObj</code></a> wmsObjData, <a href="#"><code>PlaylistCursor</code></a> flvCursor, <a href="#"><code>PlaylistWriteControl</code></a> control, <a href="#"><code>PlaylistReaderWriteResults</code></a> results, long[] sizes, <a href="#"><code>FastPlaySettings</code></a> fastPlaySettings, boolean isForceTCZero) From current location in file generate a key frame (enhanced seek) and write it to out
int	<a href="#"><code>writePackets</code></a> ( <a href="#"><code>IMediaStream</code></a> stream, java.io.OutputStream out, <a href="#"><code>AMFObj</code></a> wmsObjAudio, <a href="#"><code>AMFObj</code></a> wmsObjVideo, <a href="#"><code>AMFObj</code></a> wmsObjData, <a href="#"><code>PlaylistCursor</code></a> flvCursor, <a href="#"><code>PlaylistWriteControl</code></a> control, <a href="#"><code>PlaylistReaderWriteResults</code></a> results, long[] sizes, <a href="#"><code>FastPlaySettings</code></a> fastPlaySettings, boolean isForceTCZero) From current location in file write packets to output
int	<a href="#"><code>writePackets</code></a> (java.util.List packetList, <a href="#"><code>PlaylistCursor</code></a> flvCursor, <a href="#"><code>PlaylistWriteControl</code></a> control, <a href="#"><code>PlaylistReaderWriteResults</code></a> results, long[] sizes, <a href="#"><code>FastPlaySettings</code></a> 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	<a href="#">onMediaReaderClose</a> ( <a href="#">IMediaReader</a> mediaReader, <a href="#">IMediaStream</a> stream) Called when media reader is closed
void	<a href="#">onMediaReaderCreate</a> ( <a href="#">IMediaReader</a> mediaReader) Called when media reader is created
void	<a href="#">onMediaReaderExtractMetaData</a> ( <a href="#">IMediaReader</a> mediaReader, <a href="#">IMediaStream</a> stream) Called after media reader metadata is extraced from the file
void	<a href="#">onMediaReaderInit</a> ( <a href="#">IMediaReader</a> mediaReader, <a href="#">IMediaStream</a> stream) Called after media reader is initialized
void	<a href="#">onMediaReaderOpen</a> ( <a href="#">IMediaReader</a> mediaReader, <a href="#">IMediaStream</a> 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	<a href="#">isValid()</a>
---------	---------------------------

---

### 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	<a href="#">AUDIOSAMPLEACCESS</a> Value: <b>2</b>
public static final	<a href="#">READACCESS</a> Value: <b>0</b>
public static final	<a href="#">VIDEOSAMPLEACCESS</a> Value: <b>3</b>
public static final	<a href="#">WRITEACCESS</a> Value: <b>1</b>

### Method Summary

void	<a href="#">addAudioCodecConfigPacket</a> (long timecode, <a href="#">AMFPacket</a> packet) Set audio codec configuration packet (needed for H.264/AAC playback)
void	<a href="#">addAudioData</a> (byte[] data, int offset, int size) Add data to current audio packet
void	<a href="#">addClientListener</a> ( <a href="#">IMediaStreamActionNotify</a> actionListener) Add client listener.
void	<a href="#">addClientListener</a> ( <a href="#">IMediaStreamActionNotify2</a> actionListener) Add client listener.
void	<a href="#">addClientListener</a> ( <a href="#">IMediaStreamActionNotify3</a> actionListener) Add client listener.
void	<a href="#">addDataData</a> (byte[] data, int offset, int size) Add data to current data packet
void	<a href="#">addVideoCodecConfigPacket</a> (long timecode, <a href="#">AMFPacket</a> packet) Set video codec configuration packet (needed for H.264/AAC playback)
void	<a href="#">addVideoData</a> (byte[] data, int offset, int size) Add data to current video packet

void	<a href="#"><u>addVideoH264SEIListener</u></a> ( <a href="#"><u>IMediaStreamH264SEINotify</u></a> h264SEIListener) Add an H.264 SEI listener.
void	<a href="#"><u>clear</u></a> ( ) Delete media file pointed to by this mediaStream (be careful)
void	<a href="#"><u>clearFastPlaySettings</u></a> ( ) Clear fastPlay settings
void	<a href="#"><u>clearLoggingValues</u></a> ( )
void	<a href="#"><u>close</u></a> ( ) Close mediaStream
void	<a href="#"><u>flush</u></a> ( ) Force publishing packets to be flushed from the input buffers to the output buffers
boolean[]	<a href="#"><u>getAccess</u></a> ( <a href="#"><u>IClient</u></a> client, String name) Get the read/write access to this stream for this client
<a href="#"><u>AMFPacket</u></a>	<a href="#"><u>getAudioCodecConfigPacket</u></a> (long timecode) Get audio codec configuration packet (needed for H.264/AAC playback)
int	<a href="#"><u>getAudioMissing</u></a> ( ) Get number of audio bytes missing from current audio packet
int	<a href="#"><u>getAudioSize</u></a> ( ) Get the size of the current audio packet that is being streamed from the client to the server
long	<a href="#"><u>getAudioTC</u></a> ( ) Get last absolute audio timecode (milliseconds) sent to mediaStream
int	<a href="#"><u>getBufferTime</u></a> ( ) Get buffer time for mediaStream (milliseconds)
byte[]	<a href="#"><u>getBurstStartStop</u></a> (boolean isStart) Get the dynamic streaming burst start/stop AMF packet
String	<a href="#"><u>getCacheName</u></a> ( ) not used
<a href="#"><u>IClient</u></a>	<a href="#"><u>getClient</u></a> ( ) Get parent client connection
int	<a href="#"><u>getClientId</u></a> ( ) Get parent client connection (id)
String	<a href="#"><u>getContextStr</u></a> ( ) Returns the stream context string in the form [application]/[appInstance]/[streamName].
int	<a href="#"><u>getDataMissing</u></a> ( ) Get number of data bytes missing from current audio packet
int	<a href="#"><u>getDataSize</u></a> ( ) Get the size of the current data packet that is being streamed from the client to the server
long	<a href="#"><u>getDataTC</u></a> ( ) Get last absolute data timecode (milliseconds) sent to mediaStream



int	<a href="#">getDataType()</a> Get the data packet type: (IVHost.CONTENTTYPE_DATA0 or IVHost.CONTENTTYPE_DATA3)
String	<a href="#">getDvrRecorder()</a> Get the DVR Recorder for this stream
<a href="#">ILiveStreamDvrRecorder</a>	<a href="#">getDvrRecorder(String name)</a> Get the DVR Recorder interface to a stream by name
String	<a href="#">getDvrRecorderList()</a> Get the comma separated list of DVR Recorder names being used by this stream (see conf/Dvr.xml)
String	<a href="#">getDvrRepeater()</a> Get the DVR repeater name for this stream
<a href="#">ElapsedTimer</a>	<a href="#">getElapsedTime()</a> Get the interface to the elapse timer
String	<a href="#">getExt()</a> Get media file extension
<a href="#">FastPlaySettings</a>	<a href="#">getFastPlaySettings()</a> Get current fastPlay settings
int	<a href="#">getHeaderSize()</a> Get the last packet header size (debugging)
<a href="#">IHTTPStreamerSession</a>	<a href="#">getHTTPStreamerSession()</a> Get the HTTPStreamer session associated with this stream
<a href="#">AMFPacket</a>	<a href="#">getLastKeyFrame()</a> Get most recent video key frame
<a href="#">AMFPacket</a>	<a href="#">getLastPacket()</a> Get most recent live packet
String	<a href="#">getLiveStreamPacketizer()</a> Get the live stream packetizer that this stream is using
<a href="#">ILiveStreamPacketizer</a>	<a href="#">getLiveStreamPacketizer(String name)</a> Get the LiveStreamPacketizer interface to a stream by name
String	<a href="#">getLiveStreamPacketizerList()</a> Get the comma separated list of LiveStreamPacketizers names being used by this stream (see conf/LiveStreamPacketizers.xml)
String	<a href="#">getLiveStreamRepeater()</a> Get the live stream repeater name for the stream
<a href="#">ILiveStreamTranscoder</a>	<a href="#">getLiveStreamTranscoder(String name)</a> Get a live stream transcoder for this stream by name
String	<a href="#">getLiveStreamTranscoderList()</a> Get the comma separated list of LiveStreamTranscoders names being used by this stream (see conf/LiveStreamTranscoders.xml)
java.util.Map	<a href="#">getLiveStreamTranscoders()</a> Get the list of transcoders for this stream.

long	<a href="#"><u>getMaxTimecode()</u></a> Get the timecode of the latest received packet
<a href="#"><u>IOPerformanceCounter</u></a>	<a href="#"><u>getMediaIOPerformance()</u></a> Get IO performance counter
<a href="#"><u>IMediaStreamMetaDataProvider</u></a>	<a href="#"><u>getMetaDataProvider()</u></a> Get the metaData provider
String	<a href="#"><u>getName()</u></a> Get stream name
com.wowza.wms.netconnection.INetConnection	<a href="#"><u>getNetConnection()</u></a> Get parent netConnection (future server to server communication)
<a href="#"><u>IMediaStreamPlay</u></a>	<a href="#"><u>getPlayer()</u></a> Get underlying player (IMediaStreamPlay) object
java.util.List	<a href="#"><u>getPlayPackets()</u></a> Get all available live packets
<a href="#"><u>WMSProperties</u></a>	<a href="#"><u>getProperties()</u></a> Get mediaStream properties
int	<a href="#"><u>getPublishAudioCodecId()</u></a> Get the codec id of the most recently published audio packet
int	<a href="#"><u>getPublishVideoCodecId()</u></a> Get the codec id of the most recently published video packet
String	<a href="#"><u>getQueryStr()</u></a> Get play/publish name query string.
int	<a href="#"><u>getReceiveVideoFPS()</u></a> Set frame per seconds for video (not currently implemented)
<a href="#"><u>AMFObj</u></a>	<a href="#"><u>getRespAMFAudioObj()</u></a> Get audio response channel object
<a href="#"><u>AMFObj</u></a>	<a href="#"><u>getRespAMFDataObj()</u></a> Get data response channel object
<a href="#"><u>AMFObj</u></a>	<a href="#"><u>getRespAMFVideoObj()</u></a> Get video response channel object
<a href="#"><u>RTPStream</u></a>	<a href="#"><u>getRTPStream()</u></a> Get the RTP based stream this stream is associated with
int	<a href="#"><u>getSrc()</u></a> Get the stream id
java.io.File	<a href="#"><u>getStreamFileForRead()</u></a> Get the File object to read from a stream (get stream name, ext and query from stream object)
java.io.File	<a href="#"><u>getStreamFileForRead(String name, String ext, String query)</u></a> Get the File object to read from a stream (specify name, ext and query)
java.io.File	<a href="#"><u>getStreamFileForWrite()</u></a> Get the File object to write to a stream (get stream name, ext and query from stream object)

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

boolean	<a href="#"><u>isPlay()</u></a> Is the stream a play stream (vs a publish stream)
boolean	<a href="#"><u>isPlaying()</u></a> Is mediaStream playing (or paused - false)
boolean	<a href="#"><u>isPublishStreamReady()</u></a> (boolean checkAudio, boolean checkVideo) Returns true if the publishing stream contains enough video/audio data to start playback
boolean	<a href="#"><u>isReceiveAudio()</u></a> Is client currently receiving audio.
boolean	<a href="#"><u>isReceiveVideo()</u></a> Is client currently receiving video.
boolean	<a href="#"><u>isRecord()</u></a> Is this stream being recorded to a file
boolean	<a href="#"><u>isSendPlayStopLogEvent()</u></a> Get need to send a log event for stop
boolean	<a href="#"><u>isSendPublishStopLogEvent()</u></a> Get need to send a log event for publishing
boolean	<a href="#"><u>isSendRecordStopLogEvent()</u></a> Get need to send a log event for recording
boolean	<a href="#"><u>isTranscodeResult()</u></a> Is this stream the result of a transcode operation.
boolean	<a href="#"><u>isVideoH264SEIListenerEmpty()</u></a> Is H.264 SEI listener list empty.
double	<a href="#"><u>length()</u></a> Get length/duration (seconds) of media file pointed to by mediaStream
void	<a href="#"><u>notifyActionOnCodecInfoAudio()</u></a> (com.wowza.wms.media.model.MediaCodecInfoAudio codecInfoAudio) Notify client listeners of audio codec information change
void	<a href="#"><u>notifyActionOnCodecInfoVideo()</u></a> (com.wowza.wms.media.model.MediaCodecInfoVideo codecInfoVideo) Notify client listeners of video codec information change
void	<a href="#"><u>notifyActionOnMetaData()</u></a> ( <a href="#"><u>AMFPacket</u></a> metaDataPacket) Notify client listeners of onMetaData change
void	<a href="#"><u>notifyActionPause()</u></a> (boolean isPause, long location) Notify client listeners of pause action
void	<a href="#"><u>notifyActionPauseRaw()</u></a> (boolean isPause, long location) Notify client listeners of pauseRaw action
void	<a href="#"><u>notifyActionPlay()</u></a> (String streamName, double playStart, double playLen, int playReset) Notify client listeners of play action
void	<a href="#"><u>notifyActionPublish()</u></a> (String streamName, boolean isRecord, boolean isAppend) Notify client listeners of publish action

void	<a href="#"><u>notifyActionSeek</u></a> (double location) Notify client listeners of seek action
void	<a href="#"><u>notifyActionStop</u></a> ( ) Notify client listeners of stop action
void	<a href="#"><u>notifyActionUnPublish</u></a> (String streamName, boolean isRecord, boolean isAppend) Notify client listeners of unpublish action
void	<a href="#"><u>notifyVideoH264Packet</u></a> ( <a href="#"><u>AMFPacket</u></a> packet, <a href="#"><u>com.wowza.wms.media.h264.H264SEIMessages</u></a> seiMessages) Notify H.264 SEI listener.
void	<a href="#"><u>packetComplete</u></a> ( ) Invoked by requestAdapter when at the end of a set of packets
void	<a href="#"><u>publish</u></a> ( ) Publish mediaStream
void	<a href="#"><u>putDvrRecorder</u></a> (String name, <a href="#"><u>ILiveStreamDvrRecorder</u></a> dvr) Add a live stream dvr to this stream
void	<a href="#"><u>putLiveStreamTranscoder</u></a> (String name, <a href="#"><u>ILiveStreamTranscoder</u></a> liveStreamTranscoder) Add a live stream transcoder to this stream
void	<a href="#"><u>registerCallback</u></a> (String handlerName, <a href="#"><u>IMediaStreamCallback</u></a> callback) Register a callback handler
void	<a href="#"><u>registerOnPlayStatus</u></a> ( <a href="#"><u>IMediaStreamCallback</u></a> callback) Register onPlayStatus handler
void	<a href="#"><u>registerOnStatus</u></a> ( <a href="#"><u>IMediaStreamCallback</u></a> callback) Register onStatus handler
void	<a href="#"><u>removeClientListener</u></a> ( <a href="#"><u>IMediaStreamActionNotify</u></a> actionListener) Remove client listener.
void	<a href="#"><u>removeClientListener</u></a> ( <a href="#"><u>IMediaStreamActionNotify2</u></a> actionListener) Remove client listener.
void	<a href="#"><u>removeClientListener</u></a> ( <a href="#"><u>IMediaStreamActionNotify3</u></a> actionListener) Remove client listener.
<a href="#"><u>ILiveStreamDvrRecorder</u></a>	<a href="#"><u>removeDvrRecorder</u></a> (String name) Remove a live stream dvr by name
<a href="#"><u>ILiveStreamTranscoder</u></a>	<a href="#"><u>removeLiveStreamTranscoder</u></a> (String name) Remove a live stream transcoder by name
void	<a href="#"><u>removeVideoH264SEIListener</u></a> ( <a href="#"><u>IMediaStreamH264SEINotify</u></a> h264SEIListener) Remove an H.264 SEI listener.
void	<a href="#"><u>send</u></a> (String handlerName) Call client side NetStream method/handler with no parameters
void	<a href="#"><u>send</u></a> (String handlerName, Object[] params) Call client side NetStream method/handler

void	<a href="#"><code>sendAMF3</code></a> (String handlerName) Call client side NetStream method/handler with no parameters.
void	<a href="#"><code>sendAMF3</code></a> (String handlerName, Object[] params) Call client side NetStream method/handler.
int	<a href="#"><code>sendControlBytes</code></a> (int controlType, java.io.OutputStream out) Send playback control bytes.
void	<a href="#"><code>sendDirect</code></a> (String handlerName) Call client side NetStream method/handler and send event to underlying stream (will record event)
void	<a href="#"><code>sendDirect</code></a> (String handlerName, Object[] params) Call client side NetStream method/handler and send event to underlying stream (will record event)
void	<a href="#"><code>sendDirectAMF3</code></a> (String handlerName) Call client side NetStream method/handler and send event to underlying stream (will record event).
void	<a href="#"><code>sendDirectAMF3</code></a> (String handlerName, Object[] params) Call client side NetStream method/handler and send event to underlying stream (will record event).
int	<a href="#"><code>sendLivePlaySeek</code></a> (java.io.OutputStream out, String name, long timecode) Send onStatus(NetStream.Seek.Notify) event
int	<a href="#"><code>sendLivePlayStart</code></a> (java.io.OutputStream out, String name, long timecode, long timecodeOffset) Send onStatus(NetStream.Play.Start) event
int	<a href="#"><code>sendLivePlaySwitch</code></a> (java.io.OutputStream out, String name, long timecode) Send onStatus(NetStream.Play.Transition) event
int	<a href="#"><code>sendPauseNotify</code></a> (long timecode, String name) Send onStatus(NetStream.Pause.Notify) event
int	<a href="#"><code>sendPauseNotify</code></a> (java.io.OutputStream out, long timecode, String name) Send onStatus(NetStream.Pause.Notify) event.
int	<a href="#"><code>sendPlayReset</code></a> (java.io.OutputStream out, String name) Send onStatus(NetStream.Play.Reset) event.
int	<a href="#"><code>sendPlayReset</code></a> (String name) Send onStatus(NetStream.Play.Reset) event
int	<a href="#"><code>sendPlaySeek</code></a> (long location, long seekLocation, String name) Send onStatus(NetStream.Seek.Notify) event.
int	<a href="#"><code>sendPlaySeek</code></a> (java.io.OutputStream out, long location, long seekLocation, String name) Send onStatus(NetStream.Seek.Notify) event.
int	<a href="#"><code>sendPlaySeek</code></a> (java.io.OutputStream out, long location, long seekLocation, String name, java.util.List seekTypes) Send onStatus(NetStream.Seek.Notify) event.

int	<a href="#"><code>sendPlayStart</code></a> (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	<a href="#"><code>sendPlayStart</code></a> (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	<a href="#"><code>sendPlayStart</code></a> (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	<a href="#"><code>sendPlayStart</code></a> (String name, long timecode) Send onStatus(NetStream.Play.Start) event
int	<a href="#"><code>sendPlayStatus</code></a> (long timecode, int statusType, double duration, double bytesSent) Send onPlayStatus(NetStream.Play.Switch, NetStream.Play.Complete, NetStream.Play.Stop) event
int	<a href="#"><code>sendPlayStatus</code></a> (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	<a href="#"><code>sendPlayStop</code></a> (long location, String name) Send onStatus(NetStream.Play.Stop) event
int	<a href="#"><code>sendPlayStop</code></a> (java.io.OutputStream out, long location, String name) Send onStatus(NetStream.Play.Stop) event.
int	<a href="#"><code>sendPlaySwitch</code></a> (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	<a href="#"><code>sendPlaySwitch</code></a> (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	<a href="#"><code>sendStreamNotFound</code></a> (java.io.OutputStream out, String name) Send onStatus(NetStream.Play.StreamNotFound) event.
int	<a href="#"><code>sendStreamNotFound</code></a> (String name) Send onStatus(NetStream.Play.StreamNotFound) event
int	<a href="#"><code>sendUnpauseNotify</code></a> (long location, String name) Send onStatus(NetStream.Unpause.Notify) event
int	<a href="#"><code>sendUnpauseNotify</code></a> (java.io.OutputStream out, long location, String name) Send onStatus(NetStream.Unpause.Notify) event.
int	<a href="#"><code>sendUnpauseNotify</code></a> (java.io.OutputStream out, long location, String name, java.util.List seekTypes) Send onStatus(NetStream.Unpause.Notify) event
int	<a href="#"><code>sendVODPlaySwitch</code></a> (java.io.OutputStream out, String name, long timecode) Send onStatus(NetStream.Play.Transition) event

void	<a href="#"><u>setAppend</u></a> (boolean isAppend) Set is append to media file (only valid if isRecord)
void	<a href="#"><u>setAudioSize</u></a> (int audioSize) Set the size of the current audio packet that is being streamed from the client to the server
void	<a href="#"><u>setAudioTC</u></a> (long audioTC) Set last absolute audio timecode (milliseconds) sent to mediaStream
void	<a href="#"><u>setAudioTC</u></a> (long audioTC, boolean isAbsolute) Set last absolute audio timecode (milliseconds) sent to mediaStream
void	<a href="#"><u>setBufferTime</u></a> (int bufferTime) Set buffer time for mediaStream (milliseconds)
void	<a href="#"><u>setClient</u></a> ( <a href="#"><u>IClient</u></a> client) Set parent client connection
void	<a href="#"><u>setClustered</u></a> (boolean isClustered) not used
void	<a href="#"><u>setDataSize</u></a> (int dataSize) Set the size of the current data packet that is being streamed from the client to the server
void	<a href="#"><u>setDataTC</u></a> (long dataTC) Set last absolute data timecode (milliseconds) sent to mediaStream
void	<a href="#"><u>setDataTC</u></a> (long dataTC, boolean isAbsolute) Set last absolute data timecode (milliseconds) sent to mediaStream
void	<a href="#"><u>setDataType</u></a> (int dataType) Set the data packet type: (IVHost.CONTENTTYPE_DATA0 or IVHost.CONTENTTYPE_DATA3)
void	<a href="#"><u>setDvrRecorder</u></a> (String recorderName) Set the DVR Recorder that this stream is using
void	<a href="#"><u>setDvrRecorderList</u></a> (String recorderList) Set the comma separated list of DVR Recorder names being used by this stream (see conf/Dvr.xml)
void	<a href="#"><u>setExt</u></a> (String ext) Set media file extension
void	<a href="#"><u>setFastPlaySettings</u></a> ( <a href="#"><u>FastPlaySettings</u></a> fastPlaySettings) Set fastPlay settings
void	<a href="#"><u>setHeaderSize</u></a> (int headerSize) Set the last packet header size (debugging)
void	<a href="#"><u>setHTTPStreamerSession</u></a> ( <a href="#"><u>IHTTPStreamerSession</u></a> httpStreamerSession) Set the HTTPStreamer session associated with this stream
void	<a href="#"><u>setIsPlaying</u></a> (boolean isPlaying) Set is mediaStream playing
void	<a href="#"><u>setLiveStreamPacketizer</u></a> (String liveStreamPacketizer) Set the live stream packetizer that this stream is using



void	<a href="#"><u>setLiveStreamPacketizerList</u></a> (String liveStreamPacketizerList) Set the comma separated list of LiveStreamPacketizers names being used by this stream (see conf/LiveStreamPacketizers.xml)
void	<a href="#"><u>setLiveStreamRepeater</u></a> (String liveStreamRepeater) Set the live stream repeater name for the stream
void	<a href="#"><u>setLiveStreamTranscoderList</u></a> (String liveStreamTranscoderList) Set the comma separated list of LiveStreamTranscoders names being used by this stream (see conf/LiveStreamTranscoders.xml)
void	<a href="#"><u>setMediaCasterPlay</u></a> (boolean isMediaCasterPlay) Is MediaCaster play enabled (if true, will trigger MediaCaster startup)
void	<a href="#"><u>setMergeOnMetadata</u></a> (boolean mergeOnMetadata) If true, merge incoming onMetadata events with the current onMetadata event data.
void	<a href="#"><u>setMetaDataProvider</u></a> ( <a href="#"><u>IMediaStreamMetaDataProvider</u></a> metaDataProvider) Set the metaData provider
void	<a href="#"><u>setName</u></a> (String name) Set stream name
void	<a href="#"><u>setName</u></a> (String name, String ext) Set stream name and extension.
void	<a href="#"><u>setName</u></a> (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	<a href="#"><u>setNetConnection</u></a> (com.wowza.wms.netconnection.INetConnection netConnection) Set parent netConnection (future server to server communication)
void	<a href="#"><u>setOpen</u></a> (boolean isOpen) Set mediaStream open
void	<a href="#"><u>setPlay</u></a> (boolean isPlay) Set is the stream a play stream (vs a publish stream)
void	<a href="#"><u>setPlayer</u></a> ( <a href="#"><u>IMediaStreamPlay</u></a> player) Set underlying player (IMediaStreamPlay) object
void	<a href="#"><u>setPublishAudioCodecId</u></a> (int publishAudioCodecId) Set the codec id of the most recently published audio packet
void	<a href="#"><u>setPublishVideoCodecId</u></a> (int publishVideoCodecId) Set the codec id of the most recently published video packet
void	<a href="#"><u>setQueryStr</u></a> (String queryStr) Set play/publish name query string.
void	<a href="#"><u>setReceiveAudio</u></a> (boolean receiveAudio) Set receive audio
void	<a href="#"><u>setReceiveVideo</u></a> (boolean receiveVideo) Set receive video
void	<a href="#"><u>setReceiveVideoFPS</u></a> (int receiveVideoFPS) Set frame per second for video (not currently implemented)

void	<a href="#"><u>setRecord</u></a> (boolean isRecord) Set is the stream being recorded
void	<a href="#"><u>setRTPStream</u></a> ( <a href="#"><u>RTPStream</u></a> rtpStream) Set the RTP based stream this stream is associated with
void	<a href="#"><u>setSendPlayStopLogEvent</u></a> (boolean sendPlayStopLogEvent) Set need to send a log event for stop
void	<a href="#"><u>setSendPublishStopLogEvent</u></a> (boolean sendPlayStopLogEvent) Set need to send a log event for publishing
void	<a href="#"><u>setSendRecordStopLogEvent</u></a> (boolean sendPlayStopLogEvent) Set need to send a log event for recording
void	<a href="#"><u>setSrc</u></a> (int src) Set stream id
void	<a href="#"><u>setStreamType</u></a> (String streamType) Set mediaStream streamType.
void	<a href="#"><u>setTranscodeResult</u></a> (boolean isTranscodeResult) Is this stream the result of a transcode operation.
void	<a href="#"><u>setVideoSize</u></a> (int videoSize) Set the size of the current video packet that is being streamed from the client to the server
void	<a href="#"><u>setVideoTC</u></a> (long videoTC) Set last absolute video timecode (milliseconds) sent to mediaStream
void	<a href="#"><u>setVideoTC</u></a> (long videoTC, boolean isAbsolute) Set last absolute video timecode (milliseconds) sent to mediaStream
void	<a href="#"><u>shutdown</u></a> ( ) shutdown or close this mediaStream
long	<a href="#"><u>size</u></a> ( ) Get size (bytes) of media file pointed to by mediaStream
void	<a href="#"><u>startAudioPacket</u></a> ( ) Called when an audio packet is first being populated with data
void	<a href="#"><u>startDataPacket</u></a> ( ) Called when a data packet is first being populated with data
void	<a href="#"><u>startPublishing</u></a> ( ) Start publishing live stream
void	<a href="#"><u>startVideoPacket</u></a> ( ) Called when a video packet is first being populated with data
void	<a href="#"><u>stopName</u></a> (String name, String oldName, String ext, String queryStr, double playStart, double playLen, int playTransition) Stop stream name
void	<a href="#"><u>stopPublishing</u></a> ( ) Stop publishing live stream

void	<a href="#"><code>switchName</code></a> (String name, String oldName, String ext, String queryStr, double playStart, double playLen, int playTransition) Switch to stream name
void	<a href="#"><code>trim</code></a> () Trim mediaStream.
void	<a href="#"><code>unregisterCallback</code></a> (String handlerName) Unregister a callback handler
void	<a href="#"><code>unregisterOnPlayStatus</code></a> ( <a href="#"><code>IMediaStreamCallback</code></a> callback) Unregister onPlayStatus handler
void	<a href="#"><code>unregisterOnStatus</code></a> ( <a href="#"><code>IMediaStreamCallback</code></a> callback) Unregister onStatus handler
void	<a href="#"><code>updateLoggingDuration</code></a> ( ) Update logging.MDC with mediaStream logging information
void	<a href="#"><code>updateLoggingValues</code></a> ( ) 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&param2=data2, query string is "param1=data1&param2=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&param2=data2, query string is "param1=data1&param2=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	<a href="#">onPause</a> ( <a href="#">IMediaStream</a> stream, boolean isPause, double location) Triggered on mediaStream pause
void	<a href="#">onPlay</a> ( <a href="#">IMediaStream</a> stream, String streamName, double playStart, double playLen, int playReset) Triggered on mediaStream play
void	<a href="#">onPublish</a> ( <a href="#">IMediaStream</a> stream, String streamName, boolean isRecord, boolean isAppend) Triggered on mediaStream publish
void	<a href="#">onSeek</a> ( <a href="#">IMediaStream</a> stream, double location) Triggered on mediaStream seek
void	<a href="#">onStop</a> ( <a href="#">IMediaStream</a> stream) Triggered on mediaStream stop
void	<a href="#">onUnPublish</a> ( <a href="#">IMediaStream</a> 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	<a href="#">onMetaData</a> ( <a href="#">IMediaStream</a> stream, <a href="#">AMFPacket</a> metaDataPacket) Triggered when a published streams metadata is set or changes
void	<a href="#">onPauseRaw</a> ( <a href="#">IMediaStream</a> 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	<a href="#">onCodecInfoAudio</a> ( <a href="#">IMediaStream</a> stream, <a href="#">com.wowza.wms.media.model.MediaCodecInfoAudio</a> codecInfoAudio) Triggered when publishing stream receives codec information.
void	<a href="#">onCodecInfoVideo</a> ( <a href="#">IMediaStream</a> stream, <a href="#">com.wowza.wms.media.model.MediaCodecInfoVideo</a> 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(<a href="#">IMediaStream</a> stream, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> 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	<a href="#"><code>streamToFileForRead(IMediaStream stream)</code></a> Get the File object to read from a stream (get stream name, ext and query from stream object)
java.io.File	<a href="#"><code>streamToFileForRead(IMediaStream stream, String name, String ext, String query)</code></a> Get the File object to read from a stream (specify name, ext and query)
java.io.File	<a href="#"><code>streamToFileForWrite(IMediaStream stream)</code></a> Get the File object to write to a stream (get stream name, ext and query from stream object)
java.io.File	<a href="#"><code>streamToFileForWrite(IMediaStream stream, String name, String ext, String query)</code></a> 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	<a href="#">onVideoH264Packet</a> ( <a href="#">IMediaStream</a> stream, <a href="#">AMFPacket</a> packet, <a href="#">com.wowza.wms.media.h264.H264SEIMessages</a> 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

<a href="#">IMediaCaster</a>	<a href="#">getMediaCaster()</a>
void	<a href="#">setMediaCasterItem(<a href="#">MediaCasterItem</a> mediaCasterItem)</a>

---

### 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	<a href="#"><code>onStreamStart</code></a> (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 [IAApplication.setStreamNameAliasProvider\(IMediaStreamNameAliasProvider streamNameAliasProvider\)](#). See updated interface [IMediaStreamNameAliasProvider2](#) which provides more complete context information.

### Method Summary

String	<a href="#">resolvePlayAlias</a> ( <a href="#">IAApplicationInstance</a> appInstance, String name) Called to resolve a play alias
String	<a href="#">resolveStreamAlias</a> ( <a href="#">IAApplicationInstance</a> appInstance, String name) Called to resolve a stream alias

### Methods

#### resolvePlayAlias

```
public String resolvePlayAlias(IAApplicationInstance 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(IAApplicationInstance 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	<a href="#">resolvePlayAlias</a> ( <a href="#">IApplicationInstance</a> appInstance, String name, <a href="#">IClient</a> client) Resolve play alias for RTMP streaming
String	<a href="#">resolvePlayAlias</a> ( <a href="#">IApplicationInstance</a> appInstance, String name, <a href="#">IHTTPStreamerSession</a> httpSession) Resolve play alias for HTTP streaming.
String	<a href="#">resolvePlayAlias</a> ( <a href="#">IApplicationInstance</a> appInstance, String name, <a href="#">ILiveStreamPacketizer</a> liveStreamPacketizer) Resolve play alias for live stream packetizer
String	<a href="#">resolvePlayAlias</a> ( <a href="#">IApplicationInstance</a> appInstance, String name, <a href="#">RTPSession</a> rtpSession) Resolve play alias for RTSP/RTP streaming
String	<a href="#">resolveStreamAlias</a> ( <a href="#">IApplicationInstance</a> appInstance, String name, <a href="#">IMediaCaster</a> 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	<a href="#">onMediaStreamCreate</a> ( <a href="#">IMediaStream</a> stream) Triggered when mediaStream created
void	<a href="#">onMediaStreamDestroy</a> ( <a href="#">IMediaStream</a> 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	<a href="#"><u>PAUSE_PAUSE</u></a> Pause type: pause Value: <b>1</b>
public static final	<a href="#"><u>PAUSE_PLAY</u></a> Pause type: play Value: <b>0</b>
public static final	<a href="#"><u>PAUSE_TOGGLE</u></a> Pause type: toggle Value: <b>-1</b>
public static final	<a href="#"><u>PLAYSIZES_AUDIO_BYTES</u></a> IMediaReader sizes array: audio byte count Value: <b>0</b>
public static final	<a href="#"><u>PLAYSIZES_AUDIO_COUNT</u></a> IMediaReader sizes array: audio packet count Value: <b>1</b>
public static final	<a href="#"><u>PLAYSIZES_DATA_BYTES</u></a> IMediaReader sizes array: data byte count Value: <b>4</b>
public static final	<a href="#"><u>PLAYSIZES_DATA_COUNT</u></a> IMediaReader sizes array: data packet count Value: <b>5</b>
public static final	<a href="#"><u>PLAYSIZES_LOSS_BYTES</u></a> IMediaReader sizes array: data byte count Value: <b>6</b>
public static final	<a href="#"><u>PLAYSIZES_LOSS_COUNT</u></a> IMediaReader sizes array: data packet count Value: <b>7</b>
public static final	<a href="#"><u>PLAYSIZES_SIZE</u></a> IMediaReader sizes array: size of sizes array long[PLAYSIZES_SIZE] Value: <b>8</b>
public static final	<a href="#"><u>PLAYSIZES_VIDEO_BYTES</u></a> IMediaReader sizes array: video byte count Value: <b>2</b>

public static final	<a href="#">PLAYSIZES_VIDEO_COUNT</a> IMediaReader sizes array: video packet count Value: <b>3</b>
public static final	<a href="#">PLAYSTATUSTYPE_COMPLETE</a> onPlayStatus type: complete Value: <b>2</b>
public static final	<a href="#">PLAYSTATUSTYPE_STOP</a> onPlayStatus type: stop Value: <b>3</b>
public static final	<a href="#">PLAYSTATUSTYPE_SWITCH</a> onPlayStatus type: switch Value: <b>1</b>

## Method Summary

void	<a href="#">close()</a> Close mediaStreamPlay
<a href="#">IMediaStream</a>	<a href="#">getParent()</a> Get the parent media stream object
void	<a href="#">init()</a> ( <a href="#">IMediaStream</a> parent, <a href="#">MediaStreamMap</a> streams) Initialize mediaStreamPlayer
void	<a href="#">initLiveStreamRepeating()</a> (String liveStreamPacketizer, String liveStreamRepeater) Initialize this stream for live stream repeating
void	<a href="#">interruptPlay()</a> Interrupt play to perform operation.
double	<a href="#">length()</a> Get stream length/seconds (seconds)
void	<a href="#">pause()</a> (int pauseType, long timecode) pause mediaPlayStream
void	<a href="#">pauseRaw()</a> (int pauseType, long timecode) pauseRaw mediaPlayStream
boolean	<a href="#">play()</a> Return true if there are packets to play
int	<a href="#">play()</a> (java.io.OutputStream out, <a href="#">AMFObj</a> wmsObjAudio, <a href="#">AMFObj</a> wmsObjVideo, <a href="#">AMFObj</a> wmsObjData, long[] sizes) Write new packets or play packets
void	<a href="#">reset()</a> (boolean isReset) Reset mediaStreamPlayer
void	<a href="#">resetNoLookup()</a> Reset but do not lookup current position in live stream
void	<a href="#">seek()</a> (int location) seek mediaPlayStream

void	<a href="#"><code>setBufferTime</code></a> (int bufferTime) Set buffer time
void	<a href="#"><code>setName</code></a> (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	<a href="#"><code>shutdown</code></a> () Shutdown mediaStreamPlayer
long	<a href="#"><code>size</code></a> () Get stream media file size
void	<a href="#"><code>startPlay</code></a> () Start playing stream
void	<a href="#"><code>stopName</code></a> (String name, String oldName, String ext, String queryStr, double playStart, double playLen, int playTransition) Stop stream name
void	<a href="#"><code>switchName</code></a> (String name, String oldName, String ext, String queryStr, double playStart, double playLen, int playTransition) Switch to stream name
void	<a href="#"><code>switchPlay</code></a> ()
void	<a href="#"><code>updateLoggingValues</code></a> () 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	<a href="#">resetTimecodes()</a>
------	----------------------------------

---

## 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	<a href="#"><code>getDuration()</code></a> Get the recorded duration of the file in seconds
boolean	<a href="#"><code>isVersionFile()</code></a> Return true if the old file is to be versioned
boolean	<a href="#"><code>isWaitForVideoKeyFrame()</code></a> get wait for key frame
void	<a href="#"><code>putMetaData(String name, AMFData value)</code></a> Add metadata to the metadata packet.
void	<a href="#"><code>setMediaWriterItem(MediaWriterItem mediaWriterItem)</code></a> Set the media write definition
void	<a href="#"><code>setParent(IMediaStream parent)</code></a> Set the parent stream for this media write object
void	<a href="#"><code>setVersionFile(boolean versionFile)</code></a> Set to true if the old file is to be versioned
void	<a href="#"><code>setWaitForVideoKeyFrame(boolean waitForVideoKeyFrame)</code></a> Set to true if you want the recorder to skip opening frames until it hits a key frame
void	<a href="#"><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></a> 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	<a href="#">onFLVAddMetadata</a> ( <a href="#">IMediaStream</a> stream, java.util.Map extraMetadata) Called just before metadata is written to the file (FLV only)
void	<a href="#">onWriteComplete</a> ( <a href="#">IMediaStream</a> 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)

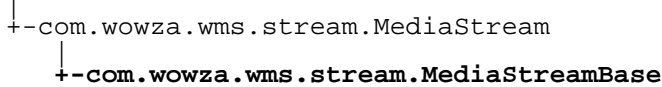
**Parameters:**

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	<a href="#"><u>p</u></a> Deprecated.
public static	<a href="#"><u>sinfo</u></a> Deprecated.

#### Fields inherited from class com.wowza.wms.stream.MediaStream

actionListeners, audioBytes, audioSize, audioTC, BASE\_STREAM\_EXT, bufferTime, callbacks, client, dataBytes, dataSize, dataTC, dataType, dvrLock, dvrMap, dvrPlayer, dvrRecorder, dvrRecorderList, dvrRepeater, elapsedTime, ext, fastPlaySettings, h264SEIListeners, headerSize, httpStreamerSession, ID3\_STREAM\_EXT, isAppend, isClustered, isMediaCasterPlay, isOpen, isPlay, isPlaying, isRecord, isTranscodeResult, liveStreamPacketizer, liveStreamPacketizerList, liveStreamRepeater, liveStreamTranscoderList, lock, 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	<a href="#"><u>MediaStreamBase</u></a> ( )
--------	--

### Method Summary

void	<a href="#"><u>init</u></a> ( <a href="#"><u>MediaStreamMap</u></a> parent, int src, <a href="#"><u>WMSProperties</u></a> properties)
void	<a href="#"><u>publish</u></a> ( )

void	<a href="#">trim()</a>
------	------------------------

#### 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	<a href="#"><u>appInstance</u></a>
protected	<a href="#"><u>dvrRecorders</u></a>
protected	<a href="#"><u>liveStreamPacketizers</u></a>
public static final	<a href="#"><u>MAXSTREAMINDEX</u></a> Value: <b>65536</b>
protected	<a href="#"><u>mediaStreamListeners</u></a>
protected	<a href="#"><u>nameGroupId</u></a>
protected	<a href="#"><u>nameGroups</u></a>
protected	<a href="#"><u>nextStreamId</u></a>
protected	<a href="#"><u>packetizerLicenses</u></a>
protected	<a href="#"><u>streamLicenses</u></a>
protected	<a href="#"><u>streamLock</u></a>
protected	<a href="#"><u>streamNames</u></a>
protected	<a href="#"><u>streamNamesLock</u></a>
protected	<a href="#"><u>streamNameToGroup</u></a>
protected	<a href="#"><u>streams</u></a>

#### Constructor Summary



public	<a href="#">MediaStreamMap</a> ( <a href="#">IApplicationInstance</a> appInstance) Create empty MediaStreamMap collection
--------	--

## Method Summary

LicenseHolder	<a href="#">addLicense</a> ( <a href="#">ILiveStreamPacketizer</a> liveStreamPacketizer, int licenseType)
LicenseHolder	<a href="#">addLicense</a> ( <a href="#">IMediaStream</a> stream, int licenseType)
void	<a href="#">addMediaStreamListener</a> ( <a href="#">IMediaStreamNotify</a> mediaStreamListener) Add a media stream listener.
MediaStreamMapGroup	<a href="#">addNameGroup</a> (MediaStreamMapGroup newGroup)
void	<a href="#">broadcastPlayMessage</a> ( <a href="#">IMediaStream</a> stream, long timecode, <a href="#">java.nio.ByteBuffer</a> msg, int objectEncoding) Send a broadcast message to all play stream that are listening to this live published stream.
int	<a href="#">broadcastGetObjectEncoding</a> ( <a href="#">IMediaStream</a> stream) Get the minimum object encoding level for the clients playing this stream.
void	<a href="#">clearStreamName</a> (String name) Unregister a published live media stream name.
void	<a href="#">clearStreamName</a> (String name, <a href="#">IMediaStream</a> stream) Unregister a published live media stream name.
<a href="#">IApplicationInstance</a>	<a href="#">getAppInstance</a> () Get the parent applicationInstance.
String	<a href="#">getAppInstanceName</a> () Get the name of the parent applicationInstance.
String	<a href="#">getAppName</a> () Get the name of the parent application.
int	<a href="#">getCount</a> () Get the total number of streams stored in the mediaStreamMap
<a href="#">ILiveStreamDvrRecorder</a>	<a href="#">getDvrRecorder</a> (String streamName, String recorderName, boolean doCreate) Get a DVR recorder by name and recorder name
java.util.List	<a href="#">getDvrRecorders</a> () Returns a list of <a href="#">ILiveStreamDvrRecorder</a> objects
<a href="#">ILiveStreamPacketizer</a>	<a href="#">getLiveStreamPacketizer</a> (String streamName, String packetizerName, boolean doCreate) Get a live stream packetizer by name and packetizer id
Object	<a href="#">getLiveStreamPacketizerLock</a> () Get the lock to the live stream packetizer system
MediaStreamMapGroup	<a href="#">getNameGroupByGroupName</a> (String groupName)
java.util.Set	<a href="#">getNameGroups</a> ()

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

void	<a href="#"><u>notifyMediaStreamDestroy</u></a> ( <a href="#"><u>IMediaStream</u></a> mediaStream) Notify all media stream listeners that a media stream object is being destroyed.
void	<a href="#"><u>notifyPlayPublish</u></a> ( <a href="#"><u>IMediaStream</u></a> 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	<a href="#"><u>notifyPlayUnpublish</u></a> ( <a href="#"><u>IMediaStream</u></a> 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	<a href="#"><u>notifyPlayUnpublish</u></a> (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	<a href="#"><u>removeDvrRecorder</u></a> (String streamName)
<a href="#"><u>ILiveStreamDvrRecorder</u></a>	<a href="#"><u>removeDvrRecorder</u></a> (String streamName, String recorderName) Remove DVR Recorder
void	<a href="#"><u>removeLiveStreamPacketizer</u></a> (String streamName) Remove all live stream packetizers for this stream name
<a href="#"><u>ILiveStreamPacketizer</u></a>	<a href="#"><u>removeLiveStreamPacketizer</u></a> (String streamName, String packetizerName) Remove live stream packetizer
void	<a href="#"><u>removeMediaStreamListener</u></a> ( <a href="#"><u>IMediaStreamNotify</u></a> mediaStreamListener) Remove a media stream listener.
MediaStreamMapGroup	<a href="#"><u>removeNameGroup</u></a> (int groupId)
MediaStreamMapGroup	<a href="#"><u>removeNameGroup</u></a> (MediaStreamMapGroup nameGroup)
void	<a href="#"><u>removeStream</u></a> ( <a href="#"><u>IClient</u></a> client, int index) Remove a stream associated with a client connection
void	<a href="#"><u>removeStream</u></a> (com.wowza.wms.netconnection.INetConnection netConnection, int index) Remove a stream associated with a netConnection object
void	<a href="#"><u>removeStream</u></a> (int index) Remove a clientless media stream
void	<a href="#"><u>setStreamName</u></a> ( <a href="#"><u>IMediaStream</u></a> stream, String name) Insert live media stream into the mediaStreamMap by name.
long	<a href="#"><u>streamToIndex</u></a> ( <a href="#"><u>IMediaStream</u></a> 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	<a href="#">onDvrStreamManagerCreate</a> ( <a href="#">IDvrStreamManager</a> dvrMgr) Called when DVR stream manager created but before it is initialized.
void	<a href="#">onDvrStreamManagerDestroy</a> ( <a href="#">IDvrStreamManager</a> dvrMgr) Called when DVR stream manager destroyed.
void	<a href="#">onDvrStreamManagerInit</a> ( <a href="#">IDvrStreamManager</a> 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	<a href="#"><u>canRecordAudio()</u></a> True if audio is being recorded
boolean	<a href="#"><u>canRecordData()</u></a> True if data is being recorded
boolean	<a href="#"><u>canRecordVideo()</u></a> True if video is being recorded
<a href="#"><u>IApplicationInstance</u></a>	<a href="#"><u>getAppInstance()</u></a> Get associated applicationInstance.
<a href="#"><u>IDvrStreamManager</u></a>	<a href="#"><u>getDvrManager()</u></a> Get DVR recorder's associated DVR stream manager
int	<a href="#"><u>getDvrRecorderId()</u></a> Get the DVR recorder id
LicenseHolder	<a href="#"><u>getLicenseHolder()</u></a>
<a href="#"><u>WMSProperties</u></a>	<a href="#"><u>getProperties()</u></a> Get properties
String	<a href="#"><u>getRecordingName()</u></a>
<a href="#"><u>IMediaStream</u></a>	<a href="#"><u>getStream()</u></a> Get the current stream that is being recorded
void	<a href="#"><u>handlePacket()</u></a> ( <a href="#"><u>IMediaStream</u></a> stream, <a href="#"><u>AMFPacket</u></a> packet) Called to handle an incoming packet
void	<a href="#"><u>init()</u></a> (String streamName, String recorderName, <a href="#"><u>IApplicationInstance</u></a> appInstance, <a href="#"><u>DvrRecorderItem</u></a> dvrRecorderItem) Initialize DVR recorder.
boolean	<a href="#"><u>isActive()</u></a> Is the DVR recorder active
boolean	<a href="#"><u>isRecording()</u></a> Is this stream currently recording.
boolean	<a href="#"><u>isRecordingPaused()</u></a> Is this stream currently paused from recording.

boolean	<a href="#"><u>pauseRecording()</u></a> Request that stream recording pause.
void	<a href="#"><u>resetStream()</u></a> ( <a href="#"><u>IMediaStream</u></a> stream) Called when something happens that forces the stream to reset
boolean	<a href="#"><u>resumeRecording()</u></a> Request that stream recording resume.
void	<a href="#"><u>setDvrRecorderId()</u></a> (int liveStreamId) Set the DVR recorder id
void	<a href="#"><u>setRecordAudio()</u></a> (boolean recordAudio) Set to true to record audio
void	<a href="#"><u>setRecordData()</u></a> (boolean recordVideo) Set to true to record data
void	<a href="#"><u>setRecordingName()</u></a> (String name)
void	<a href="#"><u>setRecordVideo()</u></a> (boolean recordVideo) Set to true to record video
void	<a href="#"><u>setStartRecordingOnStartup()</u></a> (boolean shouldStartRecordingOnStartup) Set recording behavior of DVR Manager on startup.
boolean	<a href="#"><u>shouldStartRecordingOnStartup()</u></a> Should DVR start recording when packets start flowing.
void	<a href="#"><u>shutdown()</u></a> Called to shutdown the DVR recorder
<a href="#"><u>IDvrStreamManager</u></a>	<a href="#"><u>startRecording()</u></a> Request that stream recording start.
void	<a href="#"><u>startStream()</u></a> ( <a href="#"><u>IMediaStream</u></a> stream) Called when the stream starts
boolean	<a href="#"><u>stopRecording()</u></a> Request that stream recording stop.
void	<a href="#"><u>touch()</u></a> (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	<a href="#">onLiveStreamDvrRecorderCreate</a> ( <a href="#">ILiveStreamDvrRecorder</a> recorder, String streamName) Called when recorder created
void	<a href="#">onLiveStreamDvrRecorderDestroy</a> ( <a href="#">ILiveStreamDvrRecorder</a> recorder) Called when recorder destroyed
void	<a href="#">onLiveStreamDvrRecorderInit</a> ( <a href="#">ILiveStreamDvrRecorder</a> 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	<a href="#">shouldDvrRecord</a> (String recorderName, <a href="#">IMediaStream</a> 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

<a href="#">IApplicationInstance</a>	<a href="#">getApplicationInstance()</a> Get the application instance associated with this live stream packetizer.
int	<a href="#">getLiveStreamPacketizerId()</a> Get the live stream packetizer id
<a href="#">WMSProperties</a>	<a href="#">getProperties()</a> Get properties
long	<a href="#">getRepeaterLastSequence()</a> Get the sequence number of the last added repeater item
<a href="#">IMediaStream</a>	<a href="#">getStartStream()</a> Get the current stream that is being packetized
void	<a href="#">handlePacket()</a> ( <a href="#">IMediaStream</a> stream, <a href="#">AMFPacket</a> packet) Called to handle an incoming packet
void	<a href="#">init()</a> (String streamName, String packetizerName, <a href="#">IApplicationInstance</a> appInstance, LiveStreamPacketizerItem liveStreamPacketizerItem) Initialize live stream packetizer
boolean	<a href="#">isActive()</a> Is the live stream packetizer active
boolean	<a href="#">isPacketizeAudio()</a> True if audio is being packetized
boolean	<a href="#">isPacketizeData()</a> True if data is being packetized
boolean	<a href="#">isPacketizeVideo()</a> True if video is being packetized
boolean	<a href="#">isRepeaterEdge()</a> Is this packetizer a live repeater edge
void	<a href="#">resetStream()</a> ( <a href="#">IMediaStream</a> stream) Called when something happens that forces the stream to reset
void	<a href="#">setLiveStreamPacketizerId()</a> (int id) Set the live stream packetizer id

void	<a href="#"><u>setPacketizeAudio</u></a> (boolean packetizeAudio) Set to true to packetize audio
void	<a href="#"><u>setPacketizeData</u></a> (boolean packetizeVideo) Set to true to packetize data
void	<a href="#"><u>setPacketizeVideo</u></a> (boolean packetizeVideo) Set to true to packetize video
void	<a href="#"><u>setRepeaterEdge</u></a> (boolean isRepeaterEdge) Set is live repeater edge
void	<a href="#"><u>shutdown</u></a> ( ) Called to shutdown the live stream packetizer
void	<a href="#"><u>startStream</u></a> ( <a href="#"><u>IMediaStream</u></a> stream) Called when the stream starts
void	<a href="#"><u>touch</u></a> (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	<a href="#">onLiveStreamPacketizerCreate</a> ( <a href="#">ILiveStreamPacketizer</a> liveStreamPacketizer, String streamName) Called when packetizer created
void	<a href="#">onLiveStreamPacketizerDestroy</a> ( <a href="#">ILiveStreamPacketizer</a> liveStreamPacketizer) Called when packetizer destroyed
void	<a href="#">onLiveStreamPacketizerInit</a> ( <a href="#">ILiveStreamPacketizer</a> 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	<a href="#">isLiveStreamPacketize</a> (String packetizer, <a href="#">IMediaStream</a> 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	<a href="#"><u>close</u></a> ( <a href="#"><u>IMediaStream</u></a> stream) Called when live stream transcoder is stream is closed
<a href="#"><u>IApplicationInstance</u></a>	<a href="#"><u>getAppInstance</u></a> () Get the application instance associated with this live stream transcoder.
String	<a href="#"><u>getContextStr</u></a> () Get the streaming context for this live stream transcoder.
LicenseHolder	<a href="#"><u>getLicenseHolder</u></a> () Get license holder.
LiveStreamTranscoderItem	<a href="#"><u>getLiveStreamTranscoderItem</u></a> () Get the definition for live stream transcoder.
<a href="#"><u>WMSProperties</u></a>	<a href="#"><u>getProperties</u></a> () Get the user properties
String	<a href="#"><u>getStreamName</u></a> () Get the stream name of the source stream.
String	<a href="#"><u>getTranscoderName</u></a> () Get the live stream transcoder name
void	<a href="#"><u>handleOnMetadata</u></a> ( <a href="#"><u>IMediaStream</u></a> stream, <a href="#"><u>AMFPacket</u></a> packet, long timecode, boolean isSetDataFrame) Called for each new onMetaData packet
void	<a href="#"><u>handlePacket</u></a> ( <a href="#"><u>IMediaStream</u></a> stream, <a href="#"><u>AMFPacket</u></a> packet) Called for each new source packet
void	<a href="#"><u>init</u></a> (String streamName, <a href="#"><u>IMediaStream</u></a> stream, String transcoderName, <a href="#"><u>IApplicationInstance</u></a> appInstance, LiveStreamTranscoderItem liveStreamTranscoderItem) Called when live stream transcoder interface is initialized.
boolean	<a href="#"><u>isTemplateLoaded</u></a> () Is the transcoder template loaded.
boolean	<a href="#"><u>isTranscoderActive</u></a> (long currTime) Returns true if the transcoder is actively receiving packets
void	<a href="#"><u>resetStream</u></a> ( <a href="#"><u>IMediaStream</u></a> stream) Called when source stream changes.

void	<a href="#">setAppInstance</a> ( <a href="#">IApplicationInstance</a> appInstance) Set the application instance associated with this live stream transcoder.
void	<a href="#">setLiveStreamTranscoderItem</a> ( <a href="#">LiveStreamTranscoderItem</a> liveStreamTranscoderItem) Set the definition for live stream transcoder.
void	<a href="#">setStreamName</a> (String streamName) Set source stream name.
void	<a href="#">setTranscoderName</a> (String transcoderName) Get the live stream transcoder name
void	<a href="#">shutdown</a> ( <a href="#">IMediaStream</a> 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	<a href="#"><code>isLiveStreamTranscode</code></a> (String transcoder, <a href="#"><code>IMediaStream</code></a> 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	<a href="#"><code>onLiveStreamTranscoderCreate</code></a> ( <a href="#"><code>ILiveStreamTranscoder</code></a> liveStreamTranscoder, <a href="#"><code>IMediaStream</code></a> stream) Triggered when live stream transcoder is created.
void	<a href="#"><code>onLiveStreamTranscoderDestroy</code></a> ( <a href="#"><code>ILiveStreamTranscoder</code></a> liveStreamTranscoder, <a href="#"><code>IMediaStream</code></a> stream) Triggered when live stream transcoder is destroyed.
void	<a href="#"><code>onLiveStreamTranscoderInit</code></a> ( <a href="#"><code>ILiveStreamTranscoder</code></a> liveStreamTranscoder, <a href="#"><code>IMediaStream</code></a> 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	<a href="#">close()</a> Invoked on stream close
boolean	<a href="#">isSendOnMetadata()</a> Get to send onMetadata event when stream starts
boolean	<a href="#">play(Publisher publisher)</a> Invoked on play
boolean	<a href="#">seek(long timecode)</a> Invoked on seek
boolean	<a href="#">seek(long timecode, int seekType)</a> Invoked on seek
void	<a href="#">setDuration(long duration)</a> Set target duration for playback (milliseconds)
void	<a href="#">setRealTimeStartTime(long realTimeStartTime)</a> Set real start time (milliseconds)
void	<a href="#">setSendOnMetadata(boolean sendOnMetadata)</a> 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	<a href="#">onPlaylistItemStart</a> ( <a href="#">Stream</a> stream, <a href="#">PlaylistItem</a> playlistItem) Invoked when playlist item playback is started
void	<a href="#">onPlaylistItemStop</a> ( <a href="#">Stream</a> stream, <a href="#">PlaylistItem</a> 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	<a href="#">Playlist</a> (String sName) Class constructor - A simple structure that maintains a list of playlist items.
--------	--

### Method Summary

void	<a href="#">addItem</a> (String sName, int start, int length) Appends an item to this playlist
java.util.List	<a href="#">getItems</a> () Get the items in the playlist (returns a copy of the list)
String	<a href="#">getName</a> () Returns the name of this playlist as defined in the XML definition file
boolean	<a href="#">getRepeat</a> ()
boolean	<a href="#">open</a> ( <a href="#">Stream</a> s) Opens this playlist on the given stream...
void	<a href="#">removeItem</a> (int index) Remove an item from the list
void	<a href="#">setRepeat</a> (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	<a href="#">PlaylistItem</a> (String name, int start, int length, int index) Class constructor - A simple structure to define parameters associated with a playlist item
--------	---

### Method Summary

int	<a href="#">getIndex()</a>
int	<a href="#">getLength()</a> Number of seconds of track to play
String	<a href="#">getName()</a> Name of playlist stream
int	<a href="#">getStart()</a> Number of seconds into track to start from
void	<a href="#">setIndex</a> (int index)
String	<a href="#">toString()</a>

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

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

#### 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	<a href="#">sendOnMetadata</a>
-----------	--------------------------------

### Constructor Summary

public	<a href="#">PublishingProviderBase()</a>
--------	--

### Method Summary

boolean	<a href="#">isSendOnMetadata()</a> True to send onMetadata event on stream start
void	<a href="#">setSendOnMetadata</a> (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)
```

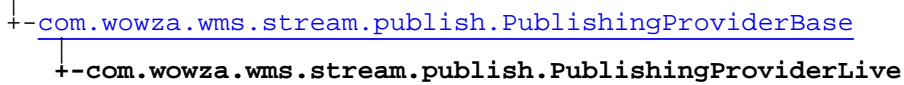
True to send onMetadata event on stream start

**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	<a href="#">PublishingProviderLive</a> ( <a href="#">Publisher</a> publisher, long timeoffset, String streamName)
--------	---

### Method Summary

void	<a href="#">close</a> ()
long	<a href="#">getDuration</a> ()
long	<a href="#">getStartOnPreviousBufferTime</a> ()
boolean	<a href="#">isStartOnPreviousKeyFrame</a> ()
boolean	<a href="#">play</a> ( <a href="#">Publisher</a> publisher)
boolean	<a href="#">seek</a> (long timecode)
boolean	<a href="#">seek</a> (long timecode, int seekType)
void	<a href="#">setDuration</a> (long duration)
void	<a href="#">setRealTimeStartTime</a> (long realTimeStartTime)
void	<a href="#">setStartOnPreviousBufferTime</a> (long startOnPreviousBufferTime)
void	<a href="#">setStartOnPreviousKeyFrame</a> (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()
```

---

---

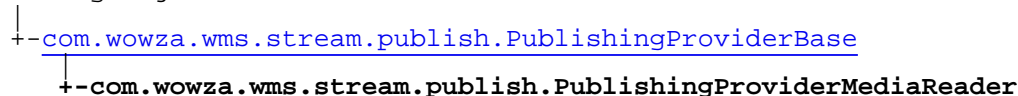
**setStartOnPreviousBufferTime**

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

---

## com.wowza.wms.stream.publish Class PublishingProviderMediaReader

java.lang.Object



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	<a href="#">PublishingProviderMediaReader</a> ( <a href="#">Publisher</a> publisher, long timeoffset, String streamName) Constructor
--------	---

### Method Summary

void	<a href="#">close</a> ()
long	<a href="#">getDuration</a> () Get the target playback duration (milliseconds)
boolean	<a href="#">play</a> ( <a href="#">Publisher</a> publisher)
boolean	<a href="#">seek</a> (long timecode)
boolean	<a href="#">seek</a> (long timecode, int seekType)
void	<a href="#">setDuration</a> (long duration)
void	<a href="#">setRealTimeStartTime</a> (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	<a href="#">Stream()</a>
--------	--------------------------

### Method Summary

void	<a href="#">addListener(IStreamActionNotify listener)</a> Add a listener
boolean	<a href="#">addToPlaylist(int index, String name, int start, int length)</a> Inserts a media source item to this playlist at a particular index, without interruption.
boolean	<a href="#">addToPlaylist(String existing, String name, int start, int length)</a> Insert item into playlist just after the first item in the playlist with a given name.
void	<a href="#">close()</a> Call this method when you have finished with the playlist object.
void	<a href="#">closeAndWait()</a> Call this method when you have finished with the playlist object.
static <a href="#">Stream</a>	<a href="#">createInstance(IApplicationInstance appInstance, String sName)</a> Use this to create a named Stream on an application instance.
static <a href="#">Stream</a>	<a href="#">createInstance(IVHost vhost, String applicationName, String sName)</a> Use this to create a named Stream on the default instance (_definst_) of an application on a particular VHost.
static <a href="#">Stream</a>	<a href="#">createInstance(IVHost vhost, String applicationName, String appInstanceName, String sName)</a> Use this to create a named Stream on the non-default instance of an application on a particular VHost.
<a href="#">PlaylistItem</a>	<a href="#">getCurrentItem()</a> Get the currently playing playlist item
Object	<a href="#">getLock()</a> Get the synchronization lock for this interface.

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

void	<a href="#"><u>run()</u></a> Overridden from class Runnable ....
void	<a href="#"><u>setPollingInterval</u></a> (int pollingInterval) Set the polling interval (milliseconds)
void	<a href="#"><u>setRepeat</u></a> (boolean repeat) Use this to make the playlist repeat or not...
void	<a href="#"><u>setSendOnMetadata</u></a> (boolean sendOnMetadata) True if sending onMetadata events
void	<a href="#"><u>setStartLiveOnPreviousBufferTime</u></a> (long startLiveOnPreviousBufferTime) Set time in milliseconds to go back in live stream buffer to get previous key frame
void	<a href="#"><u>setStartLiveOnPreviousKeyFrame</u></a> (boolean startLiveOnPreviousKeyFrame) Set to true to start live streams on most recent key frame (smoother switching)
void	<a href="#"><u>setSwitchLog</u></a> (boolean switchLog) Log when a playlist switch occurs
void	<a href="#"><u>setTimeOffsetBetweenItems</u></a> (int timeOffsetBetweenItems) Set time in milliseconds to add to stream time between playlist items (default is zero)
void	<a href="#"><u>setTimesInMilliseconds</u></a> (boolean timesInMilliseconds) If true start time and duration are 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	<a href="#">onCalculateSourceAudioBitrate</a> (LiveStreamTranscoder liveStreamTranscoder, long bitrate) Called when the bitrate of the source audio stream is calculated
void	<a href="#">onCalculateSourceVideoBitrate</a> (LiveStreamTranscoder liveStreamTranscoder, long bitrate) Called when the bitrate of the source video stream is calculated
void	<a href="#">onInitAfterLoadTemplate</a> (LiveStreamTranscoder liveStreamTranscoder) Called just after transcoder template is loaded.
void	<a href="#">onInitBeforeLoadTemplate</a> (LiveStreamTranscoder liveStreamTranscoder) Called just before transcoder template is loaded
void	<a href="#">onInitStart</a> (LiveStreamTranscoder liveStreamTranscoder, String streamName, String transcoderName, <a href="#">IApplicationInstance</a> appInstance, LiveStreamTranscoderItem liveStreamTranscoderItem) Call when live stream transcoder is first created
void	<a href="#">onInitStop</a> (LiveStreamTranscoder liveStreamTranscoder) At the end of the initialization process
void	<a href="#">onRegisterStreamNameGroup</a> (LiveStreamTranscoder liveStreamTranscoder, TranscoderStreamNameGroup streamNameGroup) Called after a stream name group is resolved and registered with MediaStreamMap
void	<a href="#">onResetStream</a> (LiveStreamTranscoder liveStreamTranscoder) Called when the stream feeding the live stream transcoder switches.
void	<a href="#">onSessionAudioDecodeCodecInfo</a> (LiveStreamTranscoder liveStreamTranscoder, com.wowza.wms.media.model.MediaCodecInfoAudio codecInfoAudio) Called when audio decoding information is available.
void	<a href="#">onSessionAudioEncodeCodecInfo</a> (LiveStreamTranscoder liveStreamTranscoder, TranscoderSessionAudioEncode sessionAudioEncode, com.wowza.wms.media.model.MediaCodecInfoAudio codecInfoAudio) Called when audio encoding information is available.
void	<a href="#">onSessionAudioEncodeCreate</a> (LiveStreamTranscoder liveStreamTranscoder, TranscoderSessionAudioEncode sessionAudioEncode) Called when audio encoder session is created.
void	<a href="#">onSessionAudioEncodeInit</a> (LiveStreamTranscoder liveStreamTranscoder, TranscoderSessionAudioEncode sessionAudioEncode) Called after audio session is initialized.

void	<a href="#"><u>onSessionAudioEncodeSetup</u></a> (LiveStreamTranscoder liveStreamTranscoder, TranscoderSessionAudioEncode sessionAudioEncode) Called after native audio encoder is created and initialized.
void	<a href="#"><u>onSessionDataEncodeCreate</u></a> (LiveStreamTranscoder liveStreamTranscoder, TranscoderSessionDataEncode sessionDataEncode) Called when data encoder session is created.
void	<a href="#"><u>onSessionDataEncodeInit</u></a> (LiveStreamTranscoder liveStreamTranscoder, TranscoderSessionDataEncode sessionDataEncode) Called after data session is initialized.
void	<a href="#"><u>onSessionDestinationCreate</u></a> (LiveStreamTranscoder liveStreamTranscoder, TranscoderSessionDestination sessionDestination) Called when transcoding destination is created
void	<a href="#"><u>onSessionVideoDecodeCodecInfo</u></a> (LiveStreamTranscoder liveStreamTranscoder, com.wowza.wms.media.model.MediaCodecInfoVideo codecInfoVideo) Called when video decoding information is available.
void	<a href="#"><u>onSessionVideoEncodeCodecInfo</u></a> (LiveStreamTranscoder liveStreamTranscoder, TranscoderSessionVideoEncode sessionVideoEncode, com.wowza.wms.media.model.MediaCodecInfoVideo codecInfoVideo) Called when video encoding information is available.
void	<a href="#"><u>onSessionVideoEncodeCreate</u></a> (LiveStreamTranscoder liveStreamTranscoder, TranscoderSessionVideoEncode sessionVideoEncode) Called when video encoder session is created.
void	<a href="#"><u>onSessionVideoEncodeInit</u></a> (LiveStreamTranscoder liveStreamTranscoder, TranscoderSessionVideoEncode sessionVideoEncode) Called after video session is initialized.
void	<a href="#"><u>onSessionVideoEncodeSetup</u></a> (LiveStreamTranscoder liveStreamTranscoder, TranscoderSessionVideoEncode sessionVideoEncode) Called after native video encoder is created and initialized.
void	<a href="#"><u>onShutdownStart</u></a> (LiveStreamTranscoder liveStreamTranscoder) Called when the live stream transcoder starts to shutdown.
void	<a href="#"><u>onShutdownStop</u></a> (LiveStreamTranscoder liveStreamTranscoder) Called when the live stream transcoder is shutdown.
void	<a href="#"><u>onUnregisterStreamNameGroup</u></a> (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	<a href="#">getAndClearPendingFrameGrabs()</a> Returns a list of all grab frame requests that are pending and clears the pending list.
void	<a href="#">grabFrame(ITranscoderFrameGrabResult grabResult)</a> Call to grab a frame.
void	<a href="#">grabFrame(ITranscoderFrameGrabResult grabResult, int width, int height)</a> 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	<a href="#">onGrabFrame</a> (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	<a href="#"><code>addOverlay</code></a> (int index, TranscoderVideoOverlayFrame overlay) Add an overlay to a video stream.
void	<a href="#"><code>clearOverlay</code></a> (int index) Clear video overlay
java.util.Map	<a href="#"><code>getAndClearPendingOverlays</code></a> ( ) Get a list of pending overlay requests and clear the pending overlay queue.
boolean	<a href="#"><code>isOverlayAvailable</code></a> ( ) 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	<a href="#"><code>onAfterDecodeFrame</code></a> (TranscoderSessionVideo sessionVideo, TranscoderStreamSourceVideo sourceVideo, long frameCount)
void	<a href="#"><code>onAfterScaleFrame</code></a> (TranscoderSessionVideo sessionVideo, TranscoderStreamSourceVideo sourceVideo, long frameCount)
void	<a href="#"><code>onBeforeDecodeFrame</code></a> (TranscoderSessionVideo sessionVideo, TranscoderStreamSourceVideo sourceVideo, long frameCount)
void	<a href="#"><code>onBeforeScaleFrame</code></a> (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	<a href="#">onAfterEncodeFrame</a> (TranscoderSessionVideoEncode sessionVideoEncode, TranscoderStreamDestinationVideo destinationVideo, long frameCount)
void	<a href="#">onBeforeEncodeFrame</a> (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	<a href="#"><code>getFrameCount()</code></a>
boolean	<a href="#"><code>isEncoderInUse()</code></a>
boolean	<a href="#"><code>isRunning()</code></a> Returns true if transcoder worker is running (internal use)
void	<a href="#"><code>setFrameCount(long frameCount)</code></a>

### 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	<a href="#"><code>addSorterPacket</code></a> (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	<a href="#">RTPUtils()</a>
--------	----------------------------

### Method Summary

static double[]	<a href="#">decodeRangeHeader</a> (String rangeStr) Decode RTP range header, Internal use.
static <a href="#">RTPStream</a>	<a href="#">decodeStreamInfo</a> ( <a href="#">RTPContext</a> context, String streamId, String streamInfo) Decode SDP info and create RTP stream
static byte[]	<a href="#">formatH264CodecConfig</a> (byte[] sps, java.util.List ppss, byte[] profileLevel) Format codec config info, Internal use.
static byte[]	<a href="#">formatH264CodecConfigPacket</a> (byte[] sps, java.util.List ppss, byte[] profileLevel) Format codec config info, Internal use.
static void	<a href="#">loadConfigFile</a> ( <a href="#">RTPContext</a> rtpContext, String fileURL) Load config file, Internal use.
static <a href="#">RTTPushPublishSession</a>	<a href="#">startRTTPull</a> ( <a href="#">IApplicationInstance</a> appInstance, String streamName, boolean streamPacketizer, String ipAddress, int streamPort) Start pushing an RTP stream
static <a href="#">RTTPushPublishSession</a>	<a href="#">startRTTPull</a> ( <a href="#">IApplicationInstance</a> appInstance, String streamName, boolean streamPacketizer, String ipAddress, int streamPort, boolean isRTPWrapped) Start pushing an RTP stream
static <a href="#">RTTPushPublishSession</a>	<a href="#">startRTTPull</a> ( <a href="#">IApplicationInstance</a> appInstance, String streamName, boolean streamPacketizer, String ipAddress, int audioPort, int videoPort) Start pushing an RTP stream
static <a href="#">RTTPushPublishSession</a>	<a href="#">startRTTPull</a> ( <a href="#">IApplicationInstance</a> appInstance, String streamName, boolean streamPacketizer, String ipAddress, int audioPort, int videoPort, boolean isRTPWrapped) Start pushing an RTP stream
static <a href="#">RTTPushPublishSession</a>	<a href="#">startRTTPull</a> ( <a href="#">IApplicationInstance</a> appInstance, String streamName, <a href="#">RTPDestination</a> rtpDestination) Start pushing an RTP stream

static void	<a href="#">stopRTPPull</a> ( <a href="#">RTPPushPublishSession</a> rtpPushPublishSession) Stop pushing an RTP stream
static String	<a href="#">updateSDPDestination</a> ( <a href="#">RTPDestination</a> rtpDestination, String sdpData) Update SDP data information with RTP destination information
static void	<a href="#">writeCodecConfig</a> (RTPTrack rtpTrack, int codecId, long adjTimecode, byte[] codecConfig)
static void	<a href="#">writeCodecConfig</a> (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	<a href="#">StreamUtils()</a>
--------	-------------------------------

### Method Summary

static int	<a href="#">directOutput</a> (byte[] dataBuffer, int[] headerValues, boolean isAbsTimecode, int src, java.io.OutputStream out, <a href="#">AMFObj</a> wmsObj, byte[] workBuffer, int chunkSize) Direct output, Internal use.
static double	<a href="#">getStreamBitrate</a> ( <a href="#">IApplicationInstance</a> appInstance, String streamName)
static double	<a href="#">getStreamBitrate</a> ( <a href="#">IMediaStream</a> stream) Get the approximate bitrate of a media file in bits/per-second.
static double	<a href="#">getStreamLength</a> ( <a href="#">IApplicationInstance</a> appInstance, String streamName) Get the duration of a media file in seconds.
static double	<a href="#">getStreamLength</a> ( <a href="#">IMediaStream</a> stream) Get the duration of a media file in seconds.
static void	<a href="#">loadConfigFile</a> ( <a href="#">StreamList</a> streamDefs, String fileURL) Load Streams.xml, Internal use.
static int	<a href="#">packetOutput</a> (java.io.OutputStream out, <a href="#">IMediaStream</a> stream, <a href="#">AMFPacket</a> packet, long timecode, <a href="#">AMFObj</a> wmsObj, byte[] workBuffer, int chunkSize) Packet output.
static int	<a href="#">packetOutput</a> (java.io.OutputStream out, <a href="#">IMediaStream</a> stream, <a href="#">AMFPacket</a> packet, long timecode, <a href="#">AMFObj</a> 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	<a href="#">HostPort()</a> Create an empty HostPort object
--------	---

### Method Summary

void	<a href="#">addHttpProvider(IHTTPProvider httpProvider)</a>
void	<a href="#">addHttpProvider(IHTTPProvider2 httpProvider)</a>
void	<a href="#">addHTTPStreamerAdapterID(String ID)</a>
void	<a href="#">configureSocketAcceptor(org.apache.mina.transport.socket.nio.SocketAcceptorConfig socketConfig)</a> Configure a socketAcceptor
java.net.InetAddress	<a href="#">getAddress()</a> Get the ipAddress as an InetAddress object
String	<a href="#">getAddressRawStr()</a>
String	<a href="#">getAddressStr()</a> Get a String representation of the address
HostPortConfig	<a href="#">getConfiguration()</a> Get the socket configuration
java.util.List	<a href="#">getHttpProviders()</a>
java.util.List	<a href="#">getHTTPStreamerAdapterIDs()</a>
int	<a href="#">getPort()</a> Get port
int	<a href="#">getProcessorCount()</a> Get the number of threads to use to service this incoming port

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

formmatted string

---

**toString**

```
public String toString()
```

Return object as formatted string

**Returns:**

formmatted 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	<a href="#">HostPortList()</a> Create empty HostPortList
--------	---

### Method Summary

void	<a href="#">add(<a href="#">HostPort</a> hostPort)</a> Add HostPort object
<a href="#">HostPort</a>	<a href="#">get(int index)</a> Get HostPort object at index, null if out of bounds
int	<a href="#">size()</a> 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	<a href="#"><code>onAcceptorCreate</code></a> ( <a href="#"><code>HostPort</code></a> hostPort, java.util.Map acceptorMap) Triggered when a new acceptor is created
void	<a href="#"><code>onAcceptorDestroy</code></a> ( <a href="#"><code>HostPort</code></a> 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	<a href="#">ACCEPTORS_ACCEPTOR</a> Acceptor types: acceptor Value: <b>acceptor</b>
public static final	<a href="#">ACCEPTORS_HANDLERADAPTER</a> Acceptor types: handler Value: <b>handlerAdapter</b>
public static final	<a href="#">CODEC_AUDIO_AAC</a> Value: <b>10</b>
public static final	<a href="#">CODEC_AUDIO_G711_ALAW</a> Value: <b>7</b>
public static final	<a href="#">CODEC_AUDIO_G711_MULAW</a> Value: <b>8</b>
public static final	<a href="#">CODEC_AUDIO_MP3</a> Value: <b>2</b>
public static final	<a href="#">CODEC_AUDIO_MP3_8</a> Value: <b>15</b>
public static final	<a href="#">CODEC_AUDIO_NELLYMOSER</a> Value: <b>6</b>
public static final	<a href="#">CODEC_AUDIO_NELLYMOSER_16MONO</a> Value: <b>4</b>
public static final	<a href="#">CODEC_AUDIO_NELLYMOSER_8MONO</a> Value: <b>5</b>
public static final	<a href="#">CODEC_AUDIO_PCM_BE</a> Value: <b>0</b>
public static final	<a href="#">CODEC_AUDIO_PCM_LE</a> Value: <b>3</b>

public static final	<a href="#"><u>CODEC_AUDIO_PCM_SWF</u></a> Value: <b>1</b>
public static final	<a href="#"><u>CODEC_AUDIO_RESERVED</u></a> Value: <b>9</b>
public static final	<a href="#"><u>CODEC_AUDIO_SPEEX</u></a> Value: <b>11</b>
public static final	<a href="#"><u>CODEC_AUDIO_UNKNOWN</u></a> Value: <b>-1</b>
public static final	<a href="#"><u>CODEC_AUDIO_VORBIS</u></a> Value: <b>9</b>
public static final	<a href="#"><u>CODEC_STREAM_MP2T</u></a> Value: <b>0</b>
public static final	<a href="#"><u>CODEC_STREAM_UNKNOWN</u></a> Value: <b>-1</b>
public static final	<a href="#"><u>CODEC_VIDEO_H263</u></a> Value: <b>9</b>
public static final	<a href="#"><u>CODEC_VIDEO_H264</u></a> Value: <b>7</b>
public static final	<a href="#"><u>CODEC_VIDEO_MPEG2</u></a> Value: <b>11</b>
public static final	<a href="#"><u>CODEC_VIDEO_MPEG4</u></a> Value: <b>10</b>
public static final	<a href="#"><u>CODEC_VIDEO_SCREEN</u></a> Value: <b>3</b>
public static final	<a href="#"><u>CODEC_VIDEO_SCREEN2</u></a> Value: <b>6</b>
public static final	<a href="#"><u>CODEC_VIDEO_SPARK</u></a> Value: <b>2</b>
public static final	<a href="#"><u>CODEC_VIDEO_UNKNOWN</u></a> Value: <b>-1</b>
public static final	<a href="#"><u>CODEC_VIDEO_VP6</u></a> Value: <b>4</b>

public static final	<a href="#">CODEC_VIDEO_VP6A</a> Value: <b>5</b>
public static final	<a href="#">CODEC_VIDEO_VP8</a> Value: <b>8</b>
public static final	<a href="#">CONTENTTYPE_ACKBANDWIDTH</a> AMF Content type: set acknowledge bandwidth size Value: <b>5</b>
public static final	<a href="#">CONTENTTYPE_AUDIO</a> AMF Content type: audio packet Value: <b>8</b>
public static final	<a href="#">CONTENTTYPE_BUFFERSIZE</a> AMF Content type: set buffer size Value: <b>4</b>
public static final	<a href="#">CONTENTTYPE_DATA</a> AMF Content type: data packet Value: <b>18</b>
public static final	<a href="#">CONTENTTYPE_DATA0</a> AMF Content type: data packet (AMF0) Value: <b>18</b>
public static final	<a href="#">CONTENTTYPE_DATA3</a> AMF Content type: data packet (AMF3) Value: <b>15</b>
public static final	<a href="#">CONTENTTYPE_FUNCTION</a> AMF Content type: function data (AMF0) Value: <b>20</b>
public static final	<a href="#">CONTENTTYPE_FUNCTION0</a> AMF Content type: function data (AMF0) Value: <b>20</b>
public static final	<a href="#">CONTENTTYPE_FUNCTION3</a> AMF Content type: function data (AMF3) Value: <b>17</b>
public static final	<a href="#">CONTENTTYPE_MEDIACHUNK</a> AMF Content type: media chunk Value: <b>22</b>
public static final	<a href="#">CONTENTTYPE_PLAYCALLBACK</a> AMF Content type: play callback Value: <b>127</b>
public static final	<a href="#">CONTENTTYPE_SETBANDWIDTH</a> AMF Content type: set bandwidth size Value: <b>6</b>

public static final	<a href="#">CONTENTTYPE_SETCHUNKSIZE</a> AMF Content type: set packet chunk size Value: <b>1</b>
public static final	<a href="#">CONTENTTYPE_SHAREDOBJECTS</a> AMF Content type: shared object packet (AMF0) Value: <b>19</b>
public static final	<a href="#">CONTENTTYPE_SHAREDOBJECTS0</a> AMF Content type: shared object packet (AMF0) Value: <b>19</b>
public static final	<a href="#">CONTENTTYPE_SHAREDOBJECTS3</a> AMF Content type: shared object packet (AMF3) Value: <b>16</b>
public static final	<a href="#">CONTENTTYPE_UNKNOWN</a> AMF Content type: unknown Value: <b>0</b>
public static final	<a href="#">CONTENTTYPE_VIDEO</a> AMF Content type: video packet Value: <b>9</b>
public static final	<a href="#">CONTENTTYPE_WATCHDOG</a> AMF Content type: watch dog Value: <b>3</b>
public static final	<a href="#">COUNTER_HTTPCUPERTINO</a> Value: <b>2</b>
public static final	<a href="#">COUNTER_HTTPDVRCHUNKS</a> Value: <b>7</b>
public static final	<a href="#">COUNTER_HTTPMPEGDASH</a> Value: <b>6</b>
public static final	<a href="#">COUNTER_HTTPSANJOSE</a> Value: <b>4</b>
public static final	<a href="#">COUNTER_HTTPSMOOTH</a> Value: <b>3</b>
public static final	<a href="#">COUNTER_HTTPWEBM</a> Value: <b>5</b>
public static final	<a href="#">COUNTER_RTMP</a> Value: <b>0</b>
public static final	<a href="#">COUNTER_RTP</a> Value: <b>1</b>

public static final	<a href="#"><u>COUNTER_TOTAL</u></a> Value: <b>8</b>
public static final	<a href="#"><u>FILEFORMAT_FLV</u></a> Value: <b>1</b>
public static final	<a href="#"><u>FILEFORMAT_MP4</u></a> Value: <b>2</b>
public static final	<a href="#"><u>FILEFORMAT_UNKNOWN</u></a> Value: <b>-1</b>
public static final	<a href="#"><u>LICENSECOUNTER_DRM_BUYDRM_LIVE</u></a> Value: <b>9</b>
public static final	<a href="#"><u>LICENSECOUNTER_DRM_BUYDRM_VOD</u></a> Value: <b>10</b>
public static final	<a href="#"><u>LICENSECOUNTER_DRM_EZDRM_LIVE</u></a> Value: <b>4</b>
public static final	<a href="#"><u>LICENSECOUNTER_DRM_EZDRM_VOD</u></a> Value: <b>5</b>
public static final	<a href="#"><u>LICENSECOUNTER_DRM_VERIMATRIX_LIVE</u></a> Value: <b>6</b>
public static final	<a href="#"><u>LICENSECOUNTER_DRM_VERIMATRIX_VOD</u></a> Value: <b>7</b>
public static final	<a href="#"><u>LICENSECOUNTER_NDVR</u></a> Value: <b>3</b>
public static final	<a href="#"><u>LICENSECOUNTER_PUBLISHER</u></a> Value: <b>0</b>
public static final	<a href="#"><u>LICENSECOUNTER_PUBLISHERTRANSCODER</u></a> Value: <b>8</b>
public static final	<a href="#"><u>LICENSECOUNTER_TOTAL</u></a> Value: <b>20</b>
public static final	<a href="#"><u>LICENSECOUNTER_TRANSCODE_DECODE</u></a> Value: <b>1</b>
public static final	<a href="#"><u>LICENSECOUNTER_TRANSCODE_DECODECOUNTAUDIO</u></a> Value: <b>14</b>

public static final	<a href="#">LICENSECOUNTER_TRANSCODE_DECODECOUNTAUDIOVIDEO</a> Value: <b>16</b>
public static final	<a href="#">LICENSECOUNTER_TRANSCODE_DECODECOUNTVIDEO</a> Value: <b>15</b>
public static final	<a href="#">LICENSECOUNTER_TRANSCODE_DECODEPOLLING</a> Value: <b>11</b>
public static final	<a href="#">LICENSECOUNTER_TRANSCODE_ENCODE</a> Value: <b>2</b>
public static final	<a href="#">LICENSECOUNTER_TRANSCODE_ENCODECOUNTAUDIO</a> Value: <b>17</b>
public static final	<a href="#">LICENSECOUNTER_TRANSCODE_ENCODECOUNTAUDIOVIDEO</a> Value: <b>19</b>
public static final	<a href="#">LICENSECOUNTER_TRANSCODE_ENCODECOUNTVIDEO</a> Value: <b>18</b>
public static final	<a href="#">LICENSECOUNTER_TRANSCODE_ENCODEPOLLING</a> Value: <b>12</b>
public static final	<a href="#">LICENSECOUNTER_TRANSCODE_STREAMNAMES</a> Value: <b>13</b>
public static final	<a href="#">VHOST_DEFAULT</a> Value: <b>_defaultVHost_</b>

## Method Summary

void	<a href="#">addAcceptorListener</a> ( <a href="#">IAcceptorNotify</a> acceptorListener) Add acceptor listener.
void	<a href="#">addApplicationListener</a> ( <a href="#">IApplicationNotify</a> applicationListener) Add application listener.
void	<a href="#">addIdleWorkerListener</a> ( <a href="#">IIdleWorkerNotify</a> idleWorkerListener) Add idleWorker listener.
void	<a href="#">addStartupStream</a> ( <a href="#">StartupStream</a> startupStream) Add a stream to the list of streams to start and virtual host startup
boolean	<a href="#">applicationExists</a> (String name) Return true if an application folder exists for this application name
void	<a href="#">closeHostPort</a> ( <a href="#">HostPort</a> hostPort, boolean isSuspend) Close an individual HostPort
boolean	<a href="#">createApplication</a> (String sName, String sStreamType, String sContentLoc) Method to create a new application



<a href="#"><u>IApplication</u></a>	<a href="#"><u>getApplication</u></a> (String applicationName) Get application by name.
java.util.List	<a href="#"><u>getApplicationFolderNames</u></a> () Get a list of application folder names
edu.emory.mathcs.backport.java.util.concurrent.locks.WMSReadWriteLock	<a href="#"><u>getApplicationLock</u></a> () Get the object used in synchronized statements to lock and application.
java.util.List	<a href="#"><u>getApplicationNames</u></a> () Get a list of application names
int	<a href="#"><u>getApplicationTimeout</u></a> () Get application time out (milliseconds).
AuthenticationList	<a href="#"><u>getAuthenticationList</u></a> () Get the list of available authentication methods
<a href="#"><u>IClient</u></a>	<a href="#"><u>getClient</u></a> (int clientId) Get client by client id.
<a href="#"><u>IClient</u></a>	<a href="#"><u>getClient</u></a> (int clientId, boolean create) Get client by client id and create if does not exist.
int	<a href="#"><u>getClientCount</u></a> () Get number of clients connected to this vHost.
int	<a href="#"><u>getClientIdleFrequency</u></a> () Get default client idle frequency (milliseconds)
int	<a href="#"><u>getClientTimeout</u></a> () Get client timeout.
<a href="#"><u>ConnectionCounter</u></a>	<a href="#"><u>getConnectionCounter</u></a> () Get vHost connection counter.
ConnectionCounterSimple	<a href="#"><u>getConnectionCounter</u></a> (int counterIndex) Get vHost connection counter for a specific technology (see IVHost.COUNTER_*)
int	<a href="#"><u>getConnectionLimit</u></a> () Get vHost connection limit.
int	<a href="#"><u>getCoreHandlerPoolSize</u></a> () Get the handler core thread pool size.
int	<a href="#"><u>getCoreTransportPoolSize</u></a> () Get the transport core thread pool size.
String	<a href="#"><u>getDateStarted</u></a> () Get date and time the server was started.
DvrRecorderList	<a href="#"><u>getDvrRecorderList</u></a> () Get the DvrRecorderList
DvrStoreList	<a href="#"><u>getDvrStoreList</u></a> () Get the list of DVR Stores

java.util.Properties	<a href="#"><u>getDynamicLogProperties()</u></a> Get the dynamic log properties defined at the vhost level in conf/log4j.properties
int	<a href="#"><u>getFileIOPoolSize()</u></a> Get the default file io pool size.
<a href="#"><u>ThreadPool</u></a>	<a href="#"><u>getHandlerThreadPool()</u></a> Get the VHost handler thread pool.
String	<a href="#"><u>getHomePath()</u></a> Get vHost configuration path.
<a href="#"><u>HostPortList</u></a>	<a href="#"><u>getHostPortsList()</u></a> Get list of host port definitions for vHost.
<a href="#"><u>IHTTPStreamerAdapter</u></a>	<a href="#"><u>getHTTPStreamerAdapter(String ID)</u></a> Get an HTTPStreamerAdapter by ID
java.util.List	<a href="#"><u>getHTTPStreamerAdapterIDs()</u></a> Get a list of HTTPStreamerAdapter IDs
HTTPStreamerContext	<a href="#"><u>getHTTPStreamerContext()</u></a> Get the HTTPStreamer (Cupertino Streaming and Silverlight Smooth Streaming) host context
HTTPStreamerList	<a href="#"><u>getHTTPStreamerList()</u></a> Get the list of HTTPStreamers
int	<a href="#"><u>getIdleCheckFrequency()</u></a> Get idle check frequency (milliseconds)
int	<a href="#"><u>getIdleMinimumWaitTime()</u></a> Get the minimum time (milliseconds) the idle worker thread will sleep before generating idle events
int	<a href="#"><u>getIdleWorkerCount()</u></a> Get number of threads used to generate idle events
IdleWorkersUtil	<a href="#"><u>getIdleWorkers()</u></a> Get the idle worker utility
<a href="#"><u>IOPerformanceCounter</u></a>	<a href="#"><u>getIoPerformanceCounter()</u></a> Get vHost IO performance counter.
<a href="#"><u>IOPerformanceCounter</u></a>	<a href="#"><u>getIoPerformanceCounter(int counterIndex)</u></a> Get vHost IO performance counter for a specific technology (see IVHost.COUNTER_*)
IOScheduler	<a href="#"><u>getIOScheduler()</u></a> Get IO scheduler for vHost.
int	<a href="#"><u>getKeepAliveTimeout()</u></a> Get the RTMPT connection keep alive timeout
LiveStreamPacketizerList	<a href="#"><u>getLiveStreamPacketizerList()</u></a> Get the LiveStreamPacketizerList
LiveStreamTranscoderList	<a href="#"><u>getLiveStreamTranscoderList()</u></a> Get the LiveStreamTranscoderList

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

com.wowza.wms.rtp.transport.RTPUDPDatagramConfig	<a href="#"><u>getRTPDatagramConfigIncoming()</u></a> Get the RTP Datagram Socket configuration
com.wowza.wms.rtp.transport.RTPUDPDatagramConfig	<a href="#"><u>getRTPDatagramConfigOutgoing()</u></a> Get the RTP Datagram Socket configuration
int	<a href="#"><u>getRTPIdleFrequency()</u></a> Get default RTP idle frequency (milliseconds)
java.util.List	<a href="#"><u>getStartupStreams()</u></a> Get the list of streams to start at virtual host startup
<a href="#"><u>StreamList</u></a>	<a href="#"><u>getStreamTypes()</u></a> Get default stream type.
<a href="#"><u>ThreadPool</u></a>	<a href="#"><u>getThreadPool()</u></a> Get the VHost handler thread pool.
String	<a href="#"><u>getTimeRunning()</u></a> Get the time vHost has been running.
double	<a href="#"><u>getTimeRunningSeconds()</u></a> Get time running in seconds
<a href="#"><u>ThreadPool</u></a>	<a href="#"><u>getTransportThreadPool()</u></a> Get the VHost transport thread pool.
com.wowza.wms.rtp.transport.UDPTransportManager	<a href="#"><u>getUDPTransportManager()</u></a> Get the UDP transport manager.
int	<a href="#"><u>getValidationFrequency()</u></a> Get time between validation pings (milliseconds)
void	<a href="#"><u>init()</u></a> (String basePath) Initialize vHost.
boolean	<a href="#"><u>isApplicationLoaded()</u></a> (String applicationName) Return true is the application is loaded
boolean	<a href="#"><u>isShuttingDown()</u></a> Is the VHost shutting down
boolean	<a href="#"><u>isStartStarupStreams()</u></a> Returns true if the startup streams are to start and vhost startup
boolean	<a href="#"><u>isSuspended()</u></a> Returns true is all HostPorts connected to this VHost are suspended
void	<a href="#"><u>killClient()</u></a> (int clientId) Remove client from vHost and send disconnect message.
void	<a href="#"><u>killRTSPSession()</u></a> (String rtspSessionId) Kill an RTSP connection by the RTSP session id
void	<a href="#"><u>putHTTPStreamerAdapter()</u></a> (String ID, <a href="#"><u>IHTTPStreamerAdapter</u></a> adapter) Add an HTTPStreamerAdapter

String	<a href="#"><u>readVHostConfig</u></a> (String sName) Method to read xml config file..
void	<a href="#"><u>removeAcceptorListener</u></a> ( <a href="#"><u>IAcceptorNotify</u></a> acceptorListener) Remove acceptor listener.
boolean	<a href="#"><u>removeApplication</u></a> (String sName) Method to remove an application
void	<a href="#"><u>removeApplicationListener</u></a> ( <a href="#"><u>IApplicationNotify</u></a> applicationListener) Remove applation listener.
void	<a href="#"><u>removeClient</u></a> (int clientId) Remove client from vHost.
void	<a href="#"><u>removeIdleWorkerListener</u></a> ( <a href="#"><u>IIIdleWorkerNotify</u></a> idleWorkerListener) Remove idleWorker listener
void	<a href="#"><u>reparentClient</u></a> ( <a href="#"><u>IClient</u></a> client) Move a client object to a new vhost.
void	<a href="#"><u>setAdminInterfaceHostPort</u></a> ( <a href="#"><u>HostPort</u></a> adminInterfaceHostPort) Set admin interface host port (not used)
void	<a href="#"><u>setApplicationTimeout</u></a> (int applicationTimeout) Set application time out (milliseconds).
void	<a href="#"><u>setClientIdleFrequency</u></a> (int clientIdleFrequency) Set default client idle frequency (milliseconds)
void	<a href="#"><u>setClientTimeout</u></a> (int clientTimeout) Set client timeout.
void	<a href="#"><u>setCoreHandlerPoolSize</u></a> (int corePoolSize) Set the handler core thread pool size.
void	<a href="#"><u>setCoreTransportPoolSize</u></a> (int corePoolSize) Set the transport core thread pool size.
void	<a href="#"><u>setDynamicLogProperties</u></a> (java.util.Properties dynamicLogProperties) Set the dynamic log properties set at the vhost level
void	<a href="#"><u>setFileIOPoolSize</u></a> (int fileIOPoolSize) Set default file io thread pool size.
void	<a href="#"><u>setIdleCheckFrequency</u></a> (int idleCheckFrequency) Set idle check frequency (milliseconds)
void	<a href="#"><u>setIdleMinimumWaitTime</u></a> (int idleMinimumWaitTime) Set the minimum time (milliseconds) the idle worker thread will sleep before generating idle events
void	<a href="#"><u>setIdleWorkerCount</u></a> (int idleWorkerCount) Set number of threads used to generate idle events
void	<a href="#"><u>setKeepAliveTimeout</u></a> (int keepAliveTimeout) Set the RTMPT connection keep alive timeout

void	<a href="#"><u>setMaximumPendingReadBytes</u></a> (int maximumPendingReaderBytes) Get maximum number of bytes a client connection can have waiting to be written before the connection is terminated.
void	<a href="#"><u>setMaximumPendingWriteBytes</u></a> (int maximumPendingWriteBytes) Set maximum number a bytes a client connection can have waiting to be sent before the connection is terminated.
void	<a href="#"><u>setMaximumSetBufferTime</u></a> (int maximumSetBufferTime) Set maximum number of milliseconds allowed for the NetStream.setBufferTime(secs) call.
void	<a href="#"><u>setNetConnectionIdleFrequency</u></a> (int netConnectionIdleFrequency) Set the server to server idle frequency (milliseconds)
void	<a href="#"><u>setNetConnectionProcessorCount</u></a> (int netConnectionProcessorCount) Set net connection processor count.
void	<a href="#"><u>setPingTimeout</u></a> (int pingTimeout) Set ping timeout (milliseconds)
void	<a href="#"><u>setRTPIdleFrequency</u></a> (int rtpIdleFrequency) Set default RTP idle frequency (milliseconds)
void	<a href="#"><u>setShuttingDown</u></a> (boolean shuttingDown) Set VHost shutting down flag
void	<a href="#"><u>setStartStarupStreams</u></a> (boolean startStarupStreams) Set to true to startup startup stream as vhost startup
void	<a href="#"><u>setValidationFrequency</u></a> (int validationFrequency) Set time between validation pings (milliseconds)
void	<a href="#"><u>shutdown</u></a> ( ) Shutdown.
void	<a href="#"><u>shutdownApplication</u></a> (String appName) Shutdown an application by name.
boolean	<a href="#"><u>startApplicationInstance</u></a> (String appName) Start an application instance.
boolean	<a href="#"><u>startApplicationInstance</u></a> (String appName, String appInstanceName) Start an application instance.
void	<a href="#"><u>startStartupStreams</u></a> ( ) Method to start startup streams
void	<a href="#"><u>stopStartupStreams</u></a> ( ) Method to stop startup streams
void	<a href="#"><u>suspendAllHostPorts</u></a> ( ) Suspend all HostPorts from accepting new connections.
boolean	<a href="#"><u>touchApplicationInstance</u></a> (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	<a href="#"><u>touchApplicationInstance</u></a> (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	<a href="#">unbindAllHostPorts()</a> Unbind all HostPorts and drop all connections
void	<a href="#">updateLoggingDuration()</a> Internal: update the internal logging values.
boolean	<a href="#">writeVHostConfig(String sName, String data)</a> 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)

---

**getRTPIdeFrequency**

```
public int getRTPIdeFrequency()
```

Get default RTP idle frequency (milliseconds)

**Returns:**

default RTP idle frequency (milliseconds)

---

**setRTPIdeFrequency**

```
public void setRTPIdeFrequency(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(IIIdleWorkerNotify idleWorkerListener)
```

Remove idleWorker listener

**Parameters:**

idleWorkerListener - idleWorker listener

---

## getNetConnectionHostPortConfig

```
public HostPortConfig getNetConnectionHostPortConfig()
```

Get the socket configuration for server to server connections

**Returns:**

socket configuration for server to server connections

---

## getMediaCasterSettings

```
public MediaCasterSettings getMediaCasterSettings()
```

Get the media caster settings

**Returns:**

media caster settings

---

## getMediaCasterList

```
public MediaCasterList getMediaCasterList()
```

Get the list of media caster definitions (MediaCaster.xml)

**Returns:**

list of media caster definitions

---

## getApplicationNames

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

Get a list of application names

**Returns:**

list of application names

---

## getApplicationFolderNames

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

Get a list of application folder names

**Returns:**

list of application folder names

## applicationExists

```
public boolean applicationExists(String name)
```

Return true if an application folder exists for this application name

**Parameters:**

name - application name

**Returns:**

true if an application folder exists for this application name

---

## getRTPContext

```
public RTPContext getRTPContext()
```

Get the RTP (real time protocol) virtual host context

**Returns:**

RTP (real time protocol) virtual host context

---

## getHTTPStreamerContext

```
public HTTPStreamerContext getHTTPStreamerContext()
```

Get the HTTPStreamer (Cupertino Streaming and Silverlight Smooth Streaming) host context

**Returns:**

HTTPStreamer (Cupertino Streaming and Silverlight Smooth Streaming) host context

---

## getRTPDatagramConfigIncoming

```
public com.wowza.wms.rtp.transport.RTPUDPDatagramConfig getRTPDatagramConfigIncoming()
```

Get the RTP Datagram Socket configuration

**Returns:**

RTP Datagram Socket configuration

---

## getRTPDatagramConfigOutgoing

```
public com.wowza.wms.rtp.transport.RTPUDPDatagramConfig getRTPDatagramConfigOutgoing()
```

Get the RTP Datagram Socket configuration

**Returns:**

RTP Datagram Socket configuration

---

## getAuthenticationList

```
public AuthenticationList getAuthenticationList()
```

Get the list of available authentication methods

**Returns:**

list of available authentication methods

---

## getIdleMinimumWaitTime

```
public int getIdleMinimumWaitTime()
```

Get the minimum time (milliseconds) the idle worker thread will sleep before generating idle events

**Returns:**

minimum time (milliseconds) the idle worker thread will sleep before generating idle events

---

## setIdleMinimumWaitTime

```
public void setIdleMinimumWaitTime(int idleMinimumWaitTime)
```

Set the minimum time (milliseconds) the idle worker thread will sleep before generating idle events

**Parameters:**

idleMinimumWaitTime - minimum time (milliseconds) the idle worker thread will sleep before generating idle events

---

## getIdleWorkers

```
public IdleWorkersUtil getIdleWorkers()
```

Get the idle worker utility

**Returns:**

idle worker utility

---

## isShuttingDown

```
public boolean isShuttingDown()
```

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	<a href="#">onVHostItemCreate</a> ( <a href="#">VHostItem</a> vhostItem) Triggered when vHostItem created
void	<a href="#">onVHostItemDestroy</a> ( <a href="#">VHostItem</a> vhostItem) Triggered when vHostItem destroyed
void	<a href="#">onVHostItemUpdate</a> ( <a href="#">VHostItem</a> 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	<a href="#">onVHostClientConnect</a> ( <a href="#">IVHost</a> vhost, <a href="#">IClient</a> inClient, <a href="#">com.wowza.wms.request.RequestFunction</a> function, <a href="#">AMFDataList</a> params) Triggered before a client connects to this virtual host.
void	<a href="#">onVHostCreate</a> ( <a href="#">IVHost</a> vhost) Triggered when vHost created
void	<a href="#">onVHostInit</a> ( <a href="#">IVHost</a> vhost) Triggered when vHost initialized
void	<a href="#">onVHostShutdownComplete</a> ( <a href="#">IVHost</a> vhost) Triggered at the end of vhost shutdown
void	<a href="#">onVHostShutdownStart</a> ( <a href="#">IVHost</a> 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	<a href="#">clear()</a>
void	<a href="#">reset()</a>

---

### 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	<a href="#">StreamItem</a> (String name, String baseClass, String playClass) Create a new streamItem
--------	---

### Method Summary

void	<a href="#">clearProperty</a> (String name) Clear property.
String	<a href="#">getBaseClass</a> () Get base class path.
String	<a href="#">getDescription</a> () Get streamType description.
String	<a href="#">getName</a> () Get streamType name.
String	<a href="#">getPlayClass</a> () Get play class path.
<a href="#">WMSProperties</a>	<a href="#">getProperties</a> () Get properties.
String	<a href="#">getProperty</a> (String name) Get property by name.
void	<a href="#">setBaseClass</a> (String baseClass) Set base class path.
void	<a href="#">setDescription</a> (String description) Set streamType description.
void	<a href="#">setName</a> (String name) Set streamType name.
void	<a href="#">setPlayClass</a> (String playClass) Set play class path.
void	<a href="#">setProperty</a> (String name, String value) Set property value.

String	<a href="#">toString()</a>
--------	----------------------------

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	<a href="#">StreamList()</a> Create empty StreamList
--------	---

### Method Summary

<a href="#">StreamItem</a>	<a href="#">getStreamDef(String name)</a> Get streamItem by streamType name.
java.util.Map	<a href="#">getStreamDefs()</a> Get Map of streamItems (by streamType names).
java.util.List	<a href="#">getStreamTypeNames()</a> 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	<a href="#">ThreadPool</a> ( <a href="#">IVHost</a> vhost, String name) Create a new thread pool attached to a vHost.
--------	--

### Method Summary

void	<a href="#">execute</a> (Runnable command) Execute a runnable object.
int	<a href="#">getActiveCount</a> () Get number of active threads.
java.util.concurrent. Executor	<a href="#">getExecutor</a> () Get the underlying Executor pool.
int	<a href="#">getQueueSize</a> () Get the number of command objects in the LinkedBlockingQueue.
void	<a href="#">init</a> (int corePoolSize) Initialize threadPool.
void	<a href="#">terminate</a> () 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	<a href="#"><u>VHostItem()</u></a> Create empty vHostItem
--------	--

### Method Summary

String	<a href="#"><u>getConfigDir()</u></a> Get configuration path for vHost.
int	<a href="#"><u>getConnectionLimit()</u></a> Get connection limit of this vHost item.
String	<a href="#"><u>getName()</u></a> Get vHost name
<a href="#"><u>WMSProperties</u></a>	<a href="#"><u>getProperties()</u></a> Get properties
boolean	<a href="#"><u>isVisited()</u></a> Has this vHostItem been visited during load of VHosts.xml file.
void	<a href="#"><u>reset()</u></a> Reset vHostItem to empty state
void	<a href="#"><u>setConfigDir</u></a> (String configDir) Set configuration path for vHost.
void	<a href="#"><u>setConnectionLimit</u></a> (int connectionLimit) Set connection limit of this vHost item.
void	<a href="#"><u>setName</u></a> (String name) Set vHost name
void	<a href="#"><u>setProperties</u></a> ( <a href="#"><u>WMSProperties</u></a> properties) Set properties
void	<a href="#"><u>setVisited</u></a> (boolean visited) Set is visited.
String	<a href="#"><u>toString()</u></a> 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	<a href="#"><u>VHostList()</u></a> Create empty vHostList
--------	--

### Method Summary

void	<a href="#"><u>addVHostItemListener()</u></a> ( <a href="#"><u>IVHostItemNotify</u></a> vHostItemListener) Add vHostItem listener.
java.util.List	<a href="#"><u>getVHostItems()</u></a> Get a list of vHostItems.
java.util.Map	<a href="#"><u>getVHostMap()</u></a> Get the Map of vHostItem definitions.
java.util.List	<a href="#"><u>getVHostNames()</u></a> Get a list of vHost names.
void	<a href="#"><u>loadConfig()</u></a>
void	<a href="#"><u>notifyVHostItemCreate()</u></a> ( <a href="#"><u>VHostItem</u></a> vhostItem) Notify vHostItem listener of item create.
void	<a href="#"><u>notifyVHostItemDestroy()</u></a> ( <a href="#"><u>VHostItem</u></a> vhostItem) Notify vHostItem listener of item destroy.
void	<a href="#"><u>notifyVHostItemUpdate()</u></a> ( <a href="#"><u>VHostItem</u></a> vhostItem) Notify vHostItem listener of item update.
void	<a href="#"><u>reloadConfig()</u></a> Reload VHosts.xml file.
void	<a href="#"><u>removeVHostItemListener()</u></a> ( <a href="#"><u>IVHostItemNotify</u></a> 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	<a href="#">VHostSingleton()</a>
--------	----------------------------------

### Method Summary

static void	<a href="#">addVHostListener</a> ( <a href="#">IVHostNotify</a> vhostListener) Add a vHost listener.
static <a href="#">IVHost</a>	<a href="#">getInstance</a> (String vhostName) Get vHost by name.
static <a href="#">IVHost</a>	<a href="#">getInstance</a> (String vhostName, boolean doCreate) Get vHost by name.
static java.util.List	<a href="#">getVHostNames</a> () Get a list of vHost names.
static void	<a href="#">init</a> (String vhostName, String configHome) Initialize vhost by name.
static void	<a href="#">notifyVHostClientConnect</a> ( <a href="#">IVHost</a> vhost, <a href="#">IClient</a> inClient, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params) Notify vhost client connect
static void	<a href="#">notifyVHostCreate</a> ( <a href="#">IVHost</a> vhost) Notify vHost listener of create.
static void	<a href="#">notifyVHostInit</a> ( <a href="#">IVHost</a> vhost) Notify vHost listener of ini.
static void	<a href="#">notifyVHostShutdownComplete</a> ( <a href="#">IVHost</a> vhost) Notify vHost listener of destruction.
static void	<a href="#">notifyVHostShutdownStart</a> ( <a href="#">IVHost</a> vhost) Notify vHost listener of destruction.
static void	<a href="#">remove</a> (String vhostName) Remove a vHost from list of running vHosts.
static void	<a href="#">removeApplicationListener</a> ( <a href="#">IVHostNotify</a> vhostListener) Remove vHost listener



static void	<a href="#">setServer</a> ( <a href="#">IServer</a> server) Set a reference to the current running server.
static void	<a href="#">shutdown</a> (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 603  
ABSOLUTE\_TIME\_PATTERN 603  
acceptConnection 320, 337  
ACCEPTORS\_ACCEPTOR 1312  
ACCEPTORS\_HANDLERADAPTER 1313  
acceptSession 566, 913  
acquire 749, 756, 757, 758, 1016, 1040  
acquireAndRelease 749, 757  
acquireMediaCaster 763  
acquireSocketAddress 881  
acquireUDPPortPair 881, 882  
actionListeners 911  
activateOptions 613  
add 77, 136, 137, 138, 171, 172, 173, 867, 1288  
addAcceptConnectionAttribute 351  
addAcceptorListener 1318  
addActionListener 922  
addApplicationInstanceListener 233  
addApplicationListener 1318  
addAudioCodecConfigPacket 1115  
addAudioData 1100, 1231, 1232  
addAudioDataInc 1231  
addBody 864  
addBouncyCastleSecurityProvider 98  
addBytes 864  
addChunk 204  
addClient 1016, 1040  
addClientListener 256, 1106, 1107  
addConnectionListener 319  
addContentDescriptor 702  
addData 215  
addDataData 1100, 1232  
addDataDataInc 1233  
addDataEx 218  
addDifference 77  
addDvrChunkListener 498  
addDvrRecorderListener 293  
addDvrRecordingListener 487  
addDvrStoreListener 487  
addDvrStreamManagerListener 294  
addFirstABSTRequest 588

addHttpProvider 1286  
addHTTPStreamerAdapterID 1286  
addHTTPStreamerSession 281  
addIdleWorkerListener 1324  
addIOPerformance 561, 919  
addIOPerformance2 561, 919  
addItem 1219  
addLicense 1182  
addListener 1250  
addLiveStreamPacketizerListener 289  
addLiveStreamTranscoderListener 290  
addManifestEntries 486, 497  
addMediaCasterListener 260  
addMediaReaderListener 286  
addMediaStreamListener 258, 1179  
addMediaWriterListener 284  
addMember 198  
addMetaDataListener 771  
addModuleListener 264  
addNameGroup 1181  
addObject 153  
addOverlay 1263  
addPlayStreamByName 262  
addPublisher 279  
addRange 390  
addRendition 790  
addRepeaterHeartBeatItem 489  
addRTPIncomingDatagramPortAll 282  
addRTPIncomingDatagramPortRange 281  
addRTPSession 274  
addRTSPStream 914  
addSegment 777  
addServerListener 985, 1002  
addSession 933  
addSessionListener 933  
addSharedObjectListener 258, 1028  
addSlotListener 1020, 1044  
addSorterPacket 1269  
addStartupStream 1331  
addStreamAttribute 952  
addStreamDomainStr 569  
addStreamDomainStrs 569  
addStreamInfo 952  
addStreamSrcToMediaCaster 759  
addString 152

addToCache 403  
addToChunk 465, 578  
addToManifest 452  
addToPlaylist 1247  
addTrack 949  
addTrackId 950  
addTrackInternal 949  
addTrailingZero 706  
addTrait 153  
addUserHTTPHeaders 573  
addVHostItemListener 1350  
addVHostListener 1354  
addVideoCodecConfigPacket 1115  
addVideoData 1099, 1230  
addVideoDataInc 1231  
addVideoH264SEILListener 1123  
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 956  
append 614  
appendParamsToUrl 103  
appInstance 852, 909, 1173  
appInstanceShutdown 909  
APPLICATION\_KEY 602  
applicationExists 1326  
ARCHIVE\_STRATEGY\_APPEND 420  
ARCHIVE\_STRATEGY\_DELETE 420  
ARCHIVE\_STRATEGY\_VERSION 420  
artifact 371, 393  
assembleQueryStr 70  
attachToWMSSession 955  
audioCodecId 782  
audioCodecStringToId 55  
audioCodecToMetaDataString 56  
audioCodecToString 56  
audioCodecTypeToString 89  
audioPacketizers 912  
AUDIOSAMPLE\_ACCESS\_ALL 333  
AUDIOSAMPLE\_ACCESS\_NONE 333  
AUDIOSAMPLEACCESS 1083  
authenticateHandler 510  
authenticateHTTPProvider 310  
authenticateHTTPProviderHandler 510  
authenticatePlayHandler 909  
authenticatePublishHandler 909  
authenticateRTSP 311  
authenticateSIP 312  
AuthenticateUsernamePasswordProviderBase 306  
authenticationMethod 510  
AUTHMETHOD\_PLAY 908  
AUTHMETHOD\_PUBLISH 908  
AUTHMETHOD\_UNKNOWN 908  
AVSYNCMETHODS\_RTPTIMECODE 946  
AVSYNCMETHODS\_SENDERREPORT 946  
AVSYNCMETHODS\_SYSTEMCLOCK 946  
AVSYNCMETHODS\_UNKNOWN 946  
avSyncNameToId 962  
  
**B**  
  
badStreams 853  
bind 894  
bitrateAudio 782  
bitrateTotal 782  
bitrateVideo 782  
booleanValue 165  
broadcastPlayMessage 1179  
broadcastGetObjectEncoding 1179  
broadcastMsg 261  
buffer 210  
BufferUtils 30

- byteArrayToFile 41
- byteArrayToInt 32, 33
- byteArrayToIntSafeSync 711
- byteArrayToLong 31, 32
- byteArrayToLongSafeSync 711
- byteArrayToShort 33, 34
- byteArrayToString 31
- byteStringLen 681
- byteValue 165
- C
  - cacheRTPStream 882
  - calcBitrate 467
  - calcTotalPacketSize 218
  - calculateChunkGroupTime 402
  - call 340
  - CALLBACK\_PARAM1 816
  - CALLBACK\_PARAM10 817
  - CALLBACK\_PARAM2 816
  - CALLBACK\_PARAM3 816
  - CALLBACK\_PARAM4 816
  - CALLBACK\_PARAM5 816
  - CALLBACK\_PARAM6 816
  - CALLBACK\_PARAM7 816
  - CALLBACK\_PARAM8 817
  - CALLBACK\_PARAM9 817
  - canHandle 511, 518, 542, 978
  - canPlay 494
  - canRecord 485, 494
  - canRecordAudio 465, 1188
  - canRecordData 465, 1189
  - canRecordVideo 465, 1189
  - CAT\_ALL 654
  - CAT\_application 642
  - CAT\_cupertino 643
  - CAT\_dvrchunk 643
  - CAT\_mpegdash 644
  - CAT\_rtsp 643
  - CAT\_sanjose 643
  - CAT\_server 642
  - CAT\_session 643
  - CAT\_smoothstreaming 643
  - CAT\_stream 643
  - CAT\_transcoder 644
  - CAT\_vhost 642
  - CAT\_webm 643
  - checkAndSetPlayLogged 560
  - checkSendMetadata 957
  - cleanUp 613
  - clear 77, 658, 868, 1021, 1039, 1113, 1336
  - clearByteContainer 207
  - clearFastPlaySettings 342, 1103
  - clearIntData 152
  - clearLoggingValues 533, 563, 587, 593, 922, 1112
  - clearOverlay 1263
  - clearProperty 737, 1339
  - clearRTSPSessionExtraLines 974
  - clearSegments 777, 790
  - clearStreamName 1176
  - clearStreamSrcToMediaCaster 759
  - clearTracks 949
  - client 306
  - clone 77, 198, 214, 447, 1053
  - cloneProperties 300
  - close 7, 27, 613, 1016, 1045, 1057, 1064, 1102, 1147, 1207, 1214, 1233, 1238, 1241, 1248
  - closeAndWait 1248
  - closeHostPort 1328
  - closeStream 836
  - CODEC\_AUDIO\_AAC 1309
  - CODEC\_AUDIO\_G711\_ALAW 1309
  - CODEC\_AUDIO\_G711\_MULAW 1309
  - CODEC\_AUDIO\_MP3 1308
  - CODEC\_AUDIO\_MP3\_8 1309
  - CODEC\_AUDIO\_NELLYMOSER 1309
  - CODEC\_AUDIO\_NELLYMOSER\_16MONO 1308
  - CODEC\_AUDIO\_NELLYMOSER\_8MONO 1308
  - CODEC\_AUDIO\_PCM\_BE 1308
  - CODEC\_AUDIO\_PCM\_LE 1308
  - CODEC\_AUDIO\_PCM\_SWF 1308
  - CODEC\_AUDIO\_RESERVED 1309
  - CODEC\_AUDIO\_SPEEX 1309
  - CODEC\_AUDIO\_UNKNOWN 1308
  - CODEC\_AUDIO\_VORBIS 1309
  - CODEC\_STREAM\_MP2T 1310
  - CODEC\_STREAM\_UNKNOWN 1310
  - CODEC\_TYPE 451
  - CODEC\_VIDEO\_H263 1307
  - CODEC\_VIDEO\_H264 1307

---

CODEC\_VIDEO\_MPEG2 1308  
CODEC\_VIDEO\_MPEG4 1307  
CODEC\_VIDEO\_SCREEN 1307  
CODEC\_VIDEO\_SCREEN2 1307  
CODEC\_VIDEO\_SPARK 1307  
CODEC\_VIDEO\_UNKNOWN 1306  
CODEC\_VIDEO\_VP6 1307  
CODEC\_VIDEO\_VP6A 1307  
CODEC\_VIDEO\_VP8 1307  
CODES\_HREF 603  
compress 148  
CONFIGURATOR\_CLASS\_KEY 602  
configureSocketAcceptor 1286  
ConnectionCounter 319  
connectionHolder 910  
containsDvrRecorder 278  
containsHeader 526  
containsHTTPStreamer 277  
containsIndex 532, 586, 592  
containsKey 191, 221  
containsLiveStreamPacketizer 278  
containsLiveStreamTranscoder 291  
containsProperty 1021, 1041  
containsSlot 1022, 1041  
containsStreamDomainStr 569  
containsStreamNameParts 567  
containsTime 506  
CONTENTTYPE\_ACKBANDWIDTH 1311  
CONTENTTYPE\_AUDIO 1311  
CONTENTTYPE\_BUFFERSIZE 1310  
CONTENTTYPE\_CHORD 699  
CONTENTTYPE\_DATA 1311  
CONTENTTYPE\_DATA0 1311  
CONTENTTYPE\_DATA3 1311  
CONTENTTYPE\_EVENTS 699  
CONTENTTYPE\_FUNCTION 1312  
CONTENTTYPE\_FUNCTION0 1312  
CONTENTTYPE\_FUNCTION3 1312  
CONTENTTYPE\_LYRICS 699  
CONTENTTYPE\_MEDIA 1061  
CONTENTTYPE\_MEDIACHUNK 1312  
CONTENTTYPE\_MEDIALIST 1061  
CONTENTTYPE\_MOVEMENT 699  
CONTENTTYPE\_OTHER 699  
CONTENTTYPE\_PLAYCALLBACK 1312  
CONTENTTYPE\_SETBANDWIDTH 1311  
CONTENTTYPE\_SETCHUNKSIZE 1310  
CONTENTTYPE\_SHAREDOBJECTS 1311  
CONTENTTYPE\_SHAREDOBJECTS0 1312  
CONTENTTYPE\_SHAREDOBJECTS3 1312  
CONTENTTYPE\_TRANSCRIPTION 699  
CONTENTTYPE\_TRIVIA 700  
CONTENTTYPE\_UNKNOWN 1310  
CONTENTTYPE\_URLIMAGES 700  
CONTENTTYPE\_URLWEBPAGES 700  
CONTENTTYPE\_VIDEO 1311  
CONTENTTYPE\_WATCHDOG 1310  
convertParams 11  
COOKIEDATEFORMAT 69  
cookieStr 909  
copyFile 47  
copyFile2 47  
COUNTER\_HTTPCUPERTINO 1306  
COUNTER\_HTTPDVRCHUNKS 1306  
COUNTER\_HTTPMPEGDASH 1306  
COUNTER\_HTTPSANJOSE 1306  
COUNTER\_HTTPSMOOTH 1306  
COUNTER\_HTTPWEBM 1306  
COUNTER\_RTMP 1305  
COUNTER\_RTP 1305  
COUNTER\_TOTAL 1306  
createApplication 1331  
createBroadcastMessage 863  
createConnectMessage 863  
createContextDeserialize 128  
createContextSerialize 128  
createDefaultMessage 863  
createDvrAudioChunk 462  
createDvrDataChunk 463  
createDvrOnMetadataChunk 463  
createDvrVideoChunk 462  
createEnhancedSeekMessage 864  
createHTTPSessionInstance 539  
createIndexLive 587  
createInstance 1228, 1229, 1245, 1246  
createPlayStatusMessage 863  
createSeekMessage 863  
createSOMessage 864  
createStream 834, 953, 1230  
CTRL\_playlist\_node 648

---

## 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 603

DATE\_AND\_TIME\_PATTERN 603

DATEFORMAT 162

dateValue 165

debug 622, 623

debugRTSPSession 912

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 1272

decodeS 1006

decodeSS 1006

decodeStorageDir 277, 999

decodeStreamInfo 1275

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 602

DEFAULT\_CONFIGURATION\_KEY 602

DEFAULT\_CUPERTINO\_PLAYLIST\_GZIP\_THRESHOLD  
443

DEFAULT\_PORT 613

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 427

DEFAULT\_PROPERTY\_CHUNK\_CACHE\_CLASS 422

DEFAULT\_PROPERTY\_CHUNK\_DURATION\_MINIMUM  
428

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  
434



- DEFAULT\_PROPERTY\_DEBUG\_MAX\_RAW\_PACKETS 436
- DEFAULT\_PROPERTY\_DEBUG\_MAX\_VALID\_CHUNKS\_LOGGED 434
- DEFAULT\_PROPERTY\_DEBUG\_RAW\_PACKETS 435
- DEFAULT\_PROPERTY\_DVR\_MAX\_CHUNK\_LOG 437
- DEFAULT\_PROPERTY\_FILE\_SYSTEM\_CLASS 421
- DEFAULT\_PROPERTY\_MANIFEST\_PERSISTENT\_CLASS 422
- DEFAULT\_PROPERTY\_MAX\_RECALC\_DURATION\_LOG 437
- 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 425
- 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 442
- DEFAULT\_PROPERTY\_SANJOSE\_ABST\_TIMESCALE 442
- DEFAULT\_PROPERTY\_SMOOTH\_MANIFEST\_LIVE\_CAN\_PAUSE 431
- DEFAULT\_PROPERTY\_SMOOTH\_MANIFEST\_LIVE\_CAN\_SEEK 431
- DEFAULT\_PROPERTY\_SMOOTH\_MANIFEST\_MAJOR\_VERSION 432
- DEFAULT\_PROPERTY\_SMOOTH\_MANIFEST\_MINOR\_VERSION 433
- DEFAULT\_PROPERTY\_SMOOTH\_MANIFEST\_RECORD\_ED\_CAN\_PAUSE 432
- DEFAULT\_PROPERTY\_SMOOTH\_MANIFEST\_RECORD\_ED\_CAN\_SEEK 431
- DEFAULT\_PROPERTY\_SMOOTH\_MANIFEST\_RECORD\_ED\_SPECIFY\_DURATION 432
- DEFAULT\_PROPERTY\_STORAGE\_DIRECTORY 419
- DEFAULT\_PROPERTY\_TEXT\_READER\_CLASS 424
- DEFAULT\_PROPERTY\_TEXT\_WRITER\_CLASS 423
- DEFAULT\_RANDOMACCESSREADER 1061
- DEFAULT\_REPOSITORY\_NAME 601
- DEFAULT\_XML\_CONFIGURATION\_FILE 602
- DefaultDvrStreamVersionHandler 368
- defaultLocale 97
- defaultTimeZone 97
- deleteArchivedStore 486
- deleteDirectory 47
- deleteSlot 1017, 1041
- deleteStream 835
- describe 965, 966
- deserialize 130, 142, 148, 165, 177, 183, 194, 195, 385, 388, 390, 461
- deserializeBody 665, 685, 688, 691, 693, 695, 701, 706, 709, 714
- deserializeDate 113
- deserializeFrame 682
- deserializeInnerObject 128
- deserializeInt 113
- deserializeString 114, 682
- detachFromWMSSession 956
- determineChunkGroupIdentifier 402
- determineExistingStoreForPlaying 369, 501
- determineExistingStoreForRecording 369, 500
- determineStreamVersion 467
- determineUtcTime 405
- directOutput 1278
- disconnect 321, 1019, 1028, 1044
- doBreak 861
- doCRC32 36
- doesFileExist 532
- doesStackContainMethod 42
- doHTTPAuthentication 512
- doIdle 766, 930
- DONT\_BREAK\_LINES 16
- doSessionRedirect 573
- doSet 78
- doubleValue 164
- doWatchdog 723, 748, 755, 879, 1000
- dummy 83

DVR\_DEFAULT\_FILESTORE 418  
DVR\_DEFAULT\_RECORDER\_ID 417  
DVR\_REPEATER\_PACKETIZER\_ID 417  
DVR\_STREAMING\_PACKETIZER\_ID 417  
DVR\_TIME 444  
DVR\_WINDOW\_DURATION\_UNLIMITED 419  
DvrManifestChunkEntry 371  
DvrManifestCodecEntry 375  
DvrManifestEntry 382  
DvrManifestEntryFactory 385  
DvrManifestEntryRangeGroup 389  
DvrManifestOnMetadataEntry 393  
DvrManifestTimeMapEntry 396  
dvrRecorders 1173  
dvrStart 381  
dvrStop 381  
dvrStreamStorageDeleted 478  
dvrStreamStorageLoaded 478  
dvrStreamStoreCreate 477  
dvrStreamStoreDestroy 477  
dvrStreamStoreInit 477  
dvrToPt 506  
dvrToUtc 506

## E

elapsedTime 911  
ElapsedTimer 44  
ENCODE 15  
encodeBytes 17, 18, 384  
encodeFileToFile 20  
encodeFromFile 20  
encodeHexString 30  
encodeObject 16  
encodeToFile 19  
encodeValue 103  
ENCRYPTION\_TYPE\_CUPERTINO 446  
ENCRYPTION\_TYPE\_PLAYREADY 446  
encryptions 371  
end 100  
endChunk 465, 577  
equals 95  
error 621  
EVT\_ALL 654  
EVT\_announce 647

EVT\_app\_start 646  
EVT\_app\_stop 646  
EVT\_comment 647  
EVT\_connect 644  
EVT\_connect\_burst 644  
EVT\_connect\_pending 644  
EVT\_create 645  
EVT\_decoderaudiostart 647  
EVT\_decoderaudiostop 647  
EVT\_decodervideostart 647  
EVT\_decodervideostop 647  
EVT\_describe 647  
EVT\_destroy 645  
EVT\_disconnect 644  
EVT\_encoderaudiostart 647  
EVT\_encoderaudiostop 647  
EVT\_encodervideostart 648  
EVT\_encodervideostop 648  
EVT\_pause 645  
EVT\_play 644  
EVT\_publish 644  
EVT\_record 646  
EVT\_recordstop 646  
EVT\_seek 645  
EVT\_server\_start 646  
EVT\_server\_stop 646  
EVT\_setbuffertime 645  
EVT\_setstreamtype 645  
EVT\_stop 645  
EVT\_unpause 645  
EVT\_unpublish 644  
EVT\_vhost\_start 646  
EVT\_vhost\_stop 646  
execute 1344  
exists 8, 1027  
existsRTSPTunnelingSession 883  
expandEndTime 401, 460  
expandEnvironmentVariables 97, 98  
expandStartTime 401, 460  
expandToPortPair 882  
extractCodecConfigFromTrackInfo 957  
extractHTTPRequestInfo 571

## F

fastDateFormat 162  
FastPlaySettings 1049  
fatal 622  
FCPublish 840  
FCSubscribe 841  
fcSubscribe 357  
FCUnPublish 840  
FCUnpublish 841  
FCUnSubscribe 842  
FCUnsubscribe 842  
fcUnSubscribe 357  
fcUnSubscribeAll 357  
FD\_ALL 654  
FD\_c\_client\_id 639  
FD\_c\_ip 639  
FD\_c\_proto 639  
FD\_c\_referrer 639  
FD\_c\_user\_agent 639  
FD\_cs\_bytes 639  
FD\_cs\_bytes\_inc 609  
FD\_cs\_stream\_bytes 640  
FD\_cs\_stream\_bytes\_inc 609  
FD\_cs\_uri\_query 641  
FD\_cs\_uri\_stem 641  
FD\_date 638  
FD\_s\_ip 640  
FD\_s\_port 642  
FD\_s\_uri 639  
FD\_sc\_bytes 640  
FD\_sc\_bytes\_inc 609  
FD\_sc\_stream\_bytes 640  
FD\_sc\_stream\_bytes\_inc 609  
FD\_time 638  
FD\_tz 638  
FD\_x\_app 638  
FD\_x\_appinst 639  
FD\_x\_category 638  
FD\_x\_comment 642  
FD\_x\_ctx 638  
FD\_x\_ctx\_override 638  
FD\_x\_duration 641  
FD\_x\_duration\_inc 609  
FD\_x\_event 637  
FD\_x\_file\_ext 641  
FD\_x\_file\_length 640  
FD\_x\_file\_name 641  
FD\_x\_file\_size 640  
FD\_x\_severity 642  
FD\_x\_sname 640  
FD\_x\_sname\_query 641  
FD\_x\_spos 640  
FD\_x\_status 641  
FD\_x\_stream\_id 642  
FD\_x\_suri 642  
FD\_x\_suri\_query 641  
FD\_x\_suri\_stem 641  
FD\_x\_vhost 638  
FILEEXTENSION 1013  
FILEFORMAT\_FLV 1310  
FILEFORMAT\_MP4 1310  
FILEFORMAT\_UNKNOWN 1310  
fileToByteArray 47  
FileUtils 46  
filters 510  
FLAGS\_DEFAULT 681  
floatValue 164  
flush 1017, 1028, 1045, 1116, 1233  
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 626  
forceReset 723  
formatBytes 40  
formatBytesShort 39  
formatBytesStruct 41  
formatDeleteCookie 71

formatH264CodecConfig 1275  
formatH264CodecConfigPacket 1275  
formatMilliseconds 42  
formatRTPInfo 967  
formatSeconds 42  
formatSetCookie 72  
formatUtcTime 42  
FORWARD 6  
frameTypeToString 56  
fS 1007

## G

generateAuth 85, 86, 87  
generateAuthHTTP 86  
generateHash 88  
generateHashBytes 87, 88  
get 139, 175, 192, 223, 224, 1026, 1289  
getAbsTimecode 206, 216  
getAccess 1046, 1114  
getActiveCount 1344  
getAdapterName 543  
getAddress 1284  
getAddressCount 897  
getAddressRawStr 1284  
getAddressStr 1284  
getAdminAgent 998  
getAdminGUID 1004  
getAdminInterfaceObjectList 985, 1002  
getAllAsStrings 302  
getAllowDomains 267  
getAndClearPendingFrameGrabs 1261  
getAndClearPendingOverlays 1264  
getAppInstance 232, 334, 493, 546, 562, 721, 823, 917, 954, 1178, 1187, 1208, 1230  
getAppInstanceName 958, 1178  
getAppInstanceNames 234  
getAppInstanceProperty 857  
getApplication 252, 334, 614, 823, 1315  
getApplicationFolderNames 1325  
getApplicationInstance 755, 1200  
getApplicationInstanceTouchTimeout 285  
getApplicationLock 1322  
getApplicationNames 1325  
getApplicationPath 232

getApplicationProperty 856  
getApplicationTimeout 264, 1320  
getAppName 957, 1178  
getArchiveStrategy 369, 484, 501  
getArtifactsTextRepresentation 373, 394  
getAudioCodec 57, 65, 375  
getAudioCodecConfigPacket 1115  
getAudioCodecId 785  
getAudioHost 888  
getAudioMissing 1099  
getAudioMP3Layer 64  
getAudioPacketizerItem 879, 927  
getAudioPort 887  
getAudioSize 1084  
getAudioTC 1087  
getAudioTrack 951  
getAuthenticatePlayHandler 916  
getAuthenticatePublishHandler 916  
getAuthenticationList 1326  
getAuthenticationMethod 512, 518  
getAutoAllocateInterleavePorts 969  
getAVSyncMethod 961  
getBaseClass 738, 1338  
getBasePath 8  
getBitrateAudio 784  
getBitrateTotal 784  
getBitrateVideo 784  
getBodySize 665, 686, 689, 691, 693, 695, 701, 707, 709, 714  
getBoolean 141, 177, 193, 194, 226, 228  
getBufferSize 696  
getBufferTime 344, 1084  
getBurstStartStop 1111  
getByte 140, 177, 193, 194, 226, 227  
getByteAllocation 73  
getByteContainerLevel 207  
getCacheName 1109  
getCallbackParamCount 819  
getChecksum 1053  
getChunkCounter 209  
getChunker 494  
getChunkGroupDuration 402  
getChunkIndex 396  
getChunks 204  
getClassName 196, 199  
getClient 255, 307, 314, 1090, 1314

getClientById 255  
getClientCount 255, 1317  
getClientCountTotal 254  
getClientID 837  
getClientId 333, 1090  
getClientIdGenerator 987, 1003  
getClientIdGeneratorRecycleDelaySize 1006  
getClientIdGeneratorRecycleSize 1006  
getClientIdGeneratorTimeout 1006  
getClientIdleFrequency 267, 1323  
getClientProperty 857  
getClients 255, 1015, 1040  
getClientsLockObj 275  
getClientTimeout 1313  
getClientUpdates 1043  
getClosestStartTime 400, 459, 499  
getCodecEntryForTime 454  
getCodecId 785  
getCommandInterface 1002  
getCommandInterfaceCommandHandler 989, 1006  
getCommandInterfaceHostPort 984, 1001  
getCommittedVirtualMemory 1005  
getCommonInitialTextRepString 384  
getConfigDir 1346  
getConfigPath 232  
getConfiguration 1285  
getConnectionCounter 233, 260, 984, 1000, 1001, 1319  
getConnectionHolder 561, 919  
getConnectionLimit 1320, 1347  
getConnectionTimeout 738  
getConnectionValidator 1001  
getConnectLastAttempt 725  
getConnectLastForceReset 725  
getConnectLastSuccess 725  
getConnectTime 342  
getContentDescriptors 702  
getContentLength 521  
getContentType 522, 701  
getContextStr 284, 481, 493, 1120, 1209  
getCookieStr 566, 926  
getCoreHandlerPoolSize 986, 1003, 1316  
getCoreTransportPoolSize 985, 1002, 1315  
getCount 1177  
getCpuTime 98  
getCryptoPoolActiveCount 1005  
getCryptoPoolMaxSize 1004  
getCurrent 321  
getCurrentHeapSize 1005  
getCurrentItem 1247  
getCurrentTime 469  
getData 218, 393, 689, 1031  
getDataBuffer 217  
getDataMissing 1099  
getDataSize 1085  
getDataTC 1088  
getDataType 1089  
getDate 44, 141, 177, 193, 194, 226, 228  
getDateStarted 234, 260, 343, 984, 1001, 1319  
getDateString 45  
getDefaultStreamingStore 483  
getDePacketizerList 882  
getDePacketizerName 883  
getDescription 666, 686, 689, 738, 1340  
getDirection 1050  
getDirecton 7  
getDouble 141, 176, 193, 194, 225, 227  
getDuration 383, 954, 1066, 1162, 1238, 1241  
getDvrApplicationContext 292  
getDvrArtifact 372, 393  
getDvrChunkAtTime 498  
getDvrChunkByIndex 498  
getDvrChunkNearTime 499  
getDvrFileSystemClass 483  
getDvrManager 493, 1188  
getDvrPlaylistRequest 467  
getDvrProperties 292  
getDvrRecorder 1117, 1121, 1181  
getDvrRecorderId 1188  
getDvrRecorderList 293, 1117, 1332  
getDvrRecorders 1182  
getDvrRepeater 1121  
getDvrSessionInfo 571  
getDvrStorageDir 483  
getDvrStorageWindowSeconds 483, 492  
getDvrStoreList 1332  
getDvrTime 458  
getDynamicLogProperties 987, 1003, 1327  
getElapsedTime 356, 570, 928, 1118  
getEmbeddedFlag 696  
getEncoding 614

getEncryptionDelegate 484  
getEncryptionRepeaterSharedSecret 484  
getEncryptions 372  
getEncryptionsTextRepresentation 373  
getEncryptionType 447  
getEndIndex 387  
getEntriesToPurge 454  
getExecutor 1344  
getExt 1112  
getExtent 1052  
getFastPlaySettings 341, 1103  
getFileExtension 1229  
getFileInBytes 81  
getFileInBytesRate 83  
getFileInfo 568  
getFileIOPoolSize 1316  
getFileOutBytes 82  
getFileOutBytesRate 83  
getFilePointer 7  
getFileSystem 494  
getFirstByte 218  
getFirstEntry 401, 456  
getFirstIndex 401  
getFirstRendition 790  
getFirstSegment 777  
getFlags 683, 714  
getFlashVer 334  
getFloat 141, 176, 193, 194, 225, 228  
getFps 1049  
getFrameCount 1268  
getFrameIdentifier 689  
getFrameMapIds 659  
getFrames 659  
getFrameType 57  
getGlobalLogValue 628  
getGUID 1004  
getHandler 894  
getHandlerThreadPool 986, 1003, 1317  
getHeader 520, 527  
getHeaderBytes 524  
getHeaderMap 520  
getHeaderNames 521  
getHeaders 527  
getHeaderSize 528, 1114  
getHeight 786  
getHighestVersionedStore 482  
getHomePath 1315  
getHost 887, 952  
getHostPortsList 1316  
getHostType 889  
getHTTPHeader 571  
getHTTPHeaderMap 571  
getHTTPHeaderNames 572  
getHTTPIntHeader 571  
getHTTPOriginMode 548  
getHttpProviders 1286  
getHTTPStreamerAdapter 561, 1330  
getHTTPStreamerAdapterIDs 1286, 1330  
getHTTPStreamerApplicationContext 281  
getHTTPStreamerContext 1326  
getHTTPStreamerItem 542  
getHTTPStreamerList 277, 1330  
getHTTPStreamerProperties 276  
getHTTPStreamerSession 1058, 1118  
getHTTPStreamerSessionCount 280, 281  
getHTTPStreamerSessionCountsByName 279  
getHTTPStreamerSessions 279, 280  
getID 544  
getId 204  
getIdleCheckFrequency 1323  
getIdleFrequency 347, 543, 924  
getIdleHandler 920  
getIdleMinimumWaitTime 1327  
getIdleTimeout 722  
getIdleWorkerCount 1324  
getIdleWorkers 1327  
getIdStr 682, 713  
getIdString 739  
getIndex 383, 533, 587, 592, 1222  
getIndexLive 587  
getIndexMap 399  
getInIpAddress 893  
getInitialRepeaterItems 489  
getInnerObj 199  
getInPort 894  
getInputStream 523  
getInstance 627, 998, 1353  
getInt 140, 176, 193, 194, 224, 227  
getIntData 151  
getIntHeader 521, 528

getIOPerformanceCounter 262, 568, 926  
getIoPerformanceCounter 234, 983, 1000, 1319  
getIOScheduler 1318  
getIoSession 929  
getIp 341, 917  
getIpAddress 563  
getItems 1218  
getJmxRemoteConfig 1002  
getKeepAliveTime 739  
getKeepAliveTimeout 1324  
getKey 192, 223  
getKeys 192, 223  
getLanguage 701  
getLastAccessed 1053  
getLastAudioTimecode 1233  
getLastAuthenticateMethod 916  
getLastClientId 1033  
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 1234  
getLastDisconnectStamp 324  
getLastDisconnectStampString 324  
getLastDisconnectTime 324  
getLastDisconnectTimeString 324  
getLastKeyFrame 1100  
getLastLiveEntry 401, 456  
getLastOperation 1033  
getLastPacket 1101  
getLastPacketsByType 965  
getLastPurgeTime 469  
getLastRecordedEntry 401, 453  
getLastRecordedIndex 398, 458  
getLastRequest 575  
getLastStreamId 835  
getLastTC 64  
getLastTouchTime 285  
getLastValidateTime 352  
getLastVideoTimecode 1233  
getLength 1066, 1222  
getLicenseCounter 1006  
getLicenseHolder 1191, 1208  
getLicenseTracker 1006  
getLicenseValidator 1001  
getLiveDuration 400, 458  
getLiveEntries 400, 456  
getLiveEntriesWithLimit 400, 457  
getLivePlaylist 593  
getLiveRangeEndingBeforeTime 401  
getLiveRepeaterCapabilities 356  
getLiveStreamDvrRecorderControl 292  
getLiveStreamingPacketizer 562  
getLiveStreamPacketizer 749, 838, 1117, 1118, 1180  
getLiveStreamPacketizerControl 282  
getLiveStreamPacketizerId 1198  
getLiveStreamPacketizerList 277, 355, 1116, 1330  
getLiveStreamPacketizerLock 1180  
getLiveStreamPacketizerProperties 275  
getLiveStreamRepeater 750, 1119  
getLiveStreamTranscoder 1122  
getLiveStreamTranscoderControl 292  
getLiveStreamTranscoderItem 1208  
getLiveStreamTranscoderList 291, 356, 1117, 1330  
getLiveStreamTranscoders 1122  
getLiveTailEntries 400  
getLiveThreads 1005  
getLocale 524  
getLock 559, 658, 746, 755, 778, 783, 791, 881, 1246  
getLockCount 748, 762  
getLogger 618, 628, 825  
getLoggerObj 626, 628  
getLong 140, 176, 193, 194, 225, 227  
getMACAddress 98  
getManifest 493  
getManifestChannel 459  
getManifestRepresentation 372, 375, 383, 394, 396  
getMapIdStr 665, 683, 713  
getMaxHeapSize 1005  
getMaximumPendingReadBytes 266, 1322  
getMaximumPendingWriteBytes 265, 355, 1321  
getMaximumSetBufferTime 266, 355, 1322

getMaxRTCPWaitTime 961  
getMaxStorageDirDepth 276  
getMaxTimecode 1100, 1233  
getMediaCaster 747, 756, 1133  
getMediaCasterCount 755  
getMediaCasterDef 721, 742  
getMediaCasterDefs 741  
getMediaCasterHostPortConfig 743  
getMediaCasterId 721, 747  
getMediaCasterList 1325  
getMediaCasterNames 741, 755  
getMediaCasterProcessorCount 744  
getMediaCasterProperties 275  
getMediacasterRTPRTSPRTPTTransportMode 295  
getMediaCasterSettings 1325  
getMediaCasterStreamItem 723  
getMediaCasterStreams 263  
getMediaCasterType 726, 953  
getMediaCasterValidator 285  
getMediaExtension 8, 1058, 1067  
getMediaIOPerformance 1104  
getMediaIOPerformanceCounter 334  
getMediaList 791, 1058  
getMediaListProvider 295  
getMediaListSegment 783  
getMediaName 8  
getMediaReader 969  
getMediaReaderContentTypes 295, 1332  
getMediaReaderProperties 275  
getMediaReaders 1320  
getMediaWriterProperties 275  
getMediaWriters 1320  
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 1064  
getMetadataEntryForTime 454  
getMetadataPacket 871, 956  
getMetaDataProvider 1114  
getMethod 521  
getMimeType 666  
getMinimumAvailableChunks 482  
getMissing 214  
getMode 966  
getModFunctions 263  
getModuleInstance 264  
getModuleList 264  
getMp3TagMap 1320  
getMPEGTSAudioBitrate 976  
getMPEGTSAudioLanguage 962  
getMPEGTSAudioPID 963  
getMPEGTSProgramID 962  
getMPEGTSVideoBitrate 976  
getMPEGTSVideoPID 962  
getMsgBytes 525  
getMultiplier 1049  
getName 232, 253, 737, 778, 785, 890, 1019, 1032, 1044, 1086,  
1220, 1222, 1248, 1317, 1338, 1346  
getNameGroupByGroupName 1181  
getNameGroups 1181  
getNameGroupStreamNames 1181  
getNetConnection 728, 1090  
getNetConnectionHostPortConfig 1325  
getNetConnectionIdleFrequency 1323  
getNetConnectionProcessorCount 1317  
getNextChunkIndex 458  
getNextCodecIndex 459  
getNextMetadataIndex 459  
getNextNameGroupId 1181  
getNextNetConnectionId 1313  
getNextSessionId 932  
getNextStreamIndex 1176, 1177  
getNextTimeMapIndex 459  
getNodeByTagName 106  
getNodeValue 107  
getNodeValueByTagName 106  
getNormalizedNTPTimecode 966  
getNumberLiveEntries 400, 455



getNumberRecordedEntries 400, 455  
getObject 141, 153, 177, 193, 194, 226, 228  
getObjectEncoding 152, 156, 208, 354  
getObjectNames 1026  
getObjectReference 157  
getOffsetToNextTag 696  
getOnMetadataData 67  
getOrCreate 1026  
getOutHost 964  
getOutIpAddress 894  
getOutPort 894  
getOutputStream 526  
getPacketizerName 482  
getPacketStartTime 382  
getPageUrl 346, 841  
getParam 819  
getParamBoolean 822, 823  
getParamCount 819  
getParamDate 821  
getParamDouble 821, 822  
getParameter 522  
getParameterMap 523  
getParameterNames 522  
getParameterValues 523  
getParamInt 821  
getParamLong 822  
getParamMixedArray 820  
getParamObj 820  
getParamString 820  
getParamType 819  
getParamValue 104  
getParent 1022, 1046, 1148  
getPassword 313  
getPath 9, 511, 524, 1057, 1066  
getPeakThreads 1005  
getPictureData 666  
getPictureType 666  
getPingRoundTripTime 343  
getPingTimeout 265, 352, 1321  
getPlayClass 1339  
getPlayDuration 572, 786  
getPlayer 1099  
getPlayerCount 748, 762  
getPlaylist 1246  
getPlaylistAlignment 467  
getPlayPackets 1100  
getPlayStart 572, 786  
getPlayStreamCount 263  
getPlayStreamCountsByName 263  
getPlayStreams 335  
getPlayStreamsByName 263  
getPollingInterval 1249  
getPort 615, 1284  
getProcessorCount 1285  
getProperties 233, 254, 336, 492, 542, 546, 570, 738, 778, 784, 791, 918, 987, 1003, 1100, 1188, 1197, 1208, 1315, 1339, 1347  
getProperty 301, 737, 1018, 1041, 1313, 1339  
getPropertyBoolean 301  
getPropertyDouble 302  
getPropertyInt 302  
getPropertyLong 302  
getPropertyStr 301  
getProtocol 347, 523  
getProtocolUsage 236, 296, 1007, 1332  
getProtocolUsageTracker 1007  
getPublishAudioCodecId 1119  
getPublisher 1250  
getPublisherCount 279  
getPublishers 278  
getPublishStreamNames 284, 1174  
getPublishStreams 335  
getPublishVideoCodecId 1119  
getPurgeController 494  
getQueryStr 345, 565, 921, 1112  
getQueryString 521  
getQueueSize 1344  
getRange 387  
getReceiveVideoFPS 1104  
getReconnectWaitTime 725  
getRecordedDuration 399, 458  
getRecordedEntries 398, 399, 452, 457  
getRecordedEntriesInRange 399, 453  
getRecordedEntriesMap 453  
getRecordedEntriesWithLimit 399, 457, 498  
getRecordedEntryByIndex 399, 453, 498  
getRecordedEntryByTime 498  
getRecordedEntryByTimeKey 399, 453  
getRecordedEntryStartingBeforeTime 399  
getRecordingName 1191  
getRecordingStreamStore 483

getRedirectSessionBody 574  
getRedirectSessionCode 573, 929  
getRedirectSessionContentType 575  
getRedirectSessionMessage 929  
getRedirectSessionURL 574, 929  
getRefCount 1016, 1040  
getReference 127  
getReferrer 346, 565, 841, 921  
getRemoteAddr 524  
getRemoteHost 524, 614  
getRendition 790  
getRenditions 790  
getRepeat 1219, 1249  
getRepeaterItemsDvr 489  
getRepeaterLastSequence 1198  
getRepeaterMediaCasterDef 547  
getRepeaterOriginUrl 266, 352, 840  
getRepeaterQueryString 267  
getRequestFilters 511, 518  
getRequestURI 521  
getRequestURL 522  
getRespAMFAudioObj 345, 1111  
getRespAMFDataObj 345, 1111  
getRespAMFVideoObj 345, 1111  
getRespFunctions 342  
getResponseAMFObj 345  
getRsoStorageDir 269  
getRsoStoragePath 269  
getRTPAVSyncMethod 272  
getRTPContext 948, 1326  
getRTPDatagramConfigIncoming 1326  
getRTPDatagramConfigOutgoing 1326  
getRTPDestination 974  
getRTPIidleFrequency 268, 1323  
getRTPIInfo 967  
getRTPMaxRTCPWaitTime 273  
getRTTPPlayAuthenticationMethod 272  
getRTTPProperties 275  
getRTTPPublishAuthenticationMethod 272  
getRTTPSession 306, 314, 899  
getRTTPSessionCount 274  
getRTTPSessionCountsByName 274  
getRTTPSessions 273  
getRTTPStream 354, 1115  
getRTTPWriteListener 919  
getRTSPBindIpAddress 287, 971  
getRTSPConnectionAddressType 288, 971  
getRTSPConnectionIpAddress 288, 971  
getRTSPMaximumPendingWriteBytes 286, 970  
getRTSPOriginAddressType 289, 972  
getRTSPOriginIpAddress 288, 971  
getRTSPPlayRangeStart 925  
getRTSPPlayRangeStop 925  
getRTSPSessionDescription 968  
getRTSPSessionExtraLines 974  
getRTSPSessionName 968  
getRTSPSessionTimeout 286, 970  
getRTSPStream 914  
getRTSPTunnelingLock 883  
getRTSPTunnelingSession 883  
getRTSPTunnelingSessionId 925  
getScheme 523  
getSDPData 898  
getSDPLang 974  
getSecondByte 218  
getSegment 777, 778  
getSeq 217  
getSerializeSize 446  
getServerGUID 1004  
getServerHostPort 347  
getServerIp 563, 920  
getServerName 523  
getServerPort 524, 564, 920  
getSession 932, 951  
getSessionDescriptionData 765  
getSessionGUID 1004  
getSessionId 558, 913  
getSessionIds 932  
getSessionProtocol 544, 563  
getSessions 880  
getSessionTimeout 560  
getSessionTracker 598  
getSessionType 562  
getSharedObjectReadAccess 271, 348  
getSharedObjects 255, 256  
getSharedObjectWriteAccess 271, 348  
getShort 140, 176, 193, 194, 225, 227  
getSize 204, 214, 659, 683, 714, 786  
getSlot 1015, 1039  
getSlotNames 1015, 1039

getSlots 1021, 1039  
getSlotVersion 1032  
getSoVersion 1032  
getSrc 205, 215, 1085  
getSSLConfig 1286  
getSslFactoryClass 1284  
getStart 1222  
getStartIndex 387  
getStartLiveOnPreviousBufferTime 1251  
getStartOnPreviousBufferTime 1239  
getStartStream 1200  
getStartTC 1050  
getStartTCOffset 1050  
getStartTimeCode 382  
getStartupStreams 1331  
getStopTimeCode 382  
getStorageDir 1020, 1028, 1045  
getStream 559, 721, 825, 951, 1174, 1175, 1187, 1229  
getStreamArray 751  
getStreamAttributes 962  
getStreamAudioSampleAccess 270, 349  
getStreamBaseName 482  
getStreamBitrate 839, 1278  
getStreamClientless 1174  
getStreamCount 263  
getStreamDef 1342  
getStreamDefs 1341  
getStreamExt 567, 749, 958  
getStreamFile 343, 344  
getStreamFileForRead 1110  
getStreamFileForWrite 1110  
getStreamFileMapper 276  
getStreamId 951  
getStreamInfo 952, 961  
getStreamIsRunningLock 725  
getStreamKeyDir 268  
getStreamKeyPath 269  
getStreamLastSeq 724  
getStreamLength 839, 1278  
getStreamList 752  
getStreamListLock 1173  
getStreamLock 948  
getStreamManager 755  
getStreamMissingTime 723  
getStreamName 493, 566, 958, 1207  
getStreamNameAliasProvider 278  
getStreamNameLock 1174  
getStreamNameLogging 959  
getStreamNameParts 567  
getStreamNames 762  
getStreamPacketizerItem 880, 927  
getStreamPort 888  
getStreamPosition 568, 1067  
getStreamProperties 275  
getStreamProperty 858  
getStreamQueryStr 964  
getStreamReadAccess 270, 350  
getStreams 253, 1091, 1174  
getStreamSessionId 959  
getStreamSessionIp 959  
getStreamSessionVersion 959  
getStreamStorageDir 268  
getStreamStoragePath 269  
getStreamStore 482  
getStreamStores 482  
getStreamTimeout 723  
getStreamTimeoutLastReset 724  
getStreamTimeoutLastTime 724  
getStreamTimeoutReason 724  
getStreamTrack 951  
getStreamType 254, 336, 736, 838, 957, 1101, 1229  
getStreamTypeNames 1342  
getStreamTypes 1313  
getStreamTypeStr 547  
getStreamVersionHandler 484  
getStreamVideoSampleAccess 269, 349  
getStreamWriteAccess 271, 351  
getString 140, 153, 175, 192, 193, 224, 226  
getStringReference 157  
getSystemTime 99  
getTargetEncoding 156  
getTextEncoding 665, 686, 701, 707, 710  
getThreadPool 986, 1003, 1316  
getTime 45  
getTimecode 205, 216, 703, 864  
getTimeMap 461, 505  
getTimeMapEntries 505  
getTimeMapper 493  
getTimeMapping 396  
getTimeOffsetBetweenItems 1252

getTimeRunning 234, 261, 343, 570, 928, 984, 1001, 1319  
getTimeRunningSeconds 234, 261, 343, 570, 928, 984, 1001, 1320  
getTimeSeconds 45  
getTimeStampFormat 701  
getTimeString 45  
getTotal 324  
getTotalAccepted 324  
getTotalIOPerformanceCounter 334  
getTotalRejected 325  
getTrack 893, 950  
getTrackId 950  
getTrackNames 950  
getTrait 153, 196  
getTraitReference 157  
getTranscoderApplicationContext 292  
getTranscoderName 1207  
getTranscoderProperties 276  
getTransportMode 966  
getTransportThreadPool 986, 1003, 1317  
getTTL 889, 896  
getType 125, 175, 205, 215, 383, 386, 398, 785, 865  
getTypeAsString 783  
getUDPManagedDeliveryCount 975  
getUDPManagedDeliveryDelay 975  
getUDPPortManager 988, 1004  
getUDPPortSharingManager 988, 1004  
getUDPTransport 893, 960  
getUDPTransportManager 881, 1329  
getUniqueId 747  
getUniqueStreamIdStr 1118  
getUnsignedShort 36  
getUri 346, 564, 921  
getURL 689, 709  
getUserAgent 564, 915  
getUserAgents 985, 1002  
getUserHTTPHeader 573  
getUserQueryStr 570  
getUserTime 99  
getUtcStartTime 382  
getValidationFrequency 265, 1321  
getValue 130, 142, 147, 165, 179, 195, 686, 704, 707  
getVersion 837, 982, 999, 1017, 1041  
getVHost 233, 253, 306, 314, 339, 543, 547, 558, 720, 824, 881, 915, 933, 952, 1178  
getVHostItems 1351  
getVHostList 987, 1004  
getVHostMap 1350  
getVHostNames 1351, 1354  
getVideoCodec 57, 65, 375  
getVideoCodecConfigPacket 1115  
getVideoCodecId 784  
getVideoFrameType 65  
getVideoHost 888  
getVideoMissing 1099  
getVideoPacketizerItem 880, 927  
getVideoPort 887  
getVideoSize 1085  
getVideoTC 1088  
getVideoTimecodeOffset 65  
getVideoTrack 951  
getVODLastTimeTC 972  
getVODPlayLen 973  
getVODStartTimeTC 972  
getWidth 786  
getWriteListener 351  
getXMLPropertyBool 109  
getXMLPropertyDouble 109  
getXMLPropertyExists 108  
getXMLPropertyInt 108  
getXMLPropertyIntSize 108  
getXMLPropertyLong 108  
getXMLPropertyLongSize 109  
getXMLPropertyStr 107  
gmtTimeZone 97  
grabFrame 1261  
GZIP 16

## H

handleArchivedStream 369, 501  
handleCallback 1105  
handleMessage 870, 896  
handleOnMetadata 1206  
handlePacket 1187, 1198, 1206  
hasAudio 460, 495  
hasCodecData 460  
hasData 460, 495  
hasEncryption 495  
hasOnMetadata 460, 495

- hasTimeMapData 461
- hasVideo 460, 495
- HEADERSIZE 673
- height 782
- hexadecimal 30
- HOSTNAME\_KEY 602
- HostPort 1283
- HostPortList 1288
- HTTPORIGINMODE\_AUTO 546
- HTTPORIGINMODE\_OFF 546
- HTTPORIGINMODE\_ON 546
- HTTPRequestToByteArray 69, 70
- HTTPRequestToFile 69
- HTTPProvider2Base 510
- HTTPStreamerSessionCupertino 532
- HTTPStreamerSessionMPEGDash 582
- HTTPStreamerSessionSanJose 586
- HTTPStreamerSessionSmoothStreamer 592
- HTTPStreamerSessionWebM 598
- HTTPUtils 69
- 
- I
- 
- ID3FOOTER\_SIZE 657
- ID3Frames 658
- ID3HEADER\_SIZE 657
- ID3HEADER\_VERSION 657
- ID3HEADERFLAGS\_DEFAULT 657
- ID3HEADERFLAGS\_EXPERIMENTAL 658
- ID3HEADERFLAGS\_EXTENDED 658
- ID3HEADERFLAGS\_FOOTERPRESENT 658
- ID3HEADERFLAGS\_UNSYNC 658
- ID3V2FrameAttachedPicture 665
- ID3V2FrameBase 681
- ID3V2FrameComment 685
- ID3V2FrameLinkedInformation 688
- ID3V2FramePrivate 691
- ID3V2FrameRawBytes 693
- ID3V2FrameRecommendedBufferSize 695
- ID3V2FrameSynchronizedText 700
- ID3V2FrameSynchronizedTextDescriptor 703
- ID3V2FrameTextInformation 706
- ID3V2FrameURLLink 709
- ID3V2Utils 711
- idle 965, 1090
- idleFrequency 912
- idleHandler 911
- idStringToName 739
- importManifest 459
- incAbsTimecode 206
- incByteContainerLevel 207
- incClientCountTotal 255
- incrementAccept 320
- incrementBytesIn 78
- incrementBytesLoss 80
- incrementBytesOut 80
- incrementFileIn 78
- incrementFileOut 79
- incrementMediaInBytes 949, 1105
- incrementMediaLossBytes 1105
- incrementMediaOutBytes 1104
- incrementMessagesIn 78, 79
- incrementMessagesLoss 79, 80, 81
- incrementMessagesOut 79, 80
- incrementReject 320
- incSlotVersion 1032
- index 381
- indexOf 37
- indexOfDifferent 37
- info 620, 621
- init 6, 308, 309, 403, 405, 468, 473, 492, 503, 504, 511, 518, 543, 546, 720, 747, 765, 879, 1031, 1056, 1063, 1083, 1144, 1167, 1186, 1197, 1206, 1313, 1344, 1353
- initContextLogging 627
- initialize 452
- initializeLogging 627
- initialManifest 489
- initialManifestEnd 489
- initLiveStreamRepeating 838, 1119, 1148
- initStream 835
- InputStream 23
- interleavePackets 60, 61
- interruptPlay 1147
- intToByteArray 34, 35
- intToByteArraySafeSync 712
- intToHexStr 94
- intToStr 94
- intValue 164
- invoke 978
- invokePrevious 825, 826

IOPerformanceCounter 77  
ioPerformanceCounter 910  
ioSession 913  
ip 909  
isAcceptConnection 254, 337  
isAcceptSession 566  
isActive 559, 614, 1188, 1197  
isAddressMulticast 91  
isAMF0 152, 157  
isAMF3 152, 157  
isAMF3Start 126  
isAnnounce 910, 918  
isAnnounceOrDescribe 919  
isAppend 1098  
isAppInstanceLoaded 232  
isApplicationLoaded 1315  
isArrayStart 126  
isAudio 217  
isAudioCodecConfig 64  
isAVSyncNonSR 961  
isBlockUDPOut 897, 975  
isByteArrayStart 127  
isByteContainerEmpty 207  
isByteContainerFull 207  
isCheckIpAddr 960  
isCheckSSRC 960  
isClient 1015, 1040  
isClustered 1109  
isConnected 336, 911, 919  
isDebugEnabled 619  
isDebugLog 83  
isDebugRTSPSession 924  
isDescribe 910, 918  
isDynamic 199  
isDynamicLogContextLoaded 987, 1004  
isEmpty 94, 388, 390, 398, 658  
isEnabledFor 619  
isEncoderInUse 1268  
isEncrypted 347  
isErrorEnabled 619  
isFileInfo 568  
isFirstABSTRequest 588  
isFlashMediaLiveEncoder 353  
isFlashVersion10 353  
isFlashVersion90115 353  
isFlashVersionH264Capable 353  
isForceAMF0 865  
isForceMPEGTSOut 975  
isForceRTSPInterleaved 967  
isGlobalLogValueSet 628  
isHitEnd 1054  
isHTTPOrigin 575  
isHTTPOriginOn 548  
isInfoEnabled 619  
isInMulticast 896  
isInRange 387, 390  
isIntData 151  
isLastSentAbsTimecode 208  
isLive 497, 969  
isLiveRepeater 352  
isLiveStreamPacketize 1203  
isLiveStreamTranscode 1210  
isLoaded 494  
isLoggedConnect 917  
isLongTimecode 208  
isMediaCasterPlay 1120  
isMember 198  
isMergeOnMetadata 1120  
isMixedArrayStart 127  
isModePlay 964  
isModePublish 964  
isModeUnknown 964  
isMPEGTSOut 968  
isMulticast 886, 896  
isNew 206  
ISO8601\_FORMAT 604  
ISO8601\_PATTERN 604  
isObjectEncodingAMF0 208, 354  
isObjectEncodingAMF3 208, 353  
isObjEnd 127  
isObjStart 126  
isOnMetadataPacket 66  
isOpen 7, 1057, 1064, 1102  
isOutMulticast 896  
isOverlayAvailable 1263  
isPacketizeAudio 578, 1199  
isPacketizeData 578, 1200  
isPacketizeVideo 578, 1199  
isPaused 954  
isPending 868

isPersistent 1016, 1027, 1040  
isPlay 1089  
isPlaying 1084  
isPlaylistReady 466, 467  
isPlayLogged 561  
isPublishStreamReady 953, 1120  
isPurgingEnabled 468  
isReceiveAudio 1103  
isReceiveVideo 1104  
isRecord 1089  
isRecording 485, 497, 1190  
isRecordingPaused 485, 497, 1190  
isRedirectSession 573, 928  
isRepeaterEdge 1199  
isResetPlayStream 954  
isResyncAudioVideoOnSR 956  
isRTPIgnoreProfileLevelId 975  
isRTPIncomingDatagramPortValid 282  
isRTPWrapped 889  
isRTSP 963  
isRTSPAlwaysUseSDPPorts 973  
isRTSPPull 963  
isRunning 1268  
isSecure 347, 524  
isSendOnMetadata 1216, 1236, 1251  
isSendPlayStopLogEvent 1113  
isSendPublishStopLogEvent 1114  
isSendRecordStopLogEvent 1113  
isSendResult 823  
isSendSDESEvents 968  
isSession 724  
isSessionValid 909, 916  
isShutdownClient 926  
isShutdownOnRelease 749  
isShuttingDown 1327  
isSSL 347  
isStartLiveOnPreviousKeyFrame 1251  
isStartOnPreviousKeyFrame 1239  
isStartStartupStreams 1331  
isStream 724, 887  
isStreamDomainProtectionActive 547  
isStreamIsRunning 725  
isStreamStarted 973  
isSuspended 988, 999, 1286, 1328  
isSwitchLog 1250

isTemplateLoaded 1209  
isTimeout 558, 970  
isTimeoutSession 560  
isTimesInMilliseconds 1251  
isTraceEnabled 619  
isTranscoderActive 1209  
isTranscodeResult 1122  
isValid 747, 1070  
isValidated 560  
isValidFMLEConnections 290, 357  
isValidAudioCodec 578  
isValidStreamDomainStr 569  
isValidVideoCodec 578  
isVersionFile 1161  
isVHostRunning 998  
isVideo 217  
isVideoCodecConfig 64  
isVideoH264SEIListenerEmpty 1123  
isVideoKeyFrame 64, 66  
isVisited 1347  
isWaitForVideoKeyFrame 1161  
isWarnEnabled 619  
isWowzaAudioOnly 785

## J

JNDI\_CONTEXT\_NAME 603

## K

KEYDATA\_MODE\_ENCRYPT 800  
KEYDATA\_MODE\_INIT 799  
KEYDATA\_MODE\_PLAYLIST 800  
killClient 1314  
killRTSPSession 1314

## L

lastAuthenticateMethod 909  
lastModified 8  
len 210  
length 9, 94, 1112, 1147  
LICENSECOUNTER\_DRM\_BUYDRM\_LIVE 1304  
LICENSECOUNTER\_DRM\_BUYDRM\_VOD 1304  
LICENSECOUNTER\_DRM\_EZDRM\_LIVE 1303

- LICENSECOUNTER\_DRM\_EZDRM\_VOD 1303
- LICENSECOUNTER\_DRM\_VERIMATRIX\_LIVE 1304
- LICENSECOUNTER\_DRM\_VERIMATRIX\_VOD 1304
- LICENSECOUNTER\_NDVR 1303
- LICENSECOUNTER\_PUBLISHER 1303
- LICENSECOUNTER\_PUBLISHERTRANSCODER 1304
- LICENSECOUNTER\_TOTAL 1305
- LICENSECOUNTER\_TRANSCODE\_DECODE 1303
- LICENSECOUNTER\_TRANSCODE\_DECODECOUNTAU  
DIO 1304
- LICENSECOUNTER\_TRANSCODE\_DECODECOUNTAU  
DIOVIDEO 1305
- LICENSECOUNTER\_TRANSCODE\_DECODECOUNTVID  
EO 1305
- LICENSECOUNTER\_TRANSCODE\_DECODEPOLLING  
1304
- LICENSECOUNTER\_TRANSCODE\_ENCODE 1303
- LICENSECOUNTER\_TRANSCODE\_ENCODECOUNTAU  
DIO 1305
- LICENSECOUNTER\_TRANSCODE\_ENCODECOUNTAU  
DIOVIDEO 1305
- LICENSECOUNTER\_TRANSCODE\_ENCODECOUNTVID  
EO 1305
- LICENSECOUNTER\_TRANSCODE\_ENCODEPOLLING  
1304
- LICENSECOUNTER\_TRANSCODE\_STREAMNAMES  
1304
- liveStreamPacketizers 1173
- load 1029, 1045
- loadArchivedRecordings 473
- loadArchivedStore 486
- loadConfig 1350
- loadConfigFile 1275, 1280
- loadConfigProperties 107
- loadFile 667
- lock 776, 781, 789, 913, 1022, 1045
- lockRepeaterStreams 565, 953
- log 619, 620
- LOG4J\_ID\_KEY 602
- LOG4J\_PACKAGE\_NAME 601
- logDebug 829
- logError 829
- loggedConnect 910
- LOGGERNAME\_SERVER 626
- logInfo 829
- logLiveChunk 533, 582, 587
- logLiveFragment 593
- logNotifier 997
- LogNotifyCalculateIncremental 609
- logVODChunk 533, 587
- logVODFragment 593
- logWarn 829
- longToByteArray 35, 36
- longToHexStr 94
- longToStr 94
- longValue 164
- M
- main 16, 998
- makeNewLoggerInstance 627
- MANIFEST\_TAGNAME\_CAN\_PLAY 451
- MANIFEST\_TAGNAME\_CAN\_RECORD 451
- MANIFEST\_TAGNAME\_CHUNK\_GROUPING 452
- MANIFEST\_TAGNAME\_CURRENT\_TIME 452
- MANIFEST\_TAGNAME\_HAS\_ENCRYPTION 451
- MANIFEST\_TAGNAME\_PURGE\_TIME 452
- 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 1171
- md5Digest 85
- MD5DigestUtils 85
- md5Lock 85
- MEDIACACHE\_PREFIX 418
- MediaCasterItem 736
- MediaCasterList 741
- MediaCasterSettings 743



MediaCasterStreamItem 746  
MediaCasterStreamManager 751  
MediaCasterStreamMap 754  
MEDIACASTERTYPE\_LIVEREPEATER 720  
MEDIACASTERTYPE\_RTPLIVE 720  
MEDIACASTERTYPE\_SHOUTCAST 720  
MEDIACASTERTYPE\_UNKNOWN 720  
MediaList 776  
mediaList 789  
MediaListRendition 783  
MediaListSegment 789  
mediaListSegment 781  
MediaStreamBase 1167  
mediaStreamListeners 1173  
MediaStreamMap 1173  
MediaUtils 89  
members 190  
MILLS\_PER\_HOUR 124  
MIMETYPE\_VIDEO\_MP4 441  
MIMETYPES\_JPEG 664  
MIMETYPES\_PNG 664  
MIMETYPES\_URL 664  
MISSING 1052  
MODE\_PLAY 946  
MODE\_PUBLISH 946  
MODE\_UNKNOWN 946  
ModuleBase 819  
ModuleClientLogging 829  
ModuleCore 834  
ModuleFastPlay 843  
ModuleMediaCaster 761  
ModuleMediaCasterStreamMonitorAdvanced 853  
ModuleProperties 856  
monitors 852  
msb0baseTime 97  
msb1baseTime 97  
MSG\_STARTSTREAM 549  
MSG\_SWITCHSTREAM 549

N

name 776, 782  
nameGroupId 1172  
nameGroups 1172  
NetworkUtils 91  
newValue 100  
newXPathFactory 109  
next 1249  
nextStreamId 1173  
NO\_OPTIONS 15  
notifyActionOnCodecInfoAudio 1109  
notifyActionOnCodecInfoVideo 1109  
notifyActionOnMetaData 1108  
notifyActionPause 1108  
notifyActionPauseRaw 1108  
notifyActionPlay 1107  
notifyActionPublish 1108  
notifyActionSeek 1108  
notifyActionStop 1109  
notifyActionUnPublish 1108  
notifyDvrStreamManagerCreate 294  
notifyDvrStreamManagerDestroy 295  
notifyDvrStreamManagerInit 294  
notifyDvrStreamStorageDeleted 488  
notifyDvrStreamStoreCreate 488  
notifyDvrStreamStoreDestroy 488  
notifyDvrStreamStoreInit 488  
notifyDvrStreamStoreLoaded 488  
notifyHTTPSessionCreate 533, 540, 588, 593  
notifyHTTPSessionDestroy 533, 540, 588, 593  
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 1179  
notifyMediaStreamDestroy 1180  
notifyMediaWriterOnFLVAddMetadata 284  
notifyMediaWriterOnWriteComplete 284  
notifyPlayPublish 1179  
notifyPlayUnpublish 1178

notifySessionCreate 934  
notifySessionDestroy 934  
notifySlotDelete 1045  
notifySlotSetValue 1045  
notifyTimeReset 488  
notifyVHostClientConnect 1354  
notifyVHostCreate 1355  
notifyVHostInit 1355  
notifyVHostItemCreate 1350  
notifyVHostItemDestroy 1351  
notifyVHostItemUpdate 1351  
notifyVHostShutdownComplete 1355  
notifyVHostShutdownStart 1355  
notifyVideoH264Packet 1123

## O

offset 210  
ON\_METADATA\_TYPE 451  
onAACEncodeInfo 772  
onAcceptConnection 361  
onAcceptLicense 364  
onAcceptorCreate 1290  
onAcceptorDestroy 1290  
onAfterDecodeFrame 1265  
onAfterEncodeFrame 1267  
onAfterScaleFrame 1265  
onAnnounce 874, 922  
onApplicationCreate 298  
onApplicationDestroy 298  
onApplicationInstanceCreate 297  
onApplicationInstanceDestroy 297  
onAppStart 761, 795, 853  
onAppStop 761, 795, 853  
onBeforeDecodeFrame 1265  
onBeforeEncodeFrame 1267  
onBeforeScaleFrame 1265  
onBind 512, 514  
onCalculateSourceAudioBitrate 1256  
onCalculateSourceVideoBitrate 1256  
onCall 796  
onCallback 1129  
onClientAccept 359  
onClientConnect 359  
onClientDisconnect 359

onClientReject 360  
onCodecConfigAAC 769  
onCodecInfoAudio 1128  
onCodecInfoVideo 1128  
onConnect 797  
onConnectAccept 797  
onConnectFailure 732  
onConnectReject 798  
onConnectStart 731  
onConnectSuccess 731  
onCreate 534  
onData 727  
onDescribe 874, 922  
onDestroy 536  
onDisconnect 362, 365, 767, 797  
onDvrStreamManagerCreate 1184  
onDvrStreamManagerDestroy 1184  
onDvrStreamManagerInit 1184  
onFile 74  
onFillChunkDataPacket 537  
onFillChunkEnd 536  
onFillChunkStart 536  
onFLVAddMetadata 1163  
onFrameAAC 769  
onFrameMP3 769  
onGetParameter 874, 922  
onGrabFrame 1262  
onHeaderData 770, 772  
onHTTPCupertinoEncryptionKeyCreateLive 801  
onHTTPCupertinoEncryptionKeyCreateVOD 800  
onHTTPCupertinoEncryptionKeyData 800  
onHTTPCupertinoEncryptionKeyLiveChunk 801  
onHTTPCupertinoEncryptionKeyRequest 800  
onHTTPCupertinoEncryptionKeyVODChunk 801  
onHTTPCupertinoStreamingSessionCreate 802  
onHTTPCupertinoStreamingSessionDestroy 802  
onHTTPRequest 514  
onHTTPSsanJoseStreamingSessionCreate 803  
onHTTPSsanJoseStreamingSessionDestroy 803  
onHTTPSessionCreate 804  
onHTTPSessionDestroy 804  
onHTTPSmoothStreamingPlayReadyCreateLive 805  
onHTTPSmoothStreamingPlayReadyCreateVOD 805  
onHTTPSmoothStreamingSessionCreate 806  
onHTTPSmoothStreamingSessionDestroy 806

onHTTPStreamerSessionCreate 576  
onHTTPStreamerSessionDestroy 576  
onIndex 535  
onInit 535  
onInitAfterLoadTemplate 1256  
onInitBeforeLoadTemplate 1256  
onInitStart 1255  
onInitStop 1256  
onLiveStreamDvrRecorderCreate 1192  
onLiveStreamDvrRecorderDestroy 1192  
onLiveStreamDvrRecorderInit 1192  
onLiveStreamPacketizerCreate 1201  
onLiveStreamPacketizerDestroy 1201  
onLiveStreamPacketizerInit 1201  
onLiveStreamTranscoderCreate 1211  
onLiveStreamTranscoderDestroy 1211  
onLiveStreamTranscoderInit 1211  
onLog 606, 610  
onMediaCasterCreate 729  
onMediaCasterDestroy 729  
onMediaReaderClose 1069  
onMediaReaderCreate 1068  
onMediaReaderExtractMetaData 1069  
onMediaReaderInit 1068  
onMediaReaderOpen 1068  
onMediaStreamCreate 1139  
onMediaStreamDestroy 1139  
onMetaData 770, 772, 1127  
onModuleLoad 794  
onModuleUnload 794  
onMP3EncodeInfo 772  
onNewVHost 1001  
onOpen 535  
onOptions 874, 923  
onPause 875, 923, 1125  
onPauseRaw 1127  
onPlay 875, 923, 1124  
onPlaylistItemStart 1217  
onPlaylistItemStop 1217  
onPublish 1125  
onRecord 875, 923  
onRedirect 875, 923  
onRegisterPlayer 729  
onRegisterStreamNameGroup 1259  
onRejectConnection 361  
onRejectLicense 364  
onResetMediaCaster 734, 854  
onResetStream 1260  
onResponseWriteStart 979  
onResponseWriteStop 979  
onResult 793, 809  
onRTPSessionCreate 807, 872  
onRTPSessionDestroy 807, 872  
onSeek 1125  
onServerConfigLoaded 992  
onServerCreate 990  
onServerInit 990  
onServerShutdownComplete 991  
onServerShutdownStart 990  
onSessionAudioDecodeCodecInfo 1259  
onSessionAudioEncodeCodecInfo 1258  
onSessionAudioEncodeCreate 1257  
onSessionAudioEncodeInit 1257  
onSessionAudioEncodeSetup 1258  
onSessionDataEncodeCreate 1257  
onSessionDataEncodeInit 1258  
onSessionDestinationCreate 1256  
onSessionVideoDecodeCodecInfo 1259  
onSessionVideoEncodeCodecInfo 1258  
onSessionVideoEncodeCreate 1257  
onSessionVideoEncodeInit 1257  
onSessionVideoEncodeSetup 1258  
onSetParameter 874, 923  
onSetSourceStream 730  
onSetup 876, 924  
onSharedObjectConnect 1023  
onSharedObjectCreate 1023  
onSharedObjectDestroy 1023  
onSharedObjectDisconnect 1024  
onShutdownStart 1259  
onShutdownStop 1260  
onSlotDelete 1034  
onSlotSetValue 1034  
onStop 1125  
onStreamCreate 808, 853  
onStreamDestroy 808, 853  
onStreamStart 732, 1134  
onStreamStop 732  
onTeardown 876, 924  
onTrim 770

onUnbind 512, 515  
onUnPublish 1125  
onUnRegisterPlayer 730  
onUnregisterStreamNameGroup 1259  
onValidateMediaCaster 733, 853  
onValidateMediaCastersStart 733, 853  
onValidateMediaCastersStop 733, 854  
onVHostClientConnect 1335  
onVHostCreate 1334  
onVHostInit 1334  
onVHostItemCreate 1333  
onVHostItemDestroy 1333  
onVHostItemUpdate 1333  
onVHostShutdownComplete 1335  
onVHostShutdownStart 1334  
onVideoH264Packet 1132  
onWriteComplete 1163  
open 6, 1057, 1063, 1219  
order 190  
ORDERED 16  
output 868  
OutputStream 26

## P

p 1167  
PACKET\_TIME 445  
packetComplete 1091  
packetizerLicenses 1172  
packetOutput 1279  
packetTime 381  
PARAM1 814  
PARAM10 816  
PARAM2 815  
PARAM3 815  
PARAM4 815  
PARAM5 815  
PARAM6 815  
PARAM7 815  
PARAM8 815  
PARAM9 815  
PARAMMETHODNAME 814  
parseAllowDomains 267  
parseBodyForParams 522  
parseIdString 739

parseQueryStr 104  
PASSWORDFILEFORMAT\_CLEAR 308  
PASSWORDFILEFORMAT\_UNKNOWN 308  
pathToFileURL 102  
pause 837, 956, 1145  
PAUSE\_PAUSE 1142  
PAUSE\_PLAY 1142  
PAUSE\_TOGGLE 1142  
pauseRaw 841, 1145  
pauseRecording 486, 496, 1190  
peekByte 125  
PICTUREMAXFILESIZE 664  
PICTURETYPE\_ARTISTLOGO 664  
PICTURETYPE\_COVERBACK 664  
PICTURETYPE\_COVERFRONT 663  
PICTURETYPE\_FILEICON 663  
PICTURETYPE\_ILLUSTRATION 664  
PICTURETYPE\_MOVIESCREENCAPTURE 664  
PICTURETYPE\_OTHER 663  
PICTURETYPE\_OTHERFILEICON 663  
PICTURETYPE\_PUBLISHERLOGO 664  
ping 340  
play 836, 955, 1144, 1145, 1214, 1238, 1241, 1247, 1248, 1249  
play2 836  
playDuration 783  
PLAYEVENT\_AFTERBUFFERFILL 1062  
PLAYEVENT\_AFTERMETADATA 1062  
PLAYEVENT\_BEFOREBUFFERFILL 1062  
PLAYEVENT\_BEFOREMETADATA 1062  
PLAYEVENT\_STARTPLAYBACK 1062  
Playlist 1218  
PlaylistItem 1221  
PLAYSIZES\_AUDIO\_BYTES 1143  
PLAYSIZES\_AUDIO\_COUNT 1143  
PLAYSIZES\_DATA\_BYTES 1143  
PLAYSIZES\_DATA\_COUNT 1143  
PLAYSIZES\_LOSS\_BYTES 1143  
PLAYSIZES\_LOSS\_COUNT 1144  
PLAYSIZES\_SIZE 1144  
PLAYSIZES\_VIDEO\_BYTES 1143  
PLAYSIZES\_VIDEO\_COUNT 1143  
playStart 783  
PLAYSTATUSTYPE\_COMPLETE 1143  
PLAYSTATUSTYPE\_STOP 1143  
PLAYSTATUSTYPE\_SWITCH 1142

- 
- PLAYTRANSITION\_APPEND 818
- PLAYTRANSITION\_APPEND\_IMMEDIATE 818
- PLAYTRANSITION\_RESET 818
- PLAYTRANSITION\_RESET\_IMMEDIATE 818
- PLAYTRANSITION\_STOP 818
- PLAYTRANSITION\_SWAP 818
- PLAYTRANSITION\_SWITCH 818
- PLAYTRANSITION\_UNKNOWN 818
- PLAYTRANSITIONSTR\_APPEND 817
- PLAYTRANSITIONSTR\_RESET 817
- PLAYTRANSITIONSTR\_STOP 817
- PLAYTRANSITIONSTR\_SWAP 817
- PLAYTRANSITIONSTR\_SWITCH 817
- PLAYTRANSITIONSTR\_UNKNOWN 817
- postChunkAdded 475
- postChunksPurged 476
- preChunkAdded 475
- preChunksPurged 475
- previous 1249
- properties 510, 776, 781, 789, 910
- PROPERTY\_ALLOWABLE\_AV\_PACKET\_DELTA 424
- PROPERTY\_APPEND\_DISCONTINUITY\_DELTA 418
- PROPERTY\_ARCHIVE\_STRATEGY 420
- PROPERTY\_AUDIO\_ONLY\_CHUNK\_TARGET\_DURATION 426
- PROPERTY\_BREAK\_ON\_PTS 427
- 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 443
- PROPERTY\_CUPERTINO\_PLAYLIST\_FORCE\_LIVE 442
- PROPERTY\_CUPERTINO\_PLAYLIST\_FORCE\_NONLIVE 442
- PROPERTY\_CUPERTINO\_PLAYLIST\_GZIP\_THRESHOLD 443
- PROPERTY\_CUPERTINO\_PLAYLIST\_USE\_GZIP 442
- PROPERTY\_DEBUG\_CHUNK\_RETRIEVALS 436
- PROPERTY\_DEBUG\_CUPERTINO\_PLAYER\_ADAPTER 438
- 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 435
- 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 439
- PROPERTY\_DEBUG\_MBR\_PLAYER\_ADAPTER 438
- PROPERTY\_DEBUG\_METHODS 437
- PROPERTY\_DEBUG\_PLAYER\_ADAPTER 437
- PROPERTY\_DEBUG\_PLAYLIST\_REQUEST 439
- PROPERTY\_DEBUG\_RAW\_PACKETS 435
- PROPERTY\_DEBUG\_RAW\_PACKETS\_MATCHER 435
- PROPERTY\_DEBUG\_REPEATER 437
- PROPERTY\_DEBUG\_SANJOSE\_PLAYER\_ADAPTER 438
- PROPERTY\_DEBUG\_SMOOTH\_PLAYER\_ADAPTER 438
- PROPERTY\_DEBUG\_STATE\_CHANGE 436
- PROPERTY\_DEBUG\_TOSSED\_HOLDERS 435
- PROPERTY\_ENCRYPTION\_INFO\_DELEGATE 439
- PROPERTY\_FILE\_SYSTEM\_CLASS 421
- PROPERTY\_MANIFEST\_PERSISTER\_CLASS 422
- PROPERTY\_MAX\_CHUNK\_LOG 436
- PROPERTY\_MAX\_RECALC\_DURATION\_LOG 437
- PROPERTY\_MAX\_RECORDING\_LENGTH 430
- PROPERTY\_MBR\_MINIMUM\_PACKETTIME\_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 429
- PROPERTY\_PACKET\_SORT\_TIME 426
- PROPERTY\_PLAYLIST\_REQUEST\_DELEGATE 439
-

- 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 442  
PROPERTY\_SANJOSE\_ABST\_TIMESCALE 441  
PROPERTY\_SANJOSE\_PLAYLIST\_DELIVERYTYPE 439  
PROPERTY\_SANJOSE\_PLAYLIST\_LIVE\_STREAMTYPE 440  
PROPERTY\_SANJOSE\_PLAYLIST\_MIMETYPE 439  
PROPERTY\_SANJOSE\_PLAYLIST\_RECORDED\_STREAMTYPE 440  
PROPERTY\_SANJOSE\_PLAYLIST\_VERSION 440  
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 432  
PROPERTY\_SMOOTH\_MANIFEST\_VERBOSE\_DURATION 433  
PROPERTY\_START\_RECORDING\_ON\_STARTUP 426  
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 653  
PROTO\_HTTPDVRCHUNK 653  
PROTO\_HTTPSANJOSE 653  
PROTO\_HTTPSCUPERTINO 653  
PROTO\_HTTPSDVRCHUNK 653  
PROTO\_HTTPSMOOTH 653  
PROTO\_HTTPSSANJOSE 653  
PROTO\_HTTPSSMOOTH 653  
PROTO\_HTTPSSTREAMER 653  
PROTO\_HTTPSTREAMER 652  
PROTO\_RTMP 652  
PROTO\_RTMPE 652  
PROTO\_RTMPS 652  
PROTO\_RTMPT 652  
PROTO\_RTMPTE 652  
PROTO\_RTMPTS 652  
PROTO\_RTSP 652  
ptToDvr 506  
publish 835, 1105, 1167, 1229, 1230  
PublishingProviderBase 1235  
PublishingProviderLive 1238  
PublishingProviderMediaReader 1241  
purge 1021, 1039  
purgeEntries 454, 497  
purgeManifestEntries 487  
put 191, 192, 222, 223, 1026  
putAll 300  
putAudioPacketizerItem 879, 927  
putDvrRecorder 1121  
putFileInfo 568  
putFrame 659  
putGlobalLogValue 628  
putHTTPStreamerAdapter 1330  
putLiveStreamTranscoder 1122  
putMetaData 1162  
putRTSPSessionExtraLine 974  
putRTSPTunnelingSession 884  
putSlot 1015, 1040  
putStreamNameParts 567  
putStreamPacketizerItem 880, 928  
putVideoPacketizerItem 880, 928

## Q

QUERY\_PARAM\_DVR 466

queryStr 911

## R

ranges 389

read 7, 23

READ\_ACCESS\_ALL 333

READ\_ACCESS\_NONE 333

READACCESS 1038, 1083

readAppConfig 235

readAppInstConfig 282

readChunk 58

readChunkHeader 58

readConfig 988, 1005

readHeader 58, 59

readPrevChunkHeader 58

readVHostConfig 1332

readXMLConfig 1005

receiveAudio 839

RECEIVER\_NAME\_KEY 602

receiveVideo 839

record 955

recordingPaused 471

recordingReset 472

recordingResumed 471

recordingStarted 471

recordingStopped 472

redirectConnection 338, 339

redirectSession 574, 912, 929

redirectSessionCode 912

redirectSessionMessage 913

redirectSessionURL 913

referrer 911

refreshManifest 452

registerCallback 1106

registerLiveStreamTranscoder 1007

registerOnPlayStatus 1106

registerOnStatus 1106

registerPlayer 722, 748, 759

registerPlayRTPSession 274

rejectConnection 320, 337, 338

rejectSession 565, 913

release 749, 758, 1016, 1040

releaseMediaCaster 763

releaseSessionId 932

releaseSocketAddress 881

releaseStream 835

releaseUDPPortPair 882

reloadConfig 1350

reloadVHostConfig 983, 999

remove 135, 171, 192, 224, 758, 1027, 1354

removeAcceptorListener 1318

removeActionListener 922

removeAppInstance 235

removeApplication 1332

removeApplicationInstanceListener 233

removeApplicationListener 1318, 1354

removeClient 1016, 1028, 1040, 1314

removeClientListener 257, 1107

removeConnectionListener 319

removeDvrChunkListener 498

removeDvrRecorder 1121, 1182

removeDvrRecorderListener 293

removeDvrRecordingListener 488

removeDvrStoreListener 487

removeDvrStreamManagerListener 294

removeFrame 659

removeFromPlaylist 1248

removeGlobalLogValue 628

removeGlobalLogValues 627

removeHeader 527

removeHTTPStreamerSession 281

removeIdleWorkerListener 1325

removeItem 1219

removeListener 1250

removeLiveStreamPacketizer 1180

removeLiveStreamPacketizerListener 289

removeLiveStreamTranscoder 1122

removeLiveStreamTranscoderListener 290

removeMediaCasterListener 260

removeMediaReaderListener 286

removeMediaStreamListener 258, 1179

removeMediaWriterListener 284

removeMetaDataListener 771

removeModuleListener 264

removeNameGroup 1181

removePlayStreamByName 262  
removePublisher 279  
removeRendition 790  
removeRTPSession 274  
removeRTSPStream 914  
removeRTSPTunnelingSession 883  
removeSegment 777  
removeServerListener 985, 1002  
removeSession 933  
removeSessionListener 934  
removeSharedObjectListener 259, 1028  
removeSlotListener 1020, 1045  
removeStream 1177  
removeStreamDomainStr 569  
removeTrack 949  
removeTrackInternal 949  
removeVHostItemListener 1350  
removeVideoH264SEILListener 1123  
renditions 789  
reparentClient 355, 1327  
ReplaceItem 100  
requestFilters 510  
requiresLayout 614  
reset 492, 748, 777, 1144, 1336, 1346  
resetMDC 627  
resetMediaCasterStream 283  
resetNoLookup 1144  
resetSentMetadataFlag 957  
resetStream 465, 484, 578, 762, 1187, 1198, 1207  
resetTimecodes 1149  
resolveMediaList 1055  
resolvePlayAlias 1135, 1136, 1137  
resolveStreamAlias 1135, 1137  
ResponseFunction 861, 862  
ResponseFunctions 867  
resumeEncoding 27  
resumeRecording 486, 496, 1190  
retrieveRawChunk 403, 470  
REVERSE 6  
rewind 1064  
RTPContext 879  
RTPDestination 886  
RTPPort 893  
RTPPushPublishSession 898  
RTPSession 913

rtpSession 306  
RTPSessions 932  
RTPStream 948  
RTPUtils 1272  
rtpWriteListener 910  
rtspPlayRangeStart 912  
rtspPlayRangeStop 912  
rtspTunnelingSessionId 908  
run 1246

## S

SANJOSE\_F4M\_STREAMINGTYPE\_STREAMING 441  
SANJOSE\_F4M\_STREAMTYPE\_DVR 440  
SANJOSE\_F4M\_STREAMTYPE\_LIVE 441  
SANJOSE\_F4M\_STREAMTYPE\_LIVEORRECORDED 441  
SANJOSE\_F4M\_STREAMTYPE\_RECORDED 441  
SANJOSE\_F4M\_VERSION\_1\_0 441  
SANJOSE\_F4M\_VERSION\_2\_0 441  
SDPLOCATION\_AUDIO 945  
SDPLOCATION\_STREAM 945  
SDPLOCATION\_VIDEO 945  
seek 7, 836, 1064, 1145, 1215, 1238, 1241  
SEEK\_EXACT 1063  
SEEK\_KEYCLOSE 1063  
SEEK\_KEYDOWN 1062  
SEEK\_KEYUP 1062  
SEEKTARGET\_AUDIO 1062  
SEEKTARGET\_ENHANCED 1062  
SEEKTARGET\_VIDEOKEYFRAME 1061  
segments 776  
send 1020, 1044, 1102  
sendAMF3 1102  
sendClientOnStatusError 826  
sendControlBytes 1111  
sendDirect 1101, 1102  
sendDirectAMF3 1101  
sendInternal 1044  
sendLivePlaySeek 1092  
sendLivePlayStart 1091  
sendLivePlaySwitch 1092  
sendMessage 895  
sendOnMetadata 464, 1235  
sendPauseNotify 1097  
sendPlayReset 1093



---

sendPlaySeek 1094  
sendPlayStart 1094, 1095, 1096  
sendPlayStatus 1096, 1097  
sendPlayStop 1093  
sendPlaySwitch 1095  
sendResponse 895  
sendResult 824, 825  
sendStreamNotFound 1091  
sendStreamOnStatusError 826  
sendUnpauseNotify 1097, 1098  
sendVODPlaySwitch 1092  
sendZeroLengthPacket 1066  
serialize 129, 142, 148, 165, 166, 178, 183, 195, 372, 375, 383, 387, 390, 394, 396, 447, 461, 660, 682, 714  
SERIALIZE\_CURRENT\_VERSION 381  
serializeBody 665, 685, 688, 691, 693, 695, 700, 706, 709, 714  
serializeDate 113  
serializeFooter 659  
serializeHeader 659  
serializeInt 113  
serializeString 115, 682  
serializeStringLen 682  
serializeStringNoLength 115  
serializeTags 659  
serializeZeroLengthString 114  
Server 997  
serverIp 911  
serverPort 911  
service 542  
sessionClosed 723, 896  
sessionId 908  
sessionOpened 722, 896  
SESSIONPROTOCOL\_COUNT 558  
SESSIONPROTOCOL\_CUPERTINOSTREAMING 557  
SESSIONPROTOCOL\_DVRCHUNKSTREAMING 557  
SESSIONPROTOCOL\_MPEGDASHSTREAMING 557  
SESSIONPROTOCOL\_SANJOSESTREAMING 557  
SESSIONPROTOCOL\_SMOOTHSTREAMING 557  
SESSIONPROTOCOL\_UNKNOWN 557  
SESSIONPROTOCOL\_WEBMSTREAMING 557  
sessionStart 765  
sessionStop 765  
SESSIONTYPE\_LIVE 556  
SESSIONTYPE\_LIVEDVR 557  
SESSIONTYPE\_UNKNOWN 556  
SESSIONTYPE\_VOD 556  
set 138, 139, 174, 175  
setAbsTimecode 217  
setAbsTimecodeLong 206  
setAbsTimecodeShort 206  
setAcceptConnection 254, 355  
setAcceptConnectionDescription 354  
setAcceptConnectionExObj 354  
setAcceptConnectionObj 354  
setAcceptSession 566  
setActive 559  
setAddressCount 897  
setAdminInterfaceHostPort 1317  
setAllowDomains 267  
setAnnounce 918  
setAppend 1098  
setAppInstance 562, 721, 917, 1208  
setAppInstanceName 958  
setAppInstanceProperty 857  
setApplication 614  
setApplicationInstanceTouchTimeout 285  
setApplicationProperty 856  
setApplicationTimeout 264, 1321  
setAppName 957  
setAudioCodecId 785  
setAudioHost 889  
setAudioPort 888  
setAudioSize 1085  
setAudioTC 1087, 1088  
setAuthenticatePlayHandler 916  
setAuthenticatePublishHandler 916  
setAuthenticationMethod 512, 518  
setAVSyncMethod 961  
setBandwidthLimit 836  
setBaseClass 738, 1338  
setBitrateAudio 784  
setBitrateTotal 784  
setBitrateVideo 784  
setBlockUDPOut 897, 976  
setBody 864  
setBufferSize 696  
setBufferTime 344, 837, 1084, 1146  
setByteContainerLevel 207  
setCanPlay 495  
setCanRecord 494

---

setCheckIpAddr 960  
setCheckSSRC 960  
setChecksum 1053  
setChunkCounter 209  
setChunkGroupDuration 402  
setClassName 196, 199  
setClient 307, 314, 1091  
setClientIdleFrequency 268, 1323  
setClientProperty 857  
setClientTimeout 1316  
setClustered 1109  
setCodecInfoAudio 465, 578  
setCodecInfoVideo 465, 578  
setCommandInterface 1002  
setCommandInterfaceHostPort 985, 1001  
setConfigDir 1346  
setConnected 919  
setConnectionLimit 1347  
setConnectionTimeout 738  
setConnectionValidator 319  
setContentTypes 702  
setCookieStr 566, 926  
setCoreHandlerPoolSize 986, 1003, 1316  
setCoreTransportPoolSize 986, 1003, 1315  
setCurrentDvrTime 468  
setData 689, 1031  
setDataBuffer 215  
setDataSize 1085  
setDataTC 1088, 1089  
setDataTypes 1089  
setDebugLog 83  
setDebugRTSPSession 925  
setDefaultStreamingStore 483  
setDescribe 918  
setDescription 666, 686, 689, 738, 1340  
setDirection 1050  
setDirecton 8  
setDomainName 1283  
setDuration 1215, 1239, 1242  
setDvrRecorder 1121  
setDvrRecorderId 1188  
setDvrRecorderList 293, 1117  
setDvrSessionInfo 571  
setDynamic 199  
setDynamicLogProperties 987, 1004, 1328  
setEmbeddedFlag 696  
setEncoding 614  
setEncryptions 372  
setExt 1113  
setFastPlay 844  
setFastPlaySettings 341, 1103  
setFileExtension 1229  
setFileIOPoolSize 1316  
setFlags 683, 714  
setFlashVer 334  
setForceAMF0 865  
setForceMPEGTSOut 975  
setForceRTSPInterleaved 967  
setFps 1049  
setFrameCount 1268  
setFrameIdentifier 689  
setHandler 895  
setHasEncryption 495  
setHeader 527  
setHeaderSize 1115  
setHeight 786  
setHost 887, 952  
setHostType 889  
setHTTPOrigin 575  
setHTTPOriginMode 548  
setHTTPStreamerAdapter 561  
setHTTPStreamerItem 543  
setHTTPStreamerList 277  
setHTTPStreamerSession 1058, 1118  
setID 544  
setId 204  
setIdleCheckFrequency 1324  
setIdleFrequency 348, 924  
setIdleHandler 920  
setIdleMinimumWaitTime 1327  
setIdleWorkerCount 1324  
setIdStr 682, 713  
setIndex 1222  
setInnerObj 200  
setIntData 151  
setIntHeader 527  
setIOPerformanceCounter 926  
setIoPerformanceCounter 1000  
setIoSession 930  
setIp 918

setIpAddress 563, 1284  
setIsPlaying 1084  
setKeepAliveTime 739  
setKeepAliveTimeout 1324  
setLanguage 701  
setLastAccessed 1053  
setLastAuthenticateMethod 917  
setLastClientId 1033  
setLastOperation 1033  
setLastSentAbsTimecode 208  
setLastValidateTime 352  
setLive 969  
setLivePlaylist 593  
setLiveRepeaterCapabilities 356  
setLiveStreamDvrRecorderControl 293  
setLiveStreamingPacketizer 562  
setLiveStreamPacketizer 750, 837, 1118  
setLiveStreamPacketizerControl 282  
setLiveStreamPacketizerId 1198  
setLiveStreamPacketizerList 277, 356, 1116  
setLiveStreamRepeater 750, 1119  
setLiveStreamTranscoderControl 292  
setLiveStreamTranscoderItem 1208  
setLiveStreamTranscoderList 291, 356, 1117  
setLock 783, 791  
setLoggedConnect 917  
setLongTimecode 208  
setMaximumPendingReadBytes 266, 1322  
setMaximumPendingWriteBytes 265, 1322  
setMaximumSetBufferTime 266, 1322  
setMaxRTCPWaitTime 961  
setMaxStorageDirDepth 276  
setMediaCasterDef 722  
setMediaCasterHostPortConfig 744  
setMediaCasterId 721  
setMediaCasterItem 1133  
setMediaCasterPlay 1120  
setMediaCasterProcessorCount 744  
setMediacasterRTPRTSPRTPTransportMode 295  
setMediaCasterType 726  
setMediaCasterValidator 285  
setMediaList 791  
setMediaListProvider 295  
setMediaListSegment 784  
setMediaReaderItem 1057, 1063  
setMediaWriterItem 1161  
setMergeOnMetadata 1121  
setMessageBytes 863  
setMetaDataProvider 1114  
setMimeType 666  
setMode 966  
setMPEGTSAudioBitrate 976  
setMPEGTSAudioLanguage 962  
setMPEGTSAudioPID 963  
setMPEGTSOut 969  
setMPEGTSProgramID 962  
setMPEGTSVideoBitrate 976  
setMPEGTSVideoPID 962  
setMultiplier 1049  
setName 232, 253, 737, 778, 785, 890, 1019, 1032, 1044, 1086, 1146, 1338, 1346  
setNetConnection 1090  
setNetConnectionIdleFrequency 1323  
setNetConnectionProcessorCount 1318  
setNew 206  
setObjectEncoding 152, 156, 208, 354  
setOffsetToNextTag 696  
setOpen 1103  
setOutHost 964  
setPacketizeAudio 1199  
setPacketizeData 1200  
setPacketizeVideo 1199  
setParent 1046, 1161  
setPersistent 1017, 1027, 1041  
setPictureData 666  
setPictureDataAsURL 666  
setPictureType 666  
setPingTimeout 265, 1321  
setPlay 1090  
setPlayClass 1339  
setPlayDuration 572, 787  
setPlayer 1099  
setPlayLogged 561  
setPlayStart 572, 786  
setPollingInterval 1250  
setPort 615, 1284  
setProcessorCount 1285  
setProperty 512, 515, 542, 1057, 1067, 1347  
setProperty 301, 737, 1018, 1019, 1041, 1042, 1339  
setPublishAudioCodecId 1119

setPublishVideoCodecId 1119  
setQueryStr 565, 921, 1112  
setRange 387  
setRealTimeStartTime 1215, 1239, 1242  
setReceiveAudio 1103  
setReceiveVideo 1104  
setReceiveVideoFPS 1104  
setReconnectWaitTime 726  
setRecord 1089  
setRecordAudio 1189  
setRecordData 1189  
setRecordingName 1190  
setRecordingStreamStore 483  
setRecordVideo 1189  
setRedirectSession 573, 928  
setRedirectSessionBody 574  
setRedirectSessionCode 573, 929  
setRedirectSessionContentType 575  
setRedirectSessionMessage 929  
setRedirectSessionURL 574, 929  
setReferrer 565, 921  
setRemoteHost 614  
setRepeat 1219, 1249  
setRepeaterEdge 1199  
setRepeaterMediaCasterDef 547  
setRepeaterOriginUrl 266, 352, 840  
setRepeaterQueryString 267  
setRequestFilters 511, 518  
setResetPlayStream 954  
setResponseCode 528  
setResyncAudioVideoOnSR 956  
setRetAMFNumber 864  
setRsoStorageDir 269  
setRTPAVSyncMethod 273  
setRTPDestination 974  
setRTPIIdleFrequency 268, 1323  
setRTPIgnoreProfileLevelId 975  
setRTPMaxRTCPWaitTime 273  
setRTPPlayAuthenticationMethod 272  
setRTPPublishAuthenticationMethod 272  
setRTPSession 307, 314, 899  
setRTPStream 1116  
setRTPWrapped 889  
setRTSP 963  
setRTSPAlwaysUseSDPPorts 973  
setRTSPBindIpAddress 288, 971  
setRTSPConnectionAddressType 288, 971  
setRTSPConnectionIpAddress 288, 971  
setRTSPMaximumPendingWriteBytes 286, 970  
setRTSPOriginAddressType 289, 972  
setRTSPOriginIpAddress 289, 972  
setRTSPPlayRangeStart 925  
setRTSPPlayRangeStop 925  
setRTSPPull 963  
setRTSPSessionDescription 968  
setRTSPSessionName 968  
setRTSPSessionTimeout 286, 970  
setRTSPTunnelingSessionId 925  
setSDPData 899  
setSDPLang 974  
setSendOnMetadata 1215, 1236, 1251  
setSendPlayStopLogEvent 1113  
setSendPublishStopLogEvent 1114  
setSendRecordStopLogEvent 1113  
setSendSDESEvents 968  
setSeq 217  
setServer 1354  
setServerIp 564, 920  
setServerPort 564, 920  
setSession 951  
setSessionId 558, 914  
setSessionProtocol 544, 563  
setSessionTimeout 560  
setSessionTracker 598  
setSessionType 562  
setSessionValid 916  
setSharedObjectReadAccess 271, 348  
setSharedObjectWriteAccess 272, 348  
setShutdownClient 355, 926  
setShutdownOnRelease 749  
setShuttingDown 1327  
setSize 204, 214, 786  
setSlotValue 1042, 1043  
setSlotVersion 1032  
setSoVersion 1032  
setSrc 205, 216, 862, 1085  
setSSLConfig 1287  
setSslFactoryClass 1285  
setStartLiveOnPreviousBufferTime 1252  
setStartLiveOnPreviousKeyFrame 1251

- setStartOnPreviousBufferTime 1239
- setStartOnPreviousKeyFrame 1239
- setStartRecordingOnStartup 1191
- setStartStartupStreams 1331
- setStartTC 1050
- setStartTCOffset 1050
- setStorageDir 1020, 1028, 1045
- setStream 559, 721
- setStreamAudioSampleAccess 270, 350
- setStreamDomainProtectionActive 548
- setStreamExt 567, 749, 958
- setStreamFileMapper 276
- setStreamKeyDir 268
- setStreamName 566, 958, 1176, 1207
- setStreamNameAliasProvider 278
- setStreamNameLogging 959
- setStreamPort 888
- setStreamPosition 568, 1067
- setStreamProperty 857
- setStreamQueryStr 964
- setStreamReadAccess 270, 350
- setStreamSessionId 959
- setStreamSessionIp 960
- setStreamSessionVersion 959
- setStreamStorageDir 268
- setStreamTimeout 723
- setStreamType 254, 337, 736, 838, 957, 1101, 1229
- setStreamTypeStr 547
- setStreamVersionHandler 484
- setStreamVideoSampleAccess 270, 349
- setStreamWriteAccess 271, 351
- setSuspended 1286
- setSwitchLog 1250
- setTargetEncoding 156
- setTextEncoding 665, 686, 701, 707, 710
- setThreadContext 357
- setTimecode 205, 216, 704, 865
- setTimecodes 216
- setTimeOffsetBetweenItems 1252
- setTimeoutSession 560
- setTimesInMilliseconds 1251
- setTimeStampFormat 701
- setTrack 893
- setTranscodeResult 1122
- setTranscoderName 1207
- setTransportMode 966
- setTTL 889, 896
- setType 125, 205, 215, 786, 865
- setUDPMangedDeliveryCount 975
- setUDPMangedDeliveryDelay 975
- setUri 564, 921
- setURL 689, 710
- setUserAgent 564, 915
- setUserAgents 985, 1002
- setUserHTTPHeader 572
- setUserQueryStr 570
- setValid 747
- setValidateFMLEConnections 290, 357
- setValidationFrequency 265, 1321
- setValue 686, 704, 707
- setVersion 1017, 1041
- setVersionFile 1161
- setVHost 306, 314, 543, 558, 915
- setVideoCodecId 785
- setVideoHost 888
- setVideoPort 887
- setVideoSize 1085
- setVideoTC 1088
- setVisited 1347
- setVODLastTimeTC 973
- setVODPlayLen 973
- setVODStartTimeTC 972
- setWaitForVideoKeyFrame 1161
- setWidth 786
- setWowzaAudioOnly 785
- SharedObject 1038, 1039
- SHARED\_OBJECT\_CMD\_CONNECT 1013
- SHARED\_OBJECT\_CMD\_CONNECT\_SUCCESS 1014
- SHARED\_OBJECT\_CMD\_DELETE 1014
- SHARED\_OBJECT\_CMD\_DISCONNECT 1013
- SHARED\_OBJECT\_CMD\_ERROR 1014
- SHARED\_OBJECT\_CMD\_SEND 1014
- SHARED\_OBJECT\_CMD\_SETVALUE 1013
- SHARED\_OBJECT\_STATUS\_CHANGE 1014
- SHARED\_OBJECT\_STATUS\_CLEAR 1014
- SHARED\_OBJECT\_STATUS\_DELETE 1014
- SHARED\_OBJECT\_STATUS\_SUCCESS 1014
- shortValue 164
- shouldDeleteArchivedStream 369, 502
- shouldDvrRecord 1194

shouldLoadArchivedStream 369, 502  
shouldLoadStream 473  
shouldLoadStreamVersion 474  
shouldStartRecordingOnStartup 1191  
shutdown 231, 252, 492, 532, 559, 586, 592, 598, 722, 748, 755,  
879, 894, 915, 954, 1087, 1144, 1187, 1197, 1207, 1313, 1354  
shutdownAppInstance 235  
shutdownApplication 1328  
shutdownClient 252, 336, 912  
shutdownRTSPSession 884  
shutdownSession 543  
shutdownStream 762  
sinfo 1167  
size 136, 146, 171, 191, 1016, 1026, 1039, 1112, 1147, 1289  
skipByte 126  
sloppyGetTrack 950  
splitCookie 71  
splitPragmas 70  
splitQueryStr 70  
stackTraceToString 42  
stampToString 95  
start 100, 998  
startApplicationInstance 1329  
startAudioData 1231  
startAudioPacket 1116  
startChunk 464, 577  
startCommandInterface 982, 998  
startDataData 1233  
startDataPacket 1116  
startMediaCasterStream 283  
startPlay 1147  
startPlayback 1066  
startPublishing 1110  
startRecording 485, 496, 1189  
startRTPPull 1273, 1274  
startServer 999  
startStartupStreams 1331  
startStream 752, 1187, 1198  
startsWith 37  
startVHost 983, 999  
startVHosts 983, 1000  
startVideoData 1231  
startVideoPacket 1116  
STAT\_connect\_application\_not\_available 648  
STAT\_connect\_application\_not\_found 649  
STAT\_connect\_bad\_gateway 649  
STAT\_connect\_internal\_error 649  
STAT\_connect\_license\_limit 649  
STAT\_connect\_pending\_wating 648  
STAT\_connect\_redirect 649  
STAT\_connect\_rejected\_by\_application 648  
STAT\_connect\_rejected\_by\_module 649  
STAT\_connect\_resource\_limit 649  
STAT\_connect\_service\_unavailable 649  
STAT\_connect\_successful 648  
STAT\_connect\_unknown\_protocol 648  
STAT\_general\_internal\_error 651  
STAT\_general\_successful 651  
STAT\_play\_bad\_request 650  
STAT\_play\_internal\_error 650  
STAT\_play\_rejected\_by\_application 650  
STAT\_play\_rejected\_by\_module 650  
STAT\_play\_stream\_not\_found 650  
STAT\_play\_successful 650  
STAT\_play\_unsupported\_media\_type 650  
STAT\_publish\_bad\_request 650  
STAT\_publish\_in\_use 651  
STAT\_publish\_internal\_error 651  
STAT\_publish\_rejected\_by\_application 651  
STAT\_publish\_successful 650  
STAT\_publish\_unsupported\_media\_type 651  
STAT\_stop\_client\_disconnect 651  
STAT\_stop\_successful 651  
stateChange 489  
statusCodeToStr 72  
stopAdminAgent 988, 998  
stopCommandInterface 982, 999  
stopMediaCasterStream 283  
stopName 1087, 1146  
stopPublishing 1110  
stopRecording 485, 496, 1190  
stopRTPPull 1272  
stopServer 998  
stopStartupStreams 1331  
stopStream 752  
stopVHost 983, 999  
stopVHosts 983, 999  
storeChunks 487, 497  
storeOnMetadata 487, 497  
Stream 1245

- streamCodecToString 55
- streamExists 752, 953
- STREAMINFO\_SESSIONATTRIBUTES 948
- STREAMINFO\_SESSIONBANDWIDTH 947
- STREAMINFO\_SESSIONCONNECTIONDATA 947
- STREAMINFO\_SESSIONEMAILADDRESS 947
- STREAMINFO\_SESSIONINFORMATION 947
- STREAMINFO\_SESSIONNAME 947
- STREAMINFO\_SESSIONPHONENUMBER 947
- STREAMINFO\_SESSIONPORIGIN 947
- STREAMINFO\_SESSIONPROTOCOLVERSION 946
- STREAMINFO\_SESSIONREPEATTIMES 948
- STREAMINFO\_SESSIONTIMEZONES 948
- STREAMINFO\_SESSIONTIMING 948
- STREAMINFO\_SESSIONURI 947
- StreamItem 1338
- streamLicenses 1172
- StreamList 1341
- streamLock 1172
- streamMonitorAudioStartTimeout 852
- streamMonitorAudioTCNegTolerance 851
- streamMonitorAudioTCPosTolerance 851
- streamMonitorAudioTCToleranceEnable 851
- streamMonitorAudioTimeout 852
- streamMonitorAVSyncTolerance 852
- streamMonitorAVSyncToleranceEnable 852
- streamMonitorDataTCNegTolerance 851
- streamMonitorDataTCPosTolerance 851
- streamMonitorDataTCToleranceEnable 851
- streamMonitorDebug 852
- streamMonitorResetNameGroups 852
- streamMonitorStreamStartTimeout 850
- streamMonitorStreamTimeout 850
- streamMonitorVideoStartTimeout 852
- streamMonitorVideoTCNegTolerance 851
- streamMonitorVideoTCPosTolerance 851
- streamMonitorVideoTCToleranceEnable 851
- streamMonitorVideoTimeout 852
- streamNames 1172
- streamNamesLock 1172
- streamNameToGroup 1172
- streamNameToValidFilename 47
- streamPacketizers 912
- streams 908, 1172
- streamsOrder 908
- streamSrcToMediaCaster 758
- STREAMTIMEOUTREASON\_GOOD 719
- STREAMTIMEOUTREASON\_MISSING 719
- STREAMTIMEOUTREASON\_NORTSPSESSION 719
- STREAMTIMEOUTREASON\_NOSESSION 719
- STREAMTIMEOUTREASON\_NOSTREAM 719
- STREAMTIMEOUTREASON\_NOTIMEOUT 719
- STREAMTIMEOUTREASON\_NOURL 719
- STREAMTIMEOUTREASON\_RECONNECTRUNNING 719
- STREAMTIMEOUTREASON\_UNKNOWN 718
- streamToFileForRead 1130
- streamToFileForWrite 1131
- streamToIndex 1176
- StreamUtils 1277
- StringUtils 93
- suspendAllHostPorts 1328
- suspendAllVHosts 987, 1000
- suspendCommandInterface 998
- suspendEncoding 27
- suspendServer 988, 1000
- switchName 1086, 1146
- switchPlay 1147
- switchSetupToMPEGTS 965
- SystemUtils 97
- T
- TAG\_APIC 674
- TAG\_COMM 681
- TAG\_LINK 674
- TAG\_PRIV 674
- TAG\_RBUF 674
- TAG\_SYLT 674
- TAG\_TALB 674
- TAG\_TBPM 675
- TAG\_TCOM 675
- TAG\_TCON 675
- TAG\_TCOP 675
- TAG\_TDEN 675
- TAG\_TDLY 675
- TAG\_TDOR 675
- TAG\_TDRC 675
- TAG\_TDRL 676
- TAG\_TDTG 676

TAG\_TENC 676  
TAG\_TEXT 676  
TAG\_TFLT 676  
TAG\_TIPL 676  
TAG\_TIT1 676  
TAG\_TIT2 676  
TAG\_TIT3 676  
TAG\_TKEY 677  
TAG\_TLAN 677  
TAG\_TLEN 677  
TAG\_TMCL 677  
TAG\_TMED 677  
TAG\_TMOO 677  
TAG\_TOAL 677  
TAG\_TOFN 677  
TAG\_TOLY 678  
TAG\_TOPE 678  
TAG\_TOWN 678  
TAG\_TPE1 678  
TAG\_TPE2 678  
TAG\_TPE3 678  
TAG\_TPE4 678  
TAG\_TPOS 678  
TAG\_TPRO 679  
TAG\_TPUB 679  
TAG\_TRCK 679  
TAG\_TRSN 679  
TAG\_TRSO 679  
TAG\_TSOA 679  
TAG\_TSOP 679  
TAG\_TSOT 679  
TAG\_TSRC 679  
TAG\_TSSE 680  
TAG\_TSST 680  
TAG\_TXXX 681  
TAG\_UNKN 674  
TAG\_WCOM 680  
TAG\_WCOP 680  
TAG\_WOAF 680  
TAG\_WOAR 680  
TAG\_WOAS 680  
TAG\_WORS 681  
TAG\_WPAY 681  
TAG\_WPUB 681  
TAG\_WXXX 681  
TAGS\_TEXTINFORMATION 680  
TEMP\_CONSOLE\_APPENDER\_NAME 603  
TEMP\_LIST\_APPENDER\_NAME 603  
terminate 1344  
testFlashVersion 353  
testNextByte 125  
TEXTENCODING\_ISO\_8859\_1 673  
TEXTENCODING\_UTF16 673  
TEXTENCODING\_UTF16BE 673  
TEXTENCODING\_UTF8 674  
ThreadPool 1343  
TIME\_MAP\_TYPE 451  
timeCreated 910  
timeReset 472  
TIMESTAMP\_RULE\_FORMAT 602  
TIMESTAMPFORMAT\_MILLISECONDS 700  
TIMESTAMPFORMAT\_MPEG 700  
toArray 147  
toByteBuffer 147  
toDvr 506  
toHex 41  
toLong 41  
toNTPTime 98  
toSMILString 777, 783, 790  
toString 143, 149, 165, 179, 183, 195, 205, 216, 302, 373, 375,  
388, 391, 394, 396, 739, 776, 783, 789, 886, 1222, 1285, 1340,  
1347  
totalIOPerformance2Last 910  
totalIOPerformanceLast 910  
touch 285, 336, 558, 915, 970, 1054, 1187, 1199  
touchApplicationInstance 1328, 1329  
toValidFilename 47  
trait 190  
transportFindBestMatch 969  
traverseDirectory 48  
triggerAMF3Switch 130  
trim 1105, 1167  
trimTrailingZero 682  
truncatePacket 214  
type 125, 381, 782  
  
U  
  
UDPAppender 613  
unbind 894



unbindAllHostPorts 1328  
unbindAllVHosts 988, 1000  
uncacheRTPStream 882  
unlock 1022, 1046  
unlockRepeaterStreams 953  
unpublish 1230  
unregisterCallback 1105  
unregisterLiveStreamTranscoder 1007  
unregisterOnPlayStatus 1106  
unregisterOnStatus 1106  
unregisterPlayer 722, 748, 759  
updateIOPerformance 1000  
updateLoggingDuration 1002, 1112, 1320  
updateLoggingValues 533, 563, 587, 593, 921, 1112, 1147  
updateOnCuePointTimecode 66  
updateSDPDestination 1272  
uri 911  
URL\_SAFE 16  
urlToId 103  
URLUtils 102  
userAgent 909  
userExists 313  
UTC\_TIME 445  
utcTime 381  
utcToDvr 506

## V

validateConnection 363  
validateLicense 366  
validateNewConnection 319  
validStreamDomainToString 569  
valueOf 95, 445  
values 445  
versionFile 48  
vhost 306, 908  
VHOST\_DEFAULT 1303  
VHostItem 1346  
VHostList 1350  
VHostSingleton 1353  
videoCodecId 782  
videoCodecStringToId 56  
videoCodecToMetaDataString 55  
videoCodecToString 56  
videoCodecTypeToString 90

videoPacketizers 912  
VIDEOSAMPLE\_ACCESS\_ALL 332  
VIDEOSAMPLE\_ACCESS\_NONE 333  
VIDEOSAMPLEACCESS 1083

## W

warn 623  
width 782  
WMSLogger 618  
WMSLoggerFactory 626  
WMSLoggerIDs 654  
WMSProperties 300  
wowzaAudioOnly 782  
wrap 147  
write 27, 865, 866  
WRITE\_ACCESS\_ALL 333  
WRITE\_ACCESS\_NONE 333  
WRITEACCESS 1038, 1083  
writeAppConfig 236  
writeAppInstConfig 282  
writeChunk 60  
writeCodecConfig 1274, 1275  
writeConfig 988, 1005  
writeDeleteError 1021, 1043  
writeDuration 63  
writeError 1043  
writeGeneratedKeyFrame 1065  
writeHeader 59  
writePackets 62, 63, 1064, 1065, 1160  
writeSetValueError 1020, 1043  
writeShortHeader 59  
writeString 157  
writeVHostConfig 1332  
writeXMLConfig 1006

## X

XMLUtils 106