



---

Wowza Streaming Engine™

# Server-Side API

---

# **Wowza Streaming Engine: Server-Side API**



---

## **Version 4.0**

---

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

---

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

#### **No Endorsement or Warranty for Third-Party Links and Software**

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

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

#### **Wowza Trademarks**

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

#### **Third-Party Trademarks and Copyrights**

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

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

---

## Document History

Version	Description	Release date
Doc v4.0.0	Initial document release for Wowza Streaming Engine 4.0	02-11-2014
Doc v4.0.1	General update for Wowza Streaming Engine 4.0.1	02-14-2014
Doc v4.0.2	General update for Wowza Streaming Engine 4.0.2	03-13-2014
Doc v4.0.3	General update for Wowza Streaming Engine 4.0.3	03-21-2014

### Note

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



---

Package

**com.wowza.io**

## com.wowza.io

### Interface IRandomAccessReader

public interface IRandomAccessReader  
extends

#### Field Summary

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

#### Method Summary

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

void	<a href="#"><u>open()</u></a> Open the media asset
int	<a href="#"><u>read</u>(byte[] buf, int off, int size)</a> Read bytes from the media asset
void	<a href="#"><u>seek</u>(long pos)</a> Seek to a position in the media asset
void	<a href="#"><u>setDirection</u>(int direction)</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)
```

Initialize 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(Object[] params)</a>
----------------------------------	--

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
--

---

### Constructors

#### AMFUtils

```
public AMFUtils()
```

### Methods

#### convertParams

```
public static AMFData\[\] convertParams(Object[] params)
```

(continued from last page)

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

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

**Parameters:**

params - Array of Java

**Returns:**

Array of AMFData[] objects

## com.wowza.util

### Class Base64

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

---

public class **Base64**  
extends Object

Encodes and decodes to and from Base64 notation.

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

Change Log:

- v2.2.1 - Fixed bug using URL\_SAFE and ORDERED encodings. Fixed bug when using very small files (~< 40 bytes).
- v2.2 - Added some helper methods for encoding/decoding directly from one file to the next. Also added a main() method to support command line encoding/decoding from one file to the next. Also added these Base64 dialects:
  1. The default is RFC3548 format.
  2. Calling Base64.setFormat(Base64.BASE64\_FORMAT.URLSAFE\_FORMAT) generates URL and file name friendly format as described in Section 4 of RFC3548. <http://www.faqs.org/rfcs/rfc3548.html>
  3. Calling Base64.setFormat(Base64.BASE64\_FORMAT.ORDERED\_FORMAT) generates URL and file name friendly format that preserves lexical ordering as described in <http://www.faqs.org/qa/rfcc-1940.html>
- Special thanks to Jim Kellerman at <http://www.powerset.com/> for contributing the new Base64 dialects.
- v2.1 - Cleaned up javadoc comments and unused variables and methods. Added some convenience methods for reading and writing to and from files.
- v2.0.2 - Now specifies UTF-8 encoding in places where the code fails on systems with other encodings (like EBCDIC).
- v2.0.1 - Fixed an error when decoding a single byte, that is, when the encoded data was a single byte.
- v2.0 - I got rid of methods that used booleans to set options. Now everything is more consolidated and cleaner. The code now detects when data that's being decoded is gzip-compressed and will decompress it automatically. Generally things are cleaner. You'll probably have to change some method calls that you were making to support the new options format (ints that you "OR" together).
- v1.5.1 - Fixed bug when decompressing and decoding to a byte[] using decode( String s, boolean gzipCompressed ). Added the ability to "suspend" encoding in the Output Stream so you can turn on and off the encoding if you need to embed base64 data in an otherwise "normal" stream (like an XML file).
- v1.5 - Output stream passes 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

public static final	<u><a href="#">DECODE</a></u> Specify decoding. Value: <b>0</b>
public static final	<u><a href="#">DONT_BREAK_LINES</a></u> Don't break lines when encoding (violates strict Base64 specification) Value: <b>8</b>
public static final	<u><a href="#">ENCODE</a></u> Specify encoding. Value: <b>1</b>
public static final	<u><a href="#">GZIP</a></u> Specify that data should be gzip-compressed. Value: <b>2</b>
public static final	<u><a href="#">NO_OPTIONS</a></u> No options specified. Value: <b>0</b>
public static final	<u><a href="#">ORDERED</a></u> 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>
public static final	<u><a href="#">URL_SAFE</a></u> 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

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

static Object	<a href="#"><code>decodeToObject</code></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 void	<a href="#"><code>encodeString</code></a> (String inString) Encodes inString and outputs result to stdout
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`

**Fields****NO\_OPTIONS**

`public static final int NO_OPTIONS`

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

**ENCODE**

`public static final int ENCODE`

Specify encoding.  
Constant value: **1**

## DECODE

```
public static final int DECODE
```

Specify decoding.  
Constant value: **0**

## GZIP

```
public static final int GZIP
```

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

## DONT\_BREAK\_LINES

```
public static final int DONT_BREAK_LINES
```

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

## URL\_SAFE

```
public static final int URL_SAFE
```

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

## ORDERED

```
public static final int ORDERED
```

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

## Methods

### main

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

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

### encodeObject

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

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

**Parameters:**

serializableObject - The object to encode

**Returns:**

The Base64-encoded object

## encodeObject

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

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

Valid options:

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

Example: encodeObject( myObj, Base64.GZIP ) or

Example: encodeObject( myObj, Base64.GZIP | Base64.DONT\_BREAK\_LINES )

**Parameters:**

serializableObject - The object to encode  
options - Specified options

**Returns:**

The Base64-encoded object

**See Also:**

[GZIP](#)  
[DONT\\_BREAK\\_LINES](#)

---

## encodeBytes

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

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

**Parameters:**

source - The data to convert

---

## encodeBytes

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

(continued from last page)

Encodes a byte array into Base64 notation.

Valid options:

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

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

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

**Parameters:**

`source` - The data to convert  
`options` - Specified options

**See Also:**

[GZIP](#)  
[DONT\\_BREAK\\_LINES](#)

## encodeBytes

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

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

**Parameters:**

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

## encodeBytes

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

Encodes a byte array into Base64 notation.

Valid options:

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

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

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

(continued from last page)

**Parameters:**

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

**See Also:**

[GZIP](#)  
[DONT\\_BREAK\\_LINES](#)

---

## decode

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

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

**Parameters:**

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

**Returns:**

decoded data

---

## decode

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

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

**Parameters:**

s - the string to decode

**Returns:**

the decoded data

---

## decode

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

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

**Parameters:**

s - the string to decode  
options - encode options such as URL\_SAFE

**Returns:**

the decoded data

---

## decodeToObject

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

(continued from last page)

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

**Parameters:**

encodedObject - The Base64 data to decode

**Returns:**

The decoded and deserialized object

---

**encodeToFile**

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

Convenience method for encoding data to a file.

**Parameters:**

dataToEncode - byte array of data to encode in base64 form  
filename - Filename for saving encoded data

**Returns:**

true if successful, false otherwise

---

**decodeToFile**

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

Convenience method for decoding data to a file.

**Parameters:**

dataDecode - Base64-encoded data as a string  
filename - Filename for saving decoded data

**Returns:**

true if successful, false otherwise

---

**decodeFromFile**

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

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

**Parameters:**

filename - Filename for reading encoded data

**Returns:**

decoded byte array or null if unsuccessful

---

**encodeFromFile**

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

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

**Parameters:**

filename - Filename for reading binary data

**Returns:**

base64-encoded string or null if unsuccessful

---

## encodeFileToFile

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

Reads infile and encodes it to outfile.

**Parameters:**

  infile - Input file  
  outfile - Output file

---

## encodeString

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

Encodes inString and outputs result to stdout

**Parameters:**

  inString - Input string

---

## decodeString

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

Decodes inString and outputs result to stdout

**Parameters:**

  inString - Input string

---

## decodeFileToFile

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

Reads infile and decodes it to outfile.

**Parameters:**

  infile - Input file  
  outfile - Output file

---

## com.wowza.util Class Base64.InputStream

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

All Implemented Interfaces:  
java.io.Closeable

public static class **Base64.InputStream**

extends java.io.FilterInputStream

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

See Also:

[Base64](#)

Fields inherited from class java.io.FilterInputStream

in

### Constructor Summary

public	<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(byte[] dest, int off, int len)</a> 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

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)
<i>Note: Technically, this makes your encoding non-compliant.</i>

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.ENCODE](#)  
[Base64.DECODE](#)  
[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 base640-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 base640-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(byte[] b)</a> Convert byte array to int
static int	<a href="#">byteArrayToInt(byte[] b, int offset)</a> Conver byte array to int with offset
static int	<a href="#">byteArrayToInt(byte[] b, int offset, int count)</a> Convert byte array to int with offset.
static int	<a href="#">byteArrayToInt(byte[] b, int offset, int count, boolean isReverse)</a> Convert byte array to int with offset.
static long	<a href="#">byteArrayToLong(byte[] b)</a> Convert byte array to long
static long	<a href="#">byteArrayToLong(byte[] b, int offset)</a> Conver byte array to long with offset
static long	<a href="#">byteArrayToLong(byte[] b, int offset, int count)</a> Convert byte array to long with offset.
static long	<a href="#">byteArrayToLong(byte[] b, int offset, int count, boolean isReverse)</a> Convert byte array to long with offset.
static int	<a href="#">byteArrayToShort(byte[] b)</a> Convert byte array to int

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

## 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)
```

Convert 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)
```

Convert byte array to int with offset

**Parameters:**

b - byte array (4 bytes)  
offset - offset

**Returns:**

int value

---

## byteArrayToInt

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

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

**Parameters:**

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

**Returns:**

int value

---

## byteArrayToInt

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

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

**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)
```

Convert 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 is in 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="#">formatBytesInSingleRow</a> (byte[] data, int offset, int len) 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="#"><u>formatSeconds</u></a> (long t) Give a time in seconds, return a String representing the time in hh:mm:ss.
static String	<a href="#"><u>formatUtcTime</u></a> (long utcTime)
static String	<a href="#"><u>stackTraceToSingleLineString</u></a> (Throwable e, int maxLines) Given an exception, return single line String representation.
static String	<a href="#"><u>stackTraceToString</u></a> (Throwable e) Given an exception, return String representation.
static String	<a href="#"><u>stackTraceToString</u></a> (Throwable e, int maxLines, String separator) Given an exception, return single line String representation.
static String	<a href="#"><u>toHex</u></a> (byte value) Format a byte value to a 0xff format
static String	<a href="#"><u>toHex</u></a> (int value) Format a byte value to a 0xffffffff format
static String	<a href="#"><u>toLong</u></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)
```

(continued from last page)

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

**Parameters:**

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

**Returns:**

formatted string

---

## formatBytes

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

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

**Parameters:**

data - byte array

**Returns:**

formatted string

---

## formatBytes

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

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

**Parameters:**

data - byte array

**Returns:**

formatted string

---

## formatBytes

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

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

**Parameters:**

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

**Returns:**

formatted string

---

## formatBytes

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

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

(continued from last page)

**Parameters:**

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

**Returns:**

formatted string

---

**formatBytesInSingleRow**

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

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

**Parameters:**

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

**Returns:**

formatted string

---

**formatBytesStruct**

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

---

**formatBytesStruct**

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

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

**Parameters:**

data - byte array

**Returns:**

formatted string

---

**toHex**

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

Format a byte value to a 0xff format

**Parameters:**

value - byte value

**Returns:**

return string

---

**toHex**

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

(continued from last page)

Format a byte value to a 0xffffffff format

**Parameters:**

value - int value

**Returns:**

return string

---

**toLong**

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

Format a long value to a 0xfffffffffffffff format

**Parameters:**

value - long value

**Returns:**

return string

---

**byteArrayToFile**

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

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

**Parameters:**

data - a buffer of bytes  
filePath - The path and filename

---

**formatSeconds**

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

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

123 s --> 00:02:03

**Parameters:**

t

**Returns:**

time representation

---

**formatMilliseconds**

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

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

123456 ms --> 00:02:03.456

**Parameters:**

ms

**Returns:**

time representation

## formatUtcTime

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

---

## stackTraceToString

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

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

**Parameters:**

e - throwable

**Returns:**

stack trace representation

---

## stackTraceToSingleLineString

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

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

**Parameters:**

e - throwable

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

**Returns:**

stack trace info

---

## stackTraceToString

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

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

**Parameters:**

e - throwable

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

separator - character to replace line endings with

**Returns:**

stack trace info

---

## doesStackContainMethod

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

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

### 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="#">closeQuietly</a> (java.io.InputStream stream)
static void	<a href="#">closeQuietly</a> (java.io.OutputStream stream)
static void	<a href="#">closeQuietly</a> (java.io.Reader reader)
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 fromFile, java.io.File toFile)
static boolean	<a href="#">deleteDirectory</a> (java.io.File path)
static void	<a href="#">deleteFolder</a> (java.io.File folder)
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
--

## Constructors

### FileUtils

```
public FileUtils()
```

## Methods

### fileToByteArray

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

---

### streamNameToValidFilename

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

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

**Parameters:**

name - stream name

**Returns:**

encoded filename

---

### toValidFilename

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

Encode a name to a valid filename.

**Parameters:**

name

**Returns:**

encoded filename

---

### copyFile

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

Simple file copy routine

**Parameters:**

fromFile - from file

toFile - to file

---

(continued from last page)

## copyFile2

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

---

## closeQuietly

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

---

## closeQuietly

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

---

## closeQuietly

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

---

## deleteDirectory

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

---

## versionFile

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

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

**Parameters:**

newFile - input file

**Returns:**

renamed file

---

## traverseDirectory

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

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

**Parameters:**

dir - starting directory

fileNotify - file notify object

---

## deleteFolder

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

(continued from last page)

## com.wowza.util Class FLVUtils

```
java.lang.Object
+-com.wowza.util.FLVUtils
```

---

public final class **FLVUtils**  
extends Object

FLVUtils: utility for reading and writing .flv files.

### Field Summary

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

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

static void	<a href="#"><u>writeHeaderFlags</u></a> (java.io.File file, byte newFlags)
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="#"><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, IFLVWriterAdjustTimecode dataPacketTimecodeAdjuster) Write a bunch of packets to .flv file all at once.
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, long[] currentTCs) Write a bunch of packets to .flv file all at once.
static void	<a href="#"><u>writePackets</u></a> (java.io.OutputStream ds, java.util.List packetList, long tcOffset) Write audio/video/data packets to an .flv file.
static void	<a href="#"><u>writeShortHeader</u></a> (java.io.DataOutput ds)
static void	<a href="#"><u>writeShortHeader</u></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\_ITIMECODE**

public static final int **FLV\_CHUNKHEADER\_ITIMECODE**

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

## FLV\_CHUNKHEADER\_FIRSTBYTE

```
public static final int FLV_CHUNKHEADER_FIRSTBYTE
```

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

---

## FLV\_CHUNKHEADER\_SECONDBYTE

```
public static final int FLV_CHUNKHEADER_SECONDBYTE
```

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

---

## FLV\_CHUNKHEADER\_HEADERSIZE

```
public static final int FLV_CHUNKHEADER_HEADERSIZE
```

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

---

## FLV\_CHUNKHEADER\_BUFFERSIZE

```
public static final int FLV_CHUNKHEADER_BUFFERSIZE
```

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

---

## FLV\_CHUNKHEADER\_VALUESIZE

```
public static final int FLV_CHUNKHEADER_VALUESIZE
```

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

---

## FLV\_UFRAME

```
public static final int FLV_UFRAME
```

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

---

## FLV\_KFRAME

```
public static final int FLV_KFRAME
```

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

---

## FLV\_DFRAME

```
public static final int FLV_DFRAME
```

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

---

## FLV\_PFRAME

```
public static final int FLV_PFRAME
```

(continued from last page)

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

---

## FLV\_TCINDEXAUDIO

```
public static final int FLV_TCINDEXAUDIO
```

Constant value: **0**

---

## FLV\_TCINDEXVIDEO

```
public static final int FLV_TCINDEXVIDEO
```

Constant value: **1**

---

## FLV\_TCINDEXDATA

```
public static final int FLV_TCINDEXDATA
```

Constant value: **2**

## Constructors

### FLVUtils

```
public FLVUtils()
```

## Methods

### streamCodecToString

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

Get a printable string representation of the stream codecs defined as IVHost.CODEC\_STREAM\_\*

**Parameters:**

codec - codec id defined IVHost.CODEC\_STREAM\_\*

**Returns:**

codec name

---

### audioCodecStringToId

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

Parse a string to get the codec ID defined by IVHost.CODEC\_AUDIO\_\*

**Parameters:**

codecString - codec string

**Returns:**

codec ID

## **videoCodecToMetaDataString**

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

Get a printable string representation of the video codecs defined as IVHost.CODEC\_VIDEO\_\* as the string used in the onMetaData event

**Parameters:**

codec - codec id defined IVHost.CODEC\_VIDEO\_\*

**Returns:**

codec name

---

## **audioCodecToMetaDataString**

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

Get a printable string representation of the audio codecs defined as IVHost.CODEC\_AUDIO\_\* as the string used in the onMetaData event

**Parameters:**

codec - codec id defined IVHost.CODEC\_AUDIO\_\*

**Returns:**

codec name

---

## **audioCodecToString**

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

Get a printable string representation of the audio codecs defined as IVHost.CODEC\_AUDIO\_\*

**Parameters:**

codec - codec id defined IVHost.CODEC\_AUDIO\_\*

**Returns:**

codec name

---

## **videoCodecStringToInt**

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

Parse a string to get the codec ID defined by IVHost.CODEC\_VIDEO\_\*

**Parameters:**

codecString - codec string

**Returns:**

codec ID

---

## **videoCodecToString**

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

Get a printable string representation of the video codecs defined as IVHost.CODEC\_VIDEO\_\*

**Parameters:**

codec - codec id defined IVHost.CODEC\_VIDEO\_\*

(continued from last page)

**Returns:**

codec name

---

**frameTypeToString**

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

---

**getFrameType**

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

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

**Parameters:**

values - header values

**Returns:**

type of video frame (FLV\_\*FRAME)

---

**getFrameType**

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

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

**Parameters:**

value - first byte of packet

**Returns:**

type of video frame (FLV\_\*FRAME)

---

**getAudioCodec**

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

Return the codec portion of the first byte of an audio packet. Return should be one of IVHost.CODEC\_AUDIO\_\*

**Parameters:**

value - first byte of audio packet

**Returns:**

codec id

---

**getVideoCodec**

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

Return the codec portion of the first byte of an video packet. Return should be one of IVHost.CODEC\_VIDEO\_\*

**Parameters:**

value - first byte of audio packet

**Returns:**

codec id

## getFrameType

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

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

**Parameters:**

value - first byte of packet

**Returns:**

type of video frame (FLV\_\*FRAME)

---

## readChunk

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

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

**Parameters:**

is - InputStream

**Returns:**

AMFPacket of data or null if unsuccessful or end of file

---

## readChunk

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

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

**Parameters:**

is - InputStream

**Returns:**

AMFPacket of data or null if unsuccessful or end of file

---

## readPrevChunkHeader

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

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

**Parameters:**

is - RandomAccessFile

buffer - temporary buffer byte[FLV\_CHUNKHEADER\_BUFFERSIZE]

values - header values long[FLV\_CHUNKHEADER\_VALUESIZE]

---

## readChunkHeader

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

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

(continued from last page)

**Parameters:**

is - RandomAccessFile  
buffer - temporary buffer byte[FLV\_CHUNKHEADER\_BUFFERSIZE]  
values - header values long[FLV\_CHUNKHEADER\_VALUESIZE]

**readHeader**

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

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

**readHeader**

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

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

**writeShortHeader**

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

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

**Parameters:**

ds - OutputStream

**writeShortHeader**

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

**writeHeader**

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

Write file header including onMetaData packet.

**Parameters:**

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

**writeHeader**

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

(continued from last page)

Write file header including onMetaData packet.

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

```
Map extraMetadata = new HashMap();

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

#### **Parameters:**

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

## **writeChunk**

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

Write a packets worth of data.

#### **Parameters:**

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

## **writeChunk**

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

(continued from last page)

## interleavePackets

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

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

**Parameters:**

- audioPackets - list of audio packets (ByteBuffer)
- videoPackets - list of video packets (ByteBuffer)
- dataPackets - list of data packets (ByteBuffer)
- audioTCs - list of relative audio timecodes (Long)
- videoTCs - list of relative video timecodes (Long)
- dataTCs - list of relative data timecodes (Long)
- currentTCs - array of longs containing last TCs written for each packet type FLV\_TCINDEX\* (long[3]). Last timecode written for each packet type wil 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 wil 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 wil 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 wil 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 wil be returned
- `dataPacketTimecodeAdjuster` - class that implements the IFLVWriterAdjustTimecode interface for adjusting timecodes in this same array.

## adjustFirstPacketTCs

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

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

**Parameters:**

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

**Returns:**

lowest of three absolute timecodes

## writeDuration

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

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

**Parameters:**

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

## getLastTC

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

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

**Parameters:**

file

**Returns:**

duration (milliseconds)

---

## isVideoKeyFrame

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

Returns true if the packet is a video key frame

**Parameters:**

packet - packet

**Returns:**

true if is video key frame

---

## isVideoCodecConfig

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

Returns true if the packet is a video codec config packet

**Parameters:**

firstByte - first byte of packet

secondByte - second byte of packet

**Returns:**

true if video codec

---

## isVideoCodecConfig

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

Returns true if the packet is a video codec config packet

**Parameters:**

packet

**Returns:**

true if the packet is a video codec config packet

---

## isAudioCodecConfig

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

Returns true if the packet is a video codec config packet

**Parameters:**

(continued from last page)

packet

**Returns:**

true if the packet is a video codec config packet

---

## getAudioMP3Layer

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

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

**Parameters:**

packet - amf packet

**Returns:**

layer number

---

## getAudioCodec

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

Get the codec id for this audio packet. Returns IVHost.CODEC\_AUDIO\_UNKNOWN is unknown or not audio packet

**Parameters:**

packet - packet

**Returns:**

codec id IVHost.CODEC\_AUDIO\_\*

---

## getVideoCodec

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

Get the codec id for this video packet. Returns IVHost.CODEC\_VIDEO\_UNKNOWN is unknown or not video packet

**Parameters:**

packet - packet

**Returns:**

codec id IVHost.CODEC\_VIDEO\_\*

---

## getVideoFrameType

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

---

## getVideoTimecodeOffset

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

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

**Parameters:**

packet - AMFPacket

**Returns:**

timecode offset in milliseconds (can be negative)

## getVideoTimecodeOffset

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

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

**Parameters:**

buffer - video packet buffer

**Returns:**

timecode offset in milliseconds (can be negative)

---

## isVideoKeyFrame

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

Returns true if the packet is a video key frame

**Parameters:**

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

**Returns:**

true if is video key frame

---

## isVideoKeyFrame

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

Returns true if the packet is a video key frame

**Parameters:**

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

**Returns:**

true if is video key frame

---

## isVideoKeyFrame

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

Returns true if the packet is a video key frame

**Parameters:**

chunkHeaderValues - chunk header values returned by FLVUtils.readChunkHeader

**Returns:**

true if is video key frame

---

## updateOnCuePointTimecode

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

---

(continued from last page)

## updateOnCuePointTimecode

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

---

## isOnMetadataPacket

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

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

**Parameters:**

packet - packet

**Returns:**

true if onMetaData packet

---

## getOnMetadataData

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

---

## getHeaderFlags

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

---

## writeHeaderFlags

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

## com.wowza.util Class HTTPUtils

```
java.lang.Object
+-com.wowza.util.HTTPUtils
```

---

```
public class HTTPUtils
extends Object
```

HTTPUtils: utility class for making http requests.

### Field Summary

public static final	<a href="#">COOKIEDATEFORMAT</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	<a href="#"><u>statusCodeToStr(int statusCode)</u></a>
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,
```

**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="#"><b>getByteAllocation</b>(long request)</a>
	Requests bytes from the bandwidth throttler interface.

### Methods

#### **getByteAllocation**

public long **getByteAllocation**(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="#"><u>onFile</u></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

<code>public</code>	<a href="#"><u>IOPerformanceCounter()</u></a>
	Create an empty performance counter.

### Method Summary

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

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

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

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

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `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 a1Hash, 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 a1Hash, 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,
```

## 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 a1Hash,
    String realm,
    String nonce,
    String qop,
    String nonceCount,
    String cnonce)
```

Generate an HTTP authorization response

**Parameters:**

method - method  
uri - URI  
a1Hash - a1Hash  
realm - realm  
nonce - nonce  
qop - qop  
nonceCount - nonceCount  
cnonce - cnonce

**Returns:**

auth

---

## generateAuth

```
public static String generateAuth(String a2Hash,
    String a1Hash,
    String realm,
    String nonce,
    String qop,
    String nonceCount,
    String cnonce)
```

Generate an HTTP authorization response

**Parameters:**

a2Hash - a2Hash  
a1Hash - a1Hash  
realm - realm  
nonce - nonce  
qop - qop  
nonceCount - nonceCount  
cnonce - cnonce

**Returns:**

auth

---

## generateHashBytes

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

Generate MD5 hash

**Parameters:**

value - byte array to hash

**Returns:**

byte array result

## generateHashBytes

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

Generate MD5 hash

**Parameters:**

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

**Returns:**

byte array result

---

## generateHash

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

Generate MD5 hash

**Parameters:**

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

**Returns:**

hash as binhex string

## com.wowza.util Class MediaUtils

```
java.lang.Object
+-com.wowza.util.MediaUtils
```

---

```
public class MediaUtils
extends Object
```

MediaUtils: Media utilities.

### Field Summary

public static final	<a href="#">CODECSTR_FORMAT_VIDEO_POSTIOS4</a>
	Value: <b>2</b>
public static final	<a href="#">CODECSTR_FORMAT_VIDEO_PREIOS4</a>
	Value: <b>1</b>
public static final	<a href="#">CODECSTR_FORMAT_VIDEO_UNKNOWN</a>
	Value: <b>0</b>

### Constructor Summary

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

### Method Summary

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

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

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

### Fields

#### **CODECSTR\_FORMAT\_VIDEO\_UNKNOWN**

```
public static final int CODECSTR_FORMAT_VIDEO_UNKNOWN
```

Constant value: **0**

## CODECSTR\_FORMAT\_VIDEO\_PREIOS4

```
public static final int CODECSTR_FORMAT_VIDEO_PREIOS4
```

Constant value: 1

## CODECSTR\_FORMAT\_VIDEO\_POSTIOS4

```
public static final int CODECSTR_FORMAT_VIDEO_POSTIOS4
```

Constant value: 2

## Constructors

### MediaUtils

```
public MediaUtils()
```

## Methods

### audioCodecTypeToString

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

Audio codec ID to string. See IVHost.CODEC\_AUDIO\_\*

**Parameters:**

codecType - codec id. See IVHost.CODEC\_AUDIO\_\*

**Returns:**

codec string

---

### videoCodecTypeToString

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

Video codec ID to string. See IVHost.CODEC\_VIDEO\_\*

**Parameters:**

codecType - codec id. See IVHost.CODEC\_VIDEO\_\*

**Returns:**

codec string

## com.wowza.util Class NetworkUtils

```
java.lang.Object
+-com.wowza.util.NetworkUtils
```

---

```
public class NetworkUtils
extends Object
```

NetworkUtils: Networking utilities.

### Constructor Summary

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

### Method Summary

static boolean	<a href="#">isAddressMulticast(String IpAddress)</a> Returns true if IP address is multicast address
static String	<a href="#">socketAddressToUniqueStr(java.net.InetSocketAddress socketAddress)</a>

Methods inherited from class java.lang.Object

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

### Constructors

#### NetworkUtils

```
public NetworkUtils()
```

### Methods

#### isAddressMulticast

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

Returns true if IP address is multicast address

**Parameters:**

IpAddress - IP address

**Returns:**

true if IP address is multicast address

---

## socketAddressToUniqueStr

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

## com.wowza.util Interface NoMBean

```
public interface NoMBean  
extends Annotation
```

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

```
import com.wowza.util.NoMBean;  
import com.wowza.wms.module.*;  
  
class MyClass extends ModuleBase  
{  
    @NoMBean public void myMethod()  
    {  
    }  
}
```

Methods inherited from interface java.lang.annotation.Annotation

```
annotationType, equals, hashCode, toString
```

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

---

static String	<a href="#"><u>valueOf(String s)</u></a>
	Returns string, handling null String case as "".

---

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

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

---

## Constructors

**StringUtils**

```
public StringUtils()
```

## Methods

**intToStr**

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

**intToHexStr**

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

**longToStr**

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

**longToHexStr**

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

**isEmpty**

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

Checks if string is empty, handling null String case.

**Parameters:**

s

**Returns:**

(continued from last page)

false is string is null or "", true otherwise

---

## length

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

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

**Parameters:**

s

**Returns:**

0 if string is null, or length of string.

---

## valueOf

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

Returns string, handling null String case as "".

**Parameters:**

s

**Returns:**

"" if string is null, or string value.

---

## stampToString

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

Convert a duration (milliseconds) to a formatted string.

**Parameters:**

stamp - duration (milliseconds)

**Returns:**

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

---

## stampToStringNoUnits

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

Convert a duration (milliseconds) to a formatted string.

**Parameters:**

stamp - duration (milliseconds)

**Returns:**

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

---

## stringToMilliseconds

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

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

**Parameters:**

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

(continued from last page)

**Returns:**

milliseconds, -1 is parse failed

**equals**

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

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

**Parameters:**

s1 - string one  
s2 - string 2

**Returns:**

true is the strings are equal

**toStringList**

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

Convert properties to comma-separated string

**Parameters:**

properties

**Returns:**

comma-separated string

**toStringList**

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

Convert list of Strings to comma-separated string

**Parameters:**

strings

**Returns:**

comma-separated string

**toStringList**

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

Convert array of Strings to comma-separated string

**Parameters:**

strings

**Returns:**

comma-separated string

## com.wowza.util Class SystemUtils

```
java.lang.Object
+-com.wowza.util.SystemUtils
```

---

```
public class SystemUtils
extends Object
```

### Nested Class Summary

class	<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="#">msb1baseTime</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(String inValue)</a> Expand system level Java properties in a String in the form \${property-name}
static String	<a href="#">expandEnvironmentVariables(String inValue, java.util.Map valueMap)</a> 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="#"><u>getSystemTime()</u></a> Get system time in nanoseconds.
static long	<a href="#"><u>getUserTime()</u></a> Get user time in nanoseconds.
static long	<a href="#"><u>toNTPTime(long t)</u></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`

## 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(int start, int stop, String newValue)</a>
--------	---

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
--

### 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 function 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 String	<a href="#">getVersion</a> (org.w3c.dom.Element root) Given a Wowza XML file's root element, parse the version attribute
static boolean	<a href="#">getXMLPropertyBool</a> (javax.xml.xpath.XPath xpath, String xpathStr, org.w3c.dom.Element root, boolean defaultValue) Get a boolean int value pointed to by xpath or defaultValue if not found.
static double	<a href="#">getXMLPropertyDouble</a> (javax.xml.xpath.XPath xpath, String xpathStr, org.w3c.dom.Element root, double defaultValue) Get a single double value pointed to by xpath or defaultValue 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 defaultValue) Get a single int value pointed to by xpath or defaultValue if not found.
static int	<a href="#">getXMLPropertyIntSize</a> (javax.xml.xpath.XPath xpath, String xpathStr, org.w3c.dom.Element root, int defaultValue)
static long	<a href="#">getXMLPropertyLong</a> (javax.xml.xpath.XPath xpath, String xpathStr, org.w3c.dom.Element root, long defaultValue) Get a single long value pointed to by xpath or defaultValue if not found.
static long	<a href="#">getXMLPropertyLongSize</a> (javax.xml.xpath.XPath xpath, String xpathStr, org.w3c.dom.Element root, long defaultValue)

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 defaultValue) Get a single string value pointed to by xpath or defaultValue 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.XPathFactory	<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

## 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> element 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 defaultValue)
```

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

**Parameters:**

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

**Returns:**

single string value pointed to by xpath or defaultValue 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 defaultValue)
```

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

**Parameters:**

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

**Returns:**

int value or defaultValue if not found

---

## getXMLPropertyIntSize

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

---

## getXMLPropertyLong

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

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

(continued from last page)

**Parameters:**

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

**Returns:**

long value or defaultValue if not found

---

**getXMLPropertyLongSize**

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

---

**getXMLPropertyDouble**

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

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

**Parameters:**

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

**Returns:**

double value or defaultValue if not found

---

**getXMLPropertyBool**

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

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

**Parameters:**

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

**Returns:**

boolean value or defaultValue if not found

---

**newXPathFactory**

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

(continued from last page)

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

**Returns:**

XPathFactory or null if not found

---

## getVersion

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

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

**Parameters:**

root

**Returns:**

version or "0" if not specified

---

Package

**com.wowza.wms.amf**

## com.wowza.wms.amf

### Class AMF3Utils

```
java.lang.Object
+-com.wowza.wms.amf.AMF3Utils
```

---

```
public class AMF3Utils
extends Object
```

AMF3 utilities

### Constructor Summary

public	<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
--

## Constructors

### AMF3Utils

```
public AMF3Utils()
```

## Methods

### deserializeDate

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

Deserialize date type

**Parameters:**

data - buffer

**Returns:**

date

---

### serializeDate

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

Serialize a date object

**Parameters:**

out - output stream  
date - date

**Returns:**

number of bytes written

---

### deserializeInt

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

Deserialize int

**Parameters:**

data - buffer

**Returns:**

int value

---

### serializeInt

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

Serialize int value

**Parameters:**

(continued from last page)

out - output stream  
val - int value

**Returns:**

number of bytes written

---

## deserializeString

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

Deserialize string

**Parameters:**

data - data  
context - AMF context

**Returns:**

string value

**Throws:**

IOException

---

## deserializeString

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

Deserialize string

**Parameters:**

data - data

**Returns:**

string value

**Throws:**

IOException

---

## deserializeString

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

Deserialize string

**Parameters:**

data - data  
utflen - len of the string

**Returns:**

string value

**Throws:**

IOException

(continued from last page)

## serializeZeroLengthString

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

Serialize empty string

**Parameters:**

out - output stream

---

## serializeStringNoLength

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

Serialize string but do not write the length

**Parameters:**

out - output stream

str - string value

**Returns:**

number of bytes written

**Throws:**

IOException

---

## serializeString

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

Serialize a string value

**Parameters:**

out - output stream

str - string value

**Returns:**

number of bytes written

**Throws:**

IOException

## com.wowza.wms.amf Class AMFData

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

### Direct Known Subclasses:

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

public abstract class **AMFData**

extends Object

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

### Field Summary

public static final	<a href="#">AMF_LEVEL0</a> Value: 0
public static final	<a href="#">AMF_LEVEL3</a> Value: 3
public static final	<a href="#">DATA_TYPE_AMF3</a> Value: 17
public static final	<a href="#">DATA_TYPE_AMF3_ARRAY</a> Value: 9
public static final	<a href="#">DATA_TYPE_AMF3_BOOLEAN_FALSE</a> Value: 2
public static final	<a href="#">DATA_TYPE_AMF3_BOOLEAN_TRUE</a> Value: 3
public static final	<a href="#">DATA_TYPE_AMF3_BYTEARRAY</a> Value: 12
public static final	<a href="#">DATA_TYPE_AMF3_DATE</a> Value: 8
public static final	<a href="#">DATA_TYPE_AMF3_INTEGER</a> Value: 4
public static final	<a href="#">DATA_TYPE_AMF3_NULL</a> Value: 1

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

## Constructor Summary

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

## Method Summary

static <a href="#">AMFDataContextDeserializ</a> e	<a href="#">createContext Deserialize()</a> Create an AMF3 deserialization context
---	---

static <a href="#">AMFDataContextDeserialize</a>	<a href="#">createContextDeserialize(int objectEncoding)</a> 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(int objectEncoding)</a> Create an AMF3 serialization context
abstract void	<a href="#">deserialize(java.nio.ByteBuffer data)</a> Deserialize data in byte buffer
abstract void	<a href="#">deserialize(java.nio.ByteBuffer data, AMFDataContextDeserialize context)</a> Deserialize data in byte buffer
static <a href="#">AMFData</a>	<a href="#">deserializeInnerObject(java.nio.ByteBuffer data, AMFDataContextDeserialize context)</a> Deserialize next item
static <a href="#">AMFData</a>	<a href="#">getReference(java.nio.ByteBuffer data, AMFDataContextDeserialize context)</a> 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(java.nio.ByteBuffer data, AMFDataContextDeserialize context)</a> Returns true if next byte in data is AMF3 start
static boolean	<a href="#">isArrayStart(java.nio.ByteBuffer data, AMFDataContextDeserialize context)</a> Returns true if next byte in data is an array start
static boolean	<a href="#">isByteArrayStart(java.nio.ByteBuffer data, AMFDataContextDeserialize context)</a> Returns true if next byte is ByteArray start
static boolean	<a href="#">isMixedArrayStart(java.nio.ByteBuffer data, AMFDataContextDeserialize context)</a> Returns true if next byte in data is mixed array start
static boolean	<a href="#">isObjEnd(java.nio.ByteBuffer data, AMFDataContextDeserialize context)</a> Returns true if next byte is object end
static boolean	<a href="#">isObjStart(java.nio.ByteBuffer data, AMFDataContextDeserialize context)</a> Returns true if next byte in data is object start
static int	<a href="#">peekByte(java.nio.ByteBuffer data)</a> 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(AMFDataContextSerialize context)</a> 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

## 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,  
                                AMFDataContext Deserialize 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,  
                                 AMFDataContext Deserialize 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,  
                                 AMFDataContext Deserialize 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,  
    AMFDataContext Deserialize 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 AMFDataContext Serialize createContextSerialize()
```

Create an AMF3 serialization context

**Returns:**

AMF3 serialization context

---

## createContextSerialize

```
public static AMFDataContext Serialize createContextSerialize(int objectEncoding)
```

Create an AMF3 serialization context

**Parameters:**

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

**Returns:**

AMF3 serialization context

---

## createContextDeserialize

```
public static AMFDataContext Deserialize createContextDeserialize()
```

Create an AMF3 deserialization context

**Returns:**

AMF3 deserialization context

---

## createContextDeserialize

```
public static AMFDataContext Deserialize 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 AMFdataArray. 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 <a href="#">com.wowza.wms.amf.AMFData</a>
---

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

## Method Summary

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

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

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

**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 offest 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,
                    AMFDataContext Deserialize context)
```

## Methods

### remove

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

Remove an item from the array

**Parameters:**

index - index

**Returns:**

delete item or null if not found

---

**size**

```
public int size()
```

Returns the number of items in array

**Returns:**

number of items in array

---

**add**

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

Append a new item onto the array

**Parameters:**

data - AMFData object

---

**add**

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

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

**Parameters:**

data - string value

---

**add**

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

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

**Parameters:**

data - double value

---

**add**

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

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

**Parameters:**

data - int value

---

**add**

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

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

**Parameters:**

data - long value

---

**add**

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

(continued from last page)

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

**Parameters:**

data - date value

---

**add**

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

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

**Parameters:**

data - boolean value

---

**add**

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

Insert an item into the array

**Parameters:**

index - index  
data - AMFData object

---

**add**

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

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

**Parameters:**

index - index  
data - string value

---

**add**

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

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

**Parameters:**

index - index  
data - double value

---

**add**

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

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

**Parameters:**

index - index  
data - int value

---

**add**

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

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

**Parameters:**

index - index  
data - long value

---

**add**

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

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

**Parameters:**

index - index  
data - date value

---

**add**

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

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

**Parameters:**

index - index  
data - boolean value

---

**set**

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

Set an array item

**Parameters:**

index - index  
data - AMFData object

---

**set**

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

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

**Parameters:**

index - index  
data - string value

---

**set**

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

(continued from last page)

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

**Parameters:**

index - index  
data - double value

---

**set**

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

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

**Parameters:**

index - index  
data - int value

---

**set**

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

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

**Parameters:**

index - index  
data - long value

---

**set**

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

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

**Parameters:**

index - index  
data - date value

---

**set**

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

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

**Parameters:**

index - index  
data - boolean value

---

**get**

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

Get item at index

**Parameters:**

index

(continued from last page)

**Returns:**

Returns AMFData object or null if out of bounds

**getString**

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

Get item at index return as String

**Parameters:**

index

**Returns:**

Return item as String or null if out of bounds

**getInt**

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

Get item at index return as int

**Parameters:**

index

**Returns:**

Return item as int or 0 if out of bounds

**getLong**

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

Get item at index return as long

**Parameters:**

index

**Returns:**

Return item as long or 0 if out of bounds

**getShort**

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

Get item at index return as short

**Parameters:**

index

**Returns:**

Return item as short or 0 if out of bounds

**getByte**

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

Get item at index return as byte

**Parameters:**

(continued from last page)

**index****Returns:**

Return item as byte or 0 if out of bounds

---

**getBoolean****public boolean getBoolean(int index)**

Get item at index return as boolean

**Parameters:**

index

**Returns:**

Return item as boolean or false if out of bounds

---

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

Get item at index return as Date

**Parameters:**

index

**Returns:**

Return item as Date or null if out of bounds

---

**getObject****public AMFDataObj getObject(int index)**

Get item at index return as AMFDataObj

**Parameters:**

index

**Returns:**

Return item as AMFDataObj or null if out of bounds

---

**getDouble****public double getDouble(int index)**

Get item at index return as double

**Parameters:**

index

**Returns:**

Return item as double or 0 if out of bounds

---

**getFloat****public float getFloat(int index)**

Get item at index return as float

(continued from last page)

**Parameters:**

index

**Returns:**

Return item as float or 0 if out of bounds

---

**deserialize**

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

Deserialize data in byte buffer

---

**deserialize**

```
public void deserialize(java.nio.ByteBuffer data,  
                      AMFDataContext Deserialize 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,  
                      AMFDataContext Serialize 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(AMFDataContext Serialize 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
+- com.wowza.wms.amf.AMFData
  +- com.wowza.wms.amf.AMFDataByteArray
```

```
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](#), [serialize](#), [setType](#), [skipByte](#), [testNextByte](#), [triggerAMF3Switch](#)

#### Methods inherited from class [java.lang.Object](#)

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [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,  
                      AMFDataContext Deserialize 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,  
                      AMFDataContext Serialize 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(AMFDataContext Serialize 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(AMFDataTask 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="#">AMFDataTask</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="#"><u>isIntData()</u></a> Internal use, get int data
void	<a href="#"><u>setIntData(int intData)</u></a> Internal use, set int data
void	<a href="#"><u>setObjectEncoding(int objectEncoding)</u></a> Set object encoding, see AMFData.AMF_LEVEL*

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

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, 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> Contructor 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(AMFDataTask 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)
```

Contructor 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?
    amfDataDouble.doubleValue(): 0.0;

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

## Method Summary

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

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

---

```
createContextDeserialize, createContextSerialize, 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,
```

**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,  
                    AMFDataContext Deserialize 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
+- com.wowza.wms.amf.AMFData
  +- com.wowza.wms.amf.AMFDataList
```

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

## Method Summary

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

void	<a href="#"><u>add(boolean data)</u></a> Append a boolean (will be wrapped in an AMFDataItem object)
void	<a href="#"><u>add(java.util.Date data)</u></a> Append a date (will be wrapped in an AMFDataItem object)
void	<a href="#"><u>add(double data)</u></a> Append a double (will be wrapped in an AMFDataItem object)
void	<a href="#"><u>add(int data)</u></a> Append a int (will be wrapped in an AMFDataItem object)
void	<a href="#"><u>add(int index, AMFData data)</u></a> Insert an item into the array
void	<a href="#"><u>add(int index, boolean data)</u></a> Insert a boolean value (will be wrapped in an AMFDataItem object)
void	<a href="#"><u>add(int index, java.util.Date data)</u></a> Insert a date value (will be wrapped in an AMFDataItem object)
void	<a href="#"><u>add(int index, double data)</u></a> Insert a double value (will be wrapped in an AMFDataItem object)
void	<a href="#"><u>add(int index, int data)</u></a> Insert a int value (will be wrapped in an AMFDataItem object)
void	<a href="#"><u>add(int index, long data)</u></a> Insert a long value (will be wrapped in an AMFDataItem object)
void	<a href="#"><u>add(int index, String data)</u></a> Insert a string value (will be wrapped in an AMFDataItem object)
void	<a href="#"><u>add(long data)</u></a> Append a long (will be wrapped in an AMFDataItem object)
void	<a href="#"><u>add(String data)</u></a> Append a string (will be wrapped in an AMFDataItem object)
void	<a href="#"><u>deserialize(java.nio.ByteBuffer data)</u></a>
void	<a href="#"><u>deserialize(java.nio.ByteBuffer data, AMFDataContextDeserialize context)</u></a>
<a href="#"><u>AMFData</u></a>	<a href="#"><u>get(int index)</u></a> Get item at index
boolean	<a href="#"><u>getBoolean(int index)</u></a> Get item at index return as boolean
byte	<a href="#"><u>getByte(int index)</u></a> Get item at index return as byte
java.util.Date	<a href="#"><u>getDate(int index)</u></a> Get item at index return as Date
double	<a href="#"><u>getDouble(int index)</u></a> 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="#"><u>AMFDataObj</u></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="#"><u>AMFData</u></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(<u>AMFDataContextSerialize</u> context)</code></a>
byte[]	<a href="#"><code>serialize(<u>AMFDataContextSerialize</u> 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, <u>AMFDataContextSerialize</u> context)</code></a>
void	<a href="#"><code>serialize(java.io.DataOutputStream out, <u>AMFDataContextSerialize</u> 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, <u>AMFData</u> 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="#"><u>set</u></a> (int index, int data) Set an int value (will be wrapped in an AMFDataItem object)
void	<a href="#"><u>set</u></a> (int index, long data) Set an long value (will be wrapped in an AMFDataItem object)
void	<a href="#"><u>set</u></a> (int index, String data) Set an string value (will be wrapped in an AMFDataItem object)
int	<a href="#"><u>size</u></a> () Returns the number of items in array
String	<a href="#"><u>toString</u></a> () Return object as formatted string

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

[createContextDeserialize](#), [createContextSerialize](#), [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](#)

## 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,  
                    AMFDataContext Deserialize 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

---

**getByte**

```
public byte getByte(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,  
                      AMFDataContext Deserialize 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,  
                      AMFDataContext Serialize context)
```

Serialize object to output stream

---

**serialize**

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

Serial object to byte array

---

**serialize**

```
public byte[] serialize(AMFDataContext Serialize context,  
                      byte[] prepend)
```

---

### **getValue**

```
public Object getValue( )
```

Convert object to Java native class

---

### **toString**

```
public String toString( )
```

Return object as formatted string

## com.wowza.wms.amf Class AMFDataMixedArray

```
java.lang.Object
+- com.wowza.wms.amf.AMFData
  +- com.wowza.wms.amf.AMFDataObj
    +- com.wowza.wms.amf.AMFDataMixedArray
```

### All Implemented Interfaces:

[IAMFDataObj](#)

```
public class AMFDataMixedArray
extends AMFDataObj
```

AMFDataMixedArray: class for marshalling data between Wowza Pro server and Flash client. Array of mixed data types. An Array object created in the Flash client is wrapped in this data type when sent to the Wowza Pro server.

### Create Array of Strings

```
AMFDataMixedArray amfDataMixedArray = new AMFDataMixedArray();

amfDataMixedArray.put("0", "item1");
amfDataMixedArray.put("1", "item2");
amfDataMixedArray.put("2", "item3");
```

### Iterate Mixed Array

```
AMFDataMixedArray amfDataMixedArray;

int len = amfDataMixedArray.size();
for(int i=0;i<len;i++)
{
    String value = amfDataMixedArray.getString(i);
    int itemType = amfDataMixedArray.getType(i);
    WMSLoggerFactory.getLogger(null).debug("item:
[ "+i+": "+amfDataMixedArray.getKey(i)+" ]="+value+" type:"+itemType);
}
```

**NOTE:** A AMFDataMixedArray is exactly the same as a AMFDataObj except its type is DATA\_TYPE\_MIXED\_ARRAY.

This objects acts like a Map and a List at the same time. As items are added by key the order and position of each object is recorded. Objects can be retrieved either by key or by index.

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

[DECODE\\_OBJ\\_REF](#), [DECODE\\_TRAITS](#), [DECODE\\_TRAITS\\_EXT](#), [DECODE\\_TRAITS\\_REF](#), [DECODE\\_UNDEFINED](#), [members](#), [order](#), [serializer](#), [trait](#)

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

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

## Constructor Summary

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

## Method Summary

void	<a href="#">deserialize(java.nio.ByteBuffer data)</a>
void	<a href="#">deserialize(java.nio.ByteBuffer data, AMFDataContextDeserialize context)</a>
void	<a href="#">serialize(java.io.DataOutputStream out)</a>
void	<a href="#">serialize(java.io.DataOutputStream out, AMFDataContextSerialize context)</a>
void	<a href="#">serialize(java.io.DataOutputStream out, int objectEncoding)</a>
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, getByte, getByte, getClassName, getDate, getDate, getDouble, getDouble, getFloat, getFloat, getInt, getInt, getKey, getKeys, getLong, getLong, getObject, getObject, getSerializer, getShort, getShort, getString, getString, getTrait, getValue, put, put, put, put, remove, remove, serialize, serialize, serialize, serialize, size, toString
```

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

```
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
```

**Methods inherited from interface [com.wowza.wms.amf.IAMFDataObj](#)**

```
containsKey, get, get, getBoolean, getBoolean, getByte, getByte, getDate, getDate, getDouble, getDouble, getFloat, getFloat, getInt, getInt, getKey, getKeys, getLong, getLong, getObject, getObject, getShort, getShort, getString, getString, put, put, put, put, 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,  
                        AMFDataContext Deserialize context)
```

---

## Methods

### deserialize

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

Deserialize data in byte buffer

---

### deserialize

```
public void deserialize(java.nio.ByteBuffer data,  
                        AMFDataContext Deserialize 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,  
                     AMFDataContext Serialize 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: 1
public static final	<a href="#">DECODE_TRAITS</a> Value: 4
public static final	<a href="#">DECODE_TRAITS_EXT</a> Value: 3
public static final	<a href="#">DECODE_TRAITS_REF</a> Value: 2
public static final	<a href="#">DECODE_UNDEFINED</a> Value: 0
protected	<a href="#">members</a>
protected	<a href="#">order</a>
protected	<a href="#">serializer</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(byte[] data)</a> Deserialize entire data array and create AMFDataObj object
public	<a href="#">AMFDataObj(byte[] data, int offset, int size)</a> Deserialize data array starting at offset for size bytes and create AMFDataObj object
public	<a href="#">AMFDataObj(java.nio.ByteBuffer data)</a> 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="#">containsKey</a> (String name)
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)
<a href="#">AMFData</a>	<a href="#">get</a> (String name)
boolean	<a href="#">getBoolean</a> (int index)
boolean	<a href="#">getBoolean</a> (String name)
byte	<a href="#">getByte</a> (int index)
byte	<a href="#">getByte</a> (String name)
String	<a href="#">getClassName</a> ()
java.util.Date	<a href="#">getDate</a> (int index)
java.util.Date	<a href="#">getDate</a> (String name)
double	<a href="#">getDouble</a> (int index)
double	<a href="#">getDouble</a> (String name)
float	<a href="#">getFloat</a> (int index)
float	<a href="#">getFloat</a> (String name)
int	<a href="#">getInt</a> (int index)
int	<a href="#">getInt</a> (String name)
String	<a href="#">getKey</a> (int index)
java.util.List	<a href="#">getKeys</a> ()
long	<a href="#">getLong</a> (int index)
long	<a href="#">getLong</a> (String name)

<a href="#">AMFDataObj</a>	<a href="#">getObject(int index)</a>
<a href="#">AMFDataObj</a>	<a href="#">getObject(String name)</a>
<a href="#">IAMFExternalizable</a>	<a href="#">getSerializer()</a>
short	<a href="#">getShort(int index)</a>
short	<a href="#">getShort(String name)</a>
String	<a href="#">getString(int index)</a>
String	<a href="#">getString(String name)</a>
<a href="#">AMFDatatrait</a>	<a href="#">getTrait()</a>
Object	<a href="#">getValue()</a>
void	<a href="#">put(String name, AMFData data)</a>
void	<a href="#">put(String name, boolean data)</a>
void	<a href="#">put(String name, java.util.Date data)</a>
void	<a href="#">put(String name, double data)</a>
void	<a href="#">put(String name, int data)</a>
void	<a href="#">put(String name, long data)</a>
void	<a href="#">put(String name, String data)</a>
<a href="#">AMFData</a>	<a href="#">remove(int index)</a>
<a href="#">AMFData</a>	<a href="#">remove(String name)</a>
byte[]	<a href="#">serialize()</a>
byte[]	<a href="#">serialize(AMFDataContextSerialize context)</a>
void	<a href="#">serialize(java.io.DataOutputStream out)</a>
void	<a href="#">serialize(java.io.DataOutputStream out, AMFDataContextSerialize context)</a>
void	<a href="#">serialize(java.io.DataOutputStream out, int objectEncoding)</a>
byte[]	<a href="#">serialize(int objectEncoding)</a>

void	<a href="#">setClassName</a> (String className)
void	<a href="#">setSerializer</a> ( <a href="#">IAMFExternalizable</a> serializer)
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** [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](#)

**Methods inherited from interface** [com.wowza.wms.amf.IAMFDataObj](#)

[containsKey](#), [get](#), [get](#), [getBoolean](#), [getBoolean](#), [getByte](#), [getByte](#), [getDate](#), [getDate](#),  
[getDouble](#), [getDouble](#), [getFloat](#), [getFloat](#), [getInt](#), [getInt](#), [getKey](#), [getKeys](#), [getLong](#),  
[getLong](#), [getObject](#), [getObject](#), [getShort](#), [getShort](#), [getString](#), [getString](#), [put](#), [put](#),  
[put](#), [put](#), [put](#), [remove](#), [remove](#)

## Fields

**DECODE\_UNDEFINED**

```
public static final int DECODE_UNDEFINED
```

Constant value: 0

**DECODE\_OBJ\_REF**

```
public static final int DECODE_OBJ_REF
```

Constant value: 1

**DECODE\_TRAITS\_REF**

```
public static final int DECODE_TRAITS_REF
```

Constant value: 2

(continued from last page)

## DECODE\_TRAITS\_EXT

```
public static final int DECODE_TRAITS_EXT
```

Constant value: 3

## DECODE\_TRAITS

```
public static final int DECODE_TRAITS
```

Constant value: 4

## members

```
protected java.util.Map members
```

## order

```
protected java.util.List order
```

## trait

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

## serializer

```
protected com.wowza.wms.amf.IAMFExternalizable serializer
```

## Constructors

### AMFDataObj

```
public AMFDataObj()
```

Create empty AMFDataObj object

### AMFDataObj

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

Deserialize entire data array and create AMFDataObj object

**Parameters:**

data - binary data

(continued from last page)

## AMFDataObj

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

Deserialize data array starting at offest for size bytes and create AMFDataObj object

### Parameters:

data - binary data  
offset - starting offset into data  
size - size of data to deserialize

## AMFDataObj

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

Deserialize entire data array and create AMFDataObj object

### Parameters:

data - binary data

## AMFDataObj

```
public AMFDataObj(java.nio.ByteBuffer data,  
                  AMFDataContext Deserialize context)
```

## Methods

### size

```
public int size()
```

Return the number of members of this object/array

### Returns:

number of members

### containsKey

```
public boolean containsKey(String name)
```

### put

```
public void put(String name,  
                AMFData data)
```

### put

```
public void put(String name,  
                String data)
```

**put**

```
public void put(String name,  
               double data)
```

---

**put**

```
public void put(String name,  
               int data)
```

---

**put**

```
public void put(String name,  
               long data)
```

---

**put**

```
public void put(String name,  
               java.util.Date data)
```

---

**put**

```
public void put(String name,  
               boolean data)
```

---

**getKeys**

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

---

**getKey**

```
public String getKey(int index)
```

---

**get**

```
public AMFData get(String name)
```

---

**get**

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

---

**remove**

```
public AMFData remove(String name)
```

---

**remove**

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

---

**getString**

```
public String getString(String name)
```

---

**getInt**

```
public int getInt(String name)
```

---

**getLong**

```
public long getLong(String name)
```

---

**getShort**

```
public short getShort(String name)
```

---

**getDouble**

```
public double getDouble(String name)
```

---

**getFloat**

```
public float getFloat(String name)
```

---

**getByte**

```
public byte getByte(String name)
```

---

**getBoolean**

```
public boolean getBoolean(String name)
```

---

(continued from last page)

---

**getDate**

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

---

**getObject**

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

---

**getString**

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

---

**getInt**

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

---

**getLong**

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

---

**getShort**

```
public short getshort(int index)
```

---

**getByte**

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

---

**getDouble**

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

---

**getFloat**

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

---

(continued from last page)

**getBoolean**

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

**getDate**

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

**getObject**

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

**deserialize**

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

Deserialize data in byte buffer

**deserialize**

```
public void deserialize(java.nio.ByteBuffer data,  
AMFDataContext Deserialize 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,  
AMFDataContext Serialize context)
```

Serialize object to output stream

**serialize**

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

Serial object to byte array

---

(continued from last page)

### serialize

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

Serial object to byte array

---

### serialize

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

Serial object to byte array

---

### getValue

```
public Object getValue()
```

Convert object to Java native class

---

### toString

```
public String toString()
```

Return object as formatted string

---

### getClassName

```
public String getClassName()
```

---

### setClassName

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

---

### getTrait

```
public AMFDataTrait getTrait()
```

---

### getSerializer

```
public IAMFExternalizable getSerializer()
```

---

### setSerializer

```
public void setSerializer(IAMFExternalizable serializer)
```

## com.wowza.wms.amf Class AMFDataTrait

```
java.lang.Object
+-com.wowza.wms.amf.AMFDataTrait
```

---

```
public class AMFDataTrait
extends Object
```

AMF trait used to describe an AMF class in AMF3

### Constructor Summary

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

### Method Summary

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

<code>public static final</code>	<a href="#">AMFDEBUGHEADERSIZE</a> Value: <b>false</b>
<code>public static final</code>	<a href="#">WOWZDEBUGHEADERSIZE</a> Value: <b>false</b>

### Constructor Summary

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

### Method Summary

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

## Fields

**AMFDEBUGHEADERSIZE**

public static final boolean **AMFDEBUGHEADERSIZE**

Constant value: **false**

**WOWZDEBUGHEADERSIZE**

public static final boolean **WOWZDEBUGHEADERSIZE**

Constant value: **false**

## Constructors

**AMFObj**

public **AMFObj**(int id)

Create new AMFObj for a given channel (id)

**Parameters:**

id - channel id

**AMFObj**

public **AMFObj**(int id,  
int objectEncoding)

Create new AMFObj for a given channel (id)

(continued from last page)

**Parameters:**

- id - channel id
- objectEncoding - object encoding level (AMF0 or AMF3)

## Methods

### getChunks

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

Get the chunks that make up this packet

**Returns:**

- chunks that make up this packet

### addChunk

```
public void addChunk(byte[] buffer,  
                     int offset,  
                     int len)
```

Add a chunk to the chunk list

**Parameters:**

- buffer - buffer
- offset - offset
- len - length

### getId

```
public int getId()
```

Get channel id

**Returns:**

- channel id

### setId

```
public void setId(int id)
```

Set channel id

**Parameters:**

- id - channel id

### getSize

```
public int getSize()
```

Get packet size

**Returns:**

- packet size

### setSize

```
public void setSize(int size)
```

(continued from last page)

Set packet size

**Parameters:**

size - packet size

---

## getType

```
public int getType()
```

Get content type IVHost.CONTENTTYPE\_\*

**Returns:**

content type

---

## setType

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

Set content type IVHost.CONTENTTYPE\_\*

**Parameters:**

type - content type

---

## getSrc

```
public int getSrc()
```

Get stream id (0 if not stream data)

**Returns:**

stream id

---

## setSrc

```
public void setSrc(int src)
```

Set stream id (0 if not stream data)

**Parameters:**

src - stream id

---

## toString

```
public String toString()
```

Return object as formatted string

---

## getTimecode

```
public long getTimecode()
```

Get timecode (milliseconds) sometimes relative

**Returns:**

timecode (milliseconds)

---

(continued from last page)

## setTimecode

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

Set timecode (milliseconds) sometimes relative

**Parameters:**

timecode - timecode (milliseconds)

---

## isNew

```
public boolean isNew()
```

Is this a new packet. If new entire packet header needs to be sent

**Returns:**

is new packet

---

## setNew

```
public void setNew(boolean isNew)
```

Set is new packet

**Parameters:**

isNew - is new packet

---

## getAbsTimecode

```
public long getAbsTimecode()
```

Get the absolute time code

**Returns:**

absolute timecode

---

## setAbsTimecodeLong

```
public long setAbsTimecodeLong(long absTimecode)
```

Set the absolute timecode

**Parameters:**

absTimecode - absolute timecode

**Returns:**

absolute timecode

---

## setAbsTimecodeShort

```
public long setAbsTimecodeShort(long absTimecode)
```

Set the absolute timecode

**Parameters:**

absTimecode

**Returns:**

(continued from last page)

absolute timecode

---

### **incAbsTimecode**

```
public long incAbsTimecode(long absTimecode)
```

Fake container for processing

**Parameters:**

absTimecode - absolute timecode

**Returns:**

absolute timecode

---

### **getByteContainerLevel**

```
public int getByteContainerLevel()
```

Fake container for processing

**Returns:**

current container level

---

### **clearByteContainer**

```
public void clearByteContainer()
```

Clear the byte container

---

### **setByteContainerLevel**

```
public void setByteContainerLevel(int byteContainerLevel)
```

Fake container for processing

**Parameters:**

byteContainerLevel - current container level

---

### **incByteContainerLevel**

```
public void incByteContainerLevel(int byteContainerLevel)
```

Fake container for processing

**Parameters:**

byteContainerLevel - current container level

---

### **isByteContainerEmpty**

```
public boolean isByteContainerEmpty()
```

Fake container for processing

**Returns:**

is container full

---

(continued from last page)

**isByteContainerFull**

```
public boolean isByteContainerFull()
```

Fake container for processing

**Returns:**

is container full

---

**isLongTimecode**

```
public boolean isLongTimecode()
```

Get is a 32 bit timecode

**Returns:**

true if 32 bit timecode

---

**setLongTimecode**

```
public void setLongTimecode(boolean isLongTimecode)
```

Set is a 32 bit timecode

**Parameters:**

isLongTimecode - is a 32 bit timecode

---

**isLastSentAbsTimecode**

```
public boolean isLastSentAbsTimecode()
```

**setLastSentAbsTimecode**

```
public void setLastSentAbsTimecode(boolean isLastSentAbsTimecode)
```

**isObjectEncodingAMF3**

```
public boolean isObjectEncodingAMF3()
```

**isObjectEncodingAMF0**

```
public boolean isObjectEncodingAMF0()
```

**setObjectEncoding**

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

(continued from last page)

**getObjectEncoding**

```
public int getObjectEncoding( )
```

---

**getChunkCounter**

```
public long getChunkCounter( )
```

---

**setChunkCounter**

```
public void setChunkCounter(long chunkCounter)
```

## com.wowza.wms.amf Class AMFObjChunk

```
java.lang.Object
+-com.wowza.wms.amf.AMFObjChunk
```

---

```
public class AMFObjChunk
extends Object
```

### Field Summary

public	<a href="#">buffer</a>
public	<a href="#">len</a>
public	<a href="#">offset</a>

### Constructor Summary

```
public AMFObjChunk(byte[] buffer, int offset, int len)
```

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

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

### 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 AMFPacket

```
java.lang.Object
+-com.wowza.wms.amf.AMFPacket
```

---

```
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
public	<a href="#">AMFPacket(int type, int src, byte[] data)</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>
<a href="#">AMFPacket</a>	<a href="#">clone(boolean doCopy)</a>
boolean	<a href="#">containsExtraDataKey(String key)</a> Return true if extra data contains the key
static <a href="#">AMFPacket</a>	<a href="#">deserializePacket(byte[] buffer)</a>
java.util.Map	<a href="#">extraDataClone()</a> Deep copy the extra data and return as a Map
static java.util.Map	<a href="#">extraDataClone(java.util.Map extraData)</a>
long	<a href="#">getAbsTimecode()</a> Get absolute timecode (milliseconds)

byte[ ]	<a href="#"><u>getData()</u></a> Get data as byte[]
java.nio.ByteBuffer	<a href="#"><u>getDataBuffer()</u></a> Get data as ByteBuffer
<a href="#"><u>IAMFPacketExtraData</u></a>	<a href="#"><u>getExtraData(String key)</u></a> Get extra data value
java.util.Set	<a href="#"><u>getExtraDataKeys()</u></a> Get a set of the extra data keys
int	<a href="#"><u>getFirstByte()</u></a> Get first byte of data (used to peek into packet)
Object	<a href="#"><u>getLock()</u></a> Get lock (mostly used for extra data access)
int	<a href="#"><u>getMissing()</u></a> Get the number of bytes remaining unfilled in the packet
int	<a href="#"><u>getSecondByte()</u></a> Get second byte of data (used to peek into packet)
long	<a href="#"><u>getSeq()</u></a> Get packet sequence number.
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) relative to the <i>previous</i> packet.
int	<a href="#"><u>getType()</u></a> Get content type IVHost.CONTENTTYPE_*
boolean	<a href="#"><u>isAudio()</u></a> Is this an audio packet IVHost.CONTENTTYPE_AUDIO
boolean	<a href="#"><u>isExtraData()</u></a> Returns true if this packet contains any extra data
boolean	<a href="#"><u>isVideo()</u></a> Is this an audio packet IVHost.CONTENTTYPE_VIDEO
void	<a href="#"><u>putAllExtraData(java.util.Map entries)</u></a> Add extra data key value pairs
<a href="#"><u>IAMFPacketExtraData</u></a>	<a href="#"><u>putExtraData(String key, IAMFPacketExtraData value)</u></a> Add extra data value
byte[ ]	<a href="#"><u>serializePacket()</u></a>
void	<a href="#"><u>setAbsTimecode(long absTimecode)</u></a> Set absolute timecode (milliseconds)

void	<a href="#"><u>setDataBuffer</u></a> (byte[] data) Set the data buffer to a byte array
void	<a href="#"><u>setDataBuffer</u></a> (java.nio.ByteBuffer data) Set the data for this packet
void	<a href="#"><u>setSeq</u></a> (long seq) Set packet sequence.
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) relative,
void	<a href="#"><u>setTimecodes</u></a> (long timecode, long absTimecode) Set both relative and absolute timecode in one call (milliseconds)
void	<a href="#"><u>setType</u></a> (int type) Set content type IVHost.CONTENTTYPE_*
String	<a href="#"><u>toString</u></a> () Return object as formatted string
void	<a href="#"><u>truncatePacket</u></a> (int newSize)

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

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

## Constructors

### AMFPacket

```
public AMFPacket()
```

Create new empty packet

### AMFPacket

```
public AMFPacket(int type,  
                 int src,  
                 int size)
```

Create new packet with given values

**Parameters:**

- type - packet content type: IVHost.CONTENTTYPE\_\*
- src - stream id
- size - packet size (bytes)

(continued from last page)

## AMFPacket

```
public AMFPacket(int type,  
                 int src,  
                 byte[] data)
```

Create new packet with given values

### Parameters:

type - packet content type: IVHost.CONTENTTYPE\_\*

src - stream id

data - data buffer

## Methods

### clone

```
public AMFPacket clone()
```

### clone

```
public AMFPacket clone(boolean doCopy)
```

### getSize

```
public int getSize()
```

Get packet size

### Returns:

packet size

### setSize

```
public void setSize(int size)
```

Set packet size

### Parameters:

size - packet size

### truncatePacket

```
public void truncatePacket(int newSize)
```

### getMissing

```
public int getMissing()
```

Get the number of bytes remaining unfilled in the packet

### Returns:

number of bytes unfilled in packet

## setDataBuffer

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

Set the data for this packet

**Parameters:**

data - byte buffer that holds packet data

---

## setDataBuffer

```
public void setDataBuffer(byte[] data)
```

Set the data buffer to a byte array

**Parameters:**

data - data buffer byte array

---

## addData

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

Add data to the packet

**Parameters:**

data - byte buffer with data

offset - offset in byte buffer

size - size of data

**Returns:**

number of bytes unfilled in packet

---

## getType

```
public int getType()
```

Get content type IVHost.CONTENTTYPE\_\*

**Returns:**

content type

---

## setType

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

Set content type IVHost.CONTENTTYPE\_\*

**Parameters:**

type - content type

---

## getSrc

```
public int getSrc()
```

Get stream id (0 if not stream data)

(continued from last page)

**Returns:**

stream id

**setSrc****public void setSrc(int src)**

Set stream id (0 if not stream data)

**Parameters:**

src - stream id

**toString****public String toString()**

Return object as formatted string

**getTimecode****public long getTimecode()**Get timecode (milliseconds) relative to the *previous* packet.**Returns:**

timecode (milliseconds)

**setTimecodes****public void setTimecodes(long timecode,  
long absTimecode)**

Set both relative and absolute timecode in one call (milliseconds)

Note this time is relative to the *previous* packet.**Parameters:**

timecode - relative timecode (milliseconds)

absTimecode - absolute timecode (milliseconds)

**setTimecode****public void setTimecode(long timecode)**

Set timecode (milliseconds) relative,

Note this time is relative to the *previous* packet.**Parameters:**

timecode - timecode (milliseconds)

**getAbsTimecode****public long getAbsTimecode()**

Get absolute timecode (milliseconds)

**Returns:**

absolute timecode

## setAbsTimecode

```
public void setAbsTimecode(long absTimecode)
```

Set absolute timecode (milliseconds)

**Parameters:**

absTimecode - absolute timecode

---

## getSeq

```
public long getSeq()
```

Get packet sequence number. Used for live streams to keep track of packet ordering.

**Returns:**

packet sequence

---

## setSeq

```
public void setSeq(long seq)
```

Set packet sequence. Used for live streams to keep track of packet ordering.

**Parameters:**

seq - packet sequence

---

## isAudio

```
public boolean isAudio()
```

Is this an audio packet IVHost.CONTENTTYPE\_AUDIO

**Returns:**

true if audio packet

---

## isVideo

```
public boolean isVideo()
```

Is this an audio packet IVHost.CONTENTTYPE\_VIDEO

**Returns:**

true if video packet

---

## getDataBuffer

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

Get data as ByteBuffer

**Returns:**

data as ByteBuffer

---

## getData

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

(continued from last page)

Get data as byte[]

**Returns:**

data as byte[] null if no data

---

## getFirstByte

```
public int getFirstByte()
```

Get first byte of data (used to peek into packet)

**Returns:**

first byte of data in packet

---

## getSecondByte

```
public int getSecondByte()
```

Get second byte of data (used to peek into packet)

**Returns:**

second byte of data in packet

---

## calcTotalPacketSize

```
public static int calcTotalPacketSize(int packetSize,  
        int headerSize,  
        int chunkSize,  
        int amfNumber,  
        boolean isLongTimecode)
```

Calculate the total packet size for given packet parameters

**Parameters:**

packetSize - data size  
headerSize - header size  
chunkSize - chunk size  
amfNumber - amf number  
isLongTimecode - is long timecode

**Returns:**

total bytes

---

## addDataEx

```
public int addDataEx(byte[] srcData,  
        int srcOffset,  
        int destOffset,  
        int srcBytes)
```

Add data to the packet

**Parameters:**

srcData -- source byte buffer with data  
srcOffset -- start copying from source buffer at this offset  
destOffset -- copy into destination buffer from this offset  
srcBytes -- size of data to copy

**Returns:**

(continued from last page)

number of bytes unfilled in packet

---

### serializePacket

```
public byte[] serializePacket()
```

---

### deserializePacket

```
public static AMFPacket deserializePacket(byte[] buffer)
```

---

### getLock

```
public Object getLock()
```

Get lock (mostly used for extra data access)

**Returns:**

lock

---

### putExtraData

```
public IAMFPacketExtraData putExtraData(String key,  
                                         IAMFPacketExtraData value)
```

Add extra data value

**Parameters:**

key - key  
value - extra data

**Returns:**

extra data currently at that key

---

### putAllExtraData

```
public void putAllExtraData(java.util.Map entries)
```

Add extra data key value pairs

**Parameters:**

entries - extra data key value pairs

---

### getExtraData

```
public IAMFPacketExtraData getExtraData(String key)
```

Get extra data value

**Parameters:**

key - key

**Returns:**

extra data value

---

## getExtraDataKeys

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

Get a set of the extra data keys

**Returns:**

extra data keys

---

## containsExtraDataKey

```
public boolean containsExtraDataKey(String key)
```

Return true if extra data contains the key

**Parameters:**

key - key

**Returns:**

true if extra data contains the key

---

## isExtraData

```
public boolean isExtraData()
```

Returns true if this packet contains any extra data

**Returns:**

true if this packet contains any extra data

---

## extraDataClone

```
public static java.util.Map extraDataClone(java.util.Map extraData)
```

---

## extraDataClone

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

Deep copy the extra data and return as a Map

**Returns:**

deep copy of extra data

## com.wowza.wms.amf Interface IAMFDataObj

All Known Implementing Classes:  
[AMFDataObj](#)

public interface **IAMFDataObj**  
 extends

### Method Summary

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

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

---

## getByte

```
public byte getByte(String name)
```

Get item at key return as byte

**Parameters:**

name - key

**Returns:**

Return item as byte or 0 if out of bounds

---

## getBoolean

```
public boolean getBoolean(String name)
```

Get item at key return as boolean

**Parameters:**

name - key

**Returns:**

Return item as boolean or false if out of bounds

---

## getDate

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

Get item at key return as Date

**Parameters:**

name - key

**Returns:**

Return item as Date or null if out of bounds

---

## getObject

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

Get item at key return as AMFDataObj

**Parameters:**

name - key

**Returns:**

Return item as AMFDataObj or null if out of bounds

---

## getString

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

Get item at index return as String

**Parameters:**

index - index

(continued from last page)

**Returns:**

Return item as String or null if out of bounds

---

**getInt****public int getInt(int index)**

Get item at index return as int

**Parameters:**

index - index

**Returns:**

Return item as int or 0 if out of bounds

---

**getLong****public long getLong(int index)**

Get item at index return as long

**Parameters:**

index - index

**Returns:**

Return item as long or 0 if out of bounds

---

**getShort****public short getShort(int index)**

Get item at index return as short

**Parameters:**

index - index

**Returns:**

Return item as short or 0 if out of bounds

---

**getByte****public byte getByte(int index)**

Get item at index return as byte

**Parameters:**

index - index

**Returns:**

Return item as byte or 0 if out of bounds

---

**getDouble****public double getDouble(int index)**

Get item at index return as double

**Parameters:**

(continued from last page)

`index - index`**Returns:**

Return item as double or 0 if out of bounds

---

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

Get item at index return as float

**Parameters:**`index - index`**Returns:**

Return item as float or 0 if out of bounds

---

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

Get item at index return as boolean

**Parameters:**`index - index`**Returns:**

Return item as boolean or false if out of bounds

---

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

Get item at index return as Date

**Parameters:**`index - index`**Returns:**

Return item as Date or null if out of bounds

---

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

Get item at index return as AMFDataObj

**Parameters:**`index - index`**Returns:**

Return item as AMFDataObj or null if out of bounds

## com.wowza.wms.amf Interface IAMFExternalizable

public interface IAMFExternalizable  
extends

### Method Summary

void	<a href="#">deserialize(AMFDataObj amfObj, java.nio.ByteBuffer data, AMFDataContextDeserialize context)</a>
String	<a href="#">getClassName()</a>
void	<a href="#">serialize(AMFDataObj amfObj, java.io.DataOutputStream out, AMFDataContextSerialize context)</a>
void	<a href="#">setClassName(String className)</a>

### Methods

#### deserialize

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

#### serialize

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

#### getClassName

```
public String getClassName()
```

#### setClassName

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

## com.wowza.wms.amf Interface IAMFPacketExtraData

---

public interface IAMFPacketExtraData  
extends

### Method Summary

<a href="#">IAMFPacketExtraData</a>	<a href="#">clone()</a>
	Return a deep copy of the extra data item

### Methods

#### clone

public [IAMFPacketExtraData](#) **clone()**

Return a deep copy of the extra data item

**Returns:**

deep copy of the extra data item

---

Package

**com.wowza.wms.application**

## com.wowza.wms.application Interface IApplication

public interface IApplication  
extends

IApplication: public interface to Application object

### Field Summary

public static final	<a href="#">DEFAULT_APPLICATION_NAME</a>
	Value: <code>_defapp_</code>

### Method Summary

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

void	<a href="#"><code>getProtocolUsage(boolean[] protocolsInUse)</code></a> Get the protocols in use by this application instance (see IApplicationInstance.PROTOCOLUSAGE_*)
String	<a href="#"><code>getTimeRunning()</code></a> Get time application running
double	<a href="#"><code>getTimeRunningSeconds()</code></a> Get time running in seconds
<a href="#"><u>IVHost</u></a>	<a href="#"><code>getVHost()</code></a> Get the parent vHost object
boolean	<a href="#"><code>isAppInstanceLoaded(String name)</code></a> Return true if application instance is loaded
boolean	<a href="#"><code>isRunning()</code></a> Returns true if the application is in running state
String	<a href="#"><code>readAppConfig(String sName)</code></a> Method to read xml config file..
void	<a href="#"><code>removeAppInstance(IApplicationInstance appInstance)</code></a> Disconnect all clients connected to an application instance and remove it from the IApplication application list.
void	<a href="#"><code>removeApplicationInstanceListener(IApplicationInstanceNotify applicationInstanceListener)</code></a> Remove applicationInstance listener.
void	<a href="#"><code>setName(String name)</code></a> Set name of application
void	<a href="#"><code>shutdown(boolean isServerShutdown)</code></a> shutdown application
void	<a href="#"><code>shutdownAppInstance(String appInstanceName)</code></a> Shutdown an application instance by name.
boolean	<a href="#"><code>writeAppConfig(String sName, String data)</code></a> Method to write xml config file..

## Fields

### **DEFAULT\_APPLICATION\_NAME**

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

Constant value: `_defapp_`

## Methods

### **shutdown**

```
public void shutdown(boolean isServerShutdown)
```

shutdown application

(continued from last page)

**Parameters:**

isServerShutdown - true if due to shutdown of server

---

**getApplicationPath****public String getApplicationPath()**

Get the root path for application

**Returns:**

root path for application

---

**getConfigPath****public String getConfigPath()**

Get full path to Application.xml file

**Returns:**

full path to Application.xml file

---

**getAppInstance****public IApplicationInstance getAppInstance(String name)**

Get applicationInstance object by name

**Parameters:**

name - applicationInstance name

**Returns:**

applicationInstance object

---

**isAppInstanceLoaded****public boolean isAppInstanceLoaded(String name)**

Return true if application instance is loaded

**Parameters:**

name - applicationInstance name

**Returns:**

true if application instance is loaded

---

**getName****public String getName()**

Get the name of application

**Returns:**

name of application

---

**setName****public void setName(String name)**

(continued from last page)

Set name of application

**Parameters:**

name - name of application

---

**getVHost**

```
public IVHost getVHost()
```

Get the parent vHost object

**Returns:**

parent vHost

---

**getProperties**

```
public WMSProperties getProperties()
```

Get application properties

**Returns:**

application properties

---

**addApplicationInstanceListener**

```
public void addApplicationInstanceListener(IApplicationInstanceNotify  
applicationInstanceListener)
```

Add applicationInstance listener. Will be invoked each time applicationInstance created/deleted

**Parameters:**

applicationInstanceListener - applicationInstance listener

---

**removeApplicationInstanceListener**

```
public void removeApplicationInstanceListener(IApplicationInstanceNotify  
applicationInstanceListener)
```

Remove applicationInstance listener. Will be invoked each time applicationInstance created/deleted

**Parameters:**

applicationInstanceListener - applicationInstance listener

---

**getConnectionCounter**

```
public ConnectionCounter getConnectionCounter()
```

Get the connectionCounter for application

**Returns:**

connectionCounter for application

---

**getConnectionCounter**

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

Get the connectionCounter for application for application for a specific technology (see IVHost.COUNTER\_\*)

(continued from last page)

**Parameters:**

counterIndex - counter index (see IVHost.COUNTER\_\*)

**Returns:**

connection ocounter

---

**getIoPerformanceCounter****public IOPerformanceCounter getIoPerformanceCounter( )**

Get the performance counter for application

**Returns:**

performance counter for application

---

**getIoPerformanceCounter****public IOPerformanceCounter getIoPerformanceCounter(int counterIndex)**

Get the performance counter for application for a specific technology (see IVHost.COUNTER\_\*)

**Parameters:**

counterIndex - counter index (see IVHost.COUNTER\_\*)

**Returns:**

performance counter

---

**getDateStarted****public String getDateStarted( )**

Get date application started

**Returns:**

date application started

---

**getTimeRunning****public String getTimeRunning( )**

Get time application running

**Returns:**

time application running

---

**getTimeRunningSeconds****public double getTimeRunningSeconds( )**

Get time running in seconds

**Returns:**

time running in seconds

---

**getAppInstanceNames****public java.util.List getAppInstanceNames( )**

(continued from last page)

Get a list of application instance names

**Returns:**

list of application instance names

## removeAppInstance

```
public void removeAppInstance(IAppliationInstance 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(IAppliationInstance appInstance)
{
    IVHost vhost = appInstance.getVHost();
    IApplication app = appInstance.getApplication();

    WMSReadWriteLock appLock = vhost.getApplicationLock();
    appLock.writeLock().lock();
    try
    {
        app.removeAppInstance(appInstance);
    }
    catch (Exception e)
    {
        WMSLoggerFactory.getLogger(Application.class).error("Application.shutdownAppInstance ", e);
    }
    finally
    {
        appLock.writeLock().unlock();
    }
}
```

**Parameters:**

appInstance - application instance to remove

## shutdownAppInstance

```
public void shutdownAppInstance(String appInstanceName)
```

Shutdown an application instance by name. This will disconnect all clients connected to this application instance.

**Parameters:**

appInstanceName - application instance name

(continued from last page)

## **readAppConfig**

```
public String readAppConfig(String sName)
```

Method to read xml config file..

---

## **writeAppConfig**

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

Method to write xml config file..

---

## **getProtocolUsage**

```
public void getProtocolUsage(boolean[] protocolsInUse)
```

Get the protocols in use by this application instance (see IApplicationInstance.PROTOCOLUSAGE\_\*)

---

## **isRunning**

```
public boolean isRunning()
```

Returns true if the application is in running state

**Returns:**

true if the application is in running state

## com.wowza.wms.application Interface IApplicationInstance

public interface IApplicationInstance  
extends

IApplicationInstance: public interface to ApplicationInstance object

### Field Summary

public static final	<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_MC_LIVEREPEATER</u></a>
	Value: <b>44</b>
public static final	<a href="#"><u>PROTOCOLUSAGE_MC_RTP</u></a>
	Value: <b>45</b>
public static final	<a href="#"><u>PROTOCOLUSAGE_MC_SHOUTCAST</u></a>
	Value: <b>43</b>
public static final	<a href="#"><u>PROTOCOLUSAGE_MEDIACACHE</u></a>
	Value: <b>20</b>
public static final	<a href="#"><u>PROTOCOLUSAGE_MPEGDASH</u></a>
	Value: <b>10</b>
public static final	<a href="#"><u>PROTOCOLUSAGE_ORIGINCUPERTINO</u></a>
	Value: <b>15</b>
public static final	<a href="#"><u>PROTOCOLUSAGE_ORIGINMPEGDASH</u></a>
	Value: <b>18</b>
public static final	<a href="#"><u>PROTOCOLUSAGE_ORIGINSANJOSE</u></a>
	Value: <b>17</b>
public static final	<a href="#"><u>PROTOCOLUSAGE_ORIGINSMOOTH</u></a>
	Value: <b>16</b>
public static final	<a href="#"><u>PROTOCOLUSAGE_PUSHUBLISHING</u></a>
	Value: <b>19</b>

public static final	<a href="#"><u>PROTCOLUSAGE_RTMP</u></a>
	Value: <b>0</b>
public static final	<a href="#"><u>PROTCOLUSAGE_RTMP_IN</u></a>
	Value: <b>21</b>
public static final	<a href="#"><u>PROTCOLUSAGE_RTMP_OUT</u></a>
	Value: <b>22</b>
public static final	<a href="#"><u>PROTCOLUSAGE_RTMPE</u></a>
	Value: <b>4</b>
public static final	<a href="#"><u>PROTCOLUSAGE_RTMPE_IN</u></a>
	Value: <b>29</b>
public static final	<a href="#"><u>PROTCOLUSAGE_RTMPE_OUT</u></a>
	Value: <b>30</b>
public static final	<a href="#"><u>PROTCOLUSAGE_RTMPS</u></a>
	Value: <b>2</b>
public static final	<a href="#"><u>PROTCOLUSAGE_RTMPS_IN</u></a>
	Value: <b>25</b>
public static final	<a href="#"><u>PROTCOLUSAGE_RTMPS_OUT</u></a>
	Value: <b>26</b>
public static final	<a href="#"><u>PROTCOLUSAGE_RTMPT</u></a>
	Value: <b>1</b>
public static final	<a href="#"><u>PROTCOLUSAGE_RTMPT_IN</u></a>
	Value: <b>23</b>
public static final	<a href="#"><u>PROTCOLUSAGE_RTMPT_OUT</u></a>
	Value: <b>24</b>
public static final	<a href="#"><u>PROTCOLUSAGE_RTMPTE</u></a>
	Value: <b>5</b>
public static final	<a href="#"><u>PROTCOLUSAGE_RTMPTE_IN</u></a>
	Value: <b>31</b>
public static final	<a href="#"><u>PROTCOLUSAGE_RTMPTE_OUT</u></a>
	Value: <b>32</b>
public static final	<a href="#"><u>PROTCOLUSAGE_RTMPTS</u></a>
	Value: <b>3</b>

public static final	<a href="#"><u>PROTOCOLUSAGE_RTMPTS_IN</u></a>
	Value: <b>27</b>
public static final	<a href="#"><u>PROTOCOLUSAGE_RTMPTS_OUT</u></a>
	Value: <b>28</b>
public static final	<a href="#"><u>PROTOCOLUSAGE_RTP</u></a>
	Value: <b>6</b>
public static final	<a href="#"><u>PROTOCOLUSAGE_RTP_IN</u></a>
	Value: <b>33</b>
public static final	<a href="#"><u>PROTOCOLUSAGE_RTP_OUT</u></a>
	Value: <b>34</b>
public static final	<a href="#"><u>PROTOCOLUSAGE_RTSP_IN</u></a>
	Value: <b>35</b>
public static final	<a href="#"><u>PROTOCOLUSAGE_RTSP_OUT</u></a>
	Value: <b>36</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>46</b>
public static final	<a href="#"><u>PROTOCOLUSAGE_WEBM</u></a>
	Value: <b>11</b>
public static final	<a href="#"><u>PROTOCOLUSAGE_WOWZ</u></a>
	Value: <b>12</b>
public static final	<a href="#"><u>PROTOCOLUSAGE_WOWZ_IN</u></a>
	Value: <b>37</b>
public static final	<a href="#"><u>PROTOCOLUSAGE_WOWZ_OUT</u></a>
	Value: <b>38</b>
public static final	<a href="#"><u>PROTOCOLUSAGE_WOWZE</u></a>
	Value: <b>13</b>
public static final	<a href="#"><u>PROTOCOLUSAGE_WOWZE_IN</u></a>
	Value: <b>39</b>

public static final	<a href="#">PROTOCOLUSAGE_WOWZE_OUT</a> Value: <b>40</b>
public static final	<a href="#">PROTOCOLUSAGE_WOWZS</a> Value: <b>14</b>
public static final	<a href="#">PROTOCOLUSAGE_WOWZS_IN</a> Value: <b>41</b>
public static final	<a href="#">PROTOCOLUSAGE_WOWZS_OUT</a> Value: <b>42</b>

## Method Summary

void	<a href="#">addClientListener(<i>IClientNotify</i> clientListener)</a> Add client listener.
void	<a href="#">addDvrRecorderListener(<i>ILiveStreamDvrRecorderActionNotify</i> listener)</a> Add a Dvr Recorder listener (see: <i>ILiveStreamDvrRecorderActionNotify</i> )
void	<a href="#">addDvrStreamManagerListener(<i>IDvrStreamManagerActionNotify</i> listener)</a> Add a Dvr Application Store Manager listener (see: <i>IDvrStoreActionNotify</i> )
void	<a href="#">addHTTPListener(<i>IHTTPSessionNotify</i> httpListener)</a> Add listner to be notified of <i>HTTPSession</i> create and destroy.
void	<a href="#">addHTTPStreamerSession(<i>IHTTPStreamerSession</i> httpStreamerSession)</a> Add a <i>HTTPStreamerSession</i> to this application instance
void	<a href="#">addLiveStreamPacketizerListener(<i>ILiveStreamPacketizerActionNotify</i> liveStreamPacketizerListener)</a> Add a Live Stream Packetizer listener (see: <i>ILiveStreamPacketizerActionNotify</i> )
void	<a href="#">addLiveStreamTranscoderListener(<i>ILiveStreamTranscoderNotify</i> liveStreamTranscoderListener)</a> Add a live stream transcoder listener
void	<a href="#">addMediaCasterListener(<i>IMediaCasterNotify</i> mediaCasterListener)</a> Add mediaCaster listener.
void	<a href="#">addMediaCasterListener(<i>IMediaCasterNotify2</i> mediaCasterListener)</a> Add mediaCaster listener.
void	<a href="#">addMediaReaderListener(<i>IMediaReaderActionNotify</i> mediaReaderListener)</a> Add media reader listener.
void	<a href="#">addMediaStreamListener(<i>IMediaStreamNotify</i> mediaStreamListener)</a> Add mediaStream listener.
void	<a href="#">addMediaWriterListener(<i>IMediaWriterActionNotify</i> listener)</a> Add a MediaWriter listener class.
void	<a href="#">addModuleListener(<i>IModuleNotify</i> moduleListener)</a> Add module listener.
void	<a href="#">addPlayStreamByName(<i>IMediaStream</i> stream, String name)</a> Add a media stream to the list of streams that are listening for a published stream

void	<a href="#">addPublisher(Publisher publisher)</a> Add a server side publisher to this application instance
void	<a href="#">addPushPublishSession(IMediaStream stream, IPushPublishSession pushPublishSession)</a> Add a push publish session to the list of active push publish sessions
void	<a href="#">addPushPublishSessionListener(IPushPublishSessionNotify pushPublishSessionListener)</a> Add a push publish session listener (see IPushPublishSessionNotify)
void	<a href="#">addRTPIncomingDatagramPortAll()</a> Allow all incoming RTP UDP ports for this application instance
void	<a href="#">addRTPIncomingDatagramPortRange(int startPort, int endPort)</a> Add a port range to the list of valid incoming RTP UDP ports
void	<a href="#">addRTPListener(IRTPSessionNotify rtpListener)</a> Add listner to be notified of RTPSession create and destroy.
void	<a href="#">addRTPSession(RTPSession rtpSession)</a> Add an RTP session to this application instance
void	<a href="#">addSharedObjectListener(SharedObjectNotify sharedObjectListener, boolean isPersistent)</a> Add sharedObject listener.
void	<a href="#">broadcastMsg(java.util.List clientList, String handlerName)</a> Broadcast a message to a specific list of clients connected to this application instance
void	<a href="#">broadcastMsg(java.util.List clientList, String handlerName, Object[] params)</a> Broadcast a message to a specific list of clients connected to this application instance
void	<a href="#">broadcastMsg(String handlerName, Object[] params)</a> Broadcast a message to all clients connected to this application instance
boolean	<a href="#">containsDvrRecorder(String dvrRecorder)</a> Does this application instance contain a references to this DVR recorder.
boolean	<a href="#">containsHTTPStreamer(String httpStreamer)</a> Does this application instance allow streaming of a given HTTPStreamer
boolean	<a href="#">containsLiveStreamPacketizer(String liveStreamPacketizer)</a> Does this application instance contain a references to this live stream packetizer.
boolean	<a href="#">containsLiveStreamTranscoder(String liveStreamTranscoder)</a> Return true if this application instance contains the transcoder name
String	<a href="#">decodeStorageDir(String storageDir)</a> This function will take a storage path that uses variables and expand the variables based on the context.
String[]	<a href="#">getAllowDomains()</a> Get the list of domain names used to control access to this application.
IApplication	<a href="#">getApplication()</a> Get parent application
int	<a href="#">getApplicationInstanceTouchTimeout()</a> Get the application instance touch timeout (milliseconds).

int	<a href="#"><code>getApplicationTimeout()</code></a> Get application timeout (milliseconds)
<a href="#"><u>IClient</u></a>	<a href="#"><code>getClient(int index)</code></a> <b>Deprecated.</b> Get the client connection at index. This method is deprecated. It is best to use <code>getClient()</code> to return a List objects.
<a href="#"><u>IClient</u></a>	<a href="#"><code>getClientById(int index)</code></a> Get a client connection by the client Id
int	<a href="#"><code>getClientCount()</code></a> Get number of client connections currently connected to applicationInstance
int	<a href="#"><code>getClientCountTotal()</code></a> Get number of client connections in total that have connected to this applicationInstance
int	<a href="#"><code>getClientIdleFrequency()</code></a> Get default client idle frequency (milliseconds)
java.util.List	<a href="#"><code>getClients()</code></a> Get the set of clients currently connected to this application instance (replaces <code>getClient(index)</code> )
edu.emory.mathcs.backport.java.util.concurrent.locks.WMSReadWriteLock	<a href="#"><code>getClientsLockObj()</code></a> Get the read/write lock for this application instance
<a href="#"><u>ConnectionCounter</u></a>	<a href="#"><code>getConnectionCounter()</code></a> Get the connectionCounter for applicationInstance
ConnectionCounterSimple	<a href="#"><code>getConnectionCounter(int counterIndex)</code></a> Get the connectionCounter for applicationInstance for a specific technology (see IVHost.COUNTER_*)
String	<a href="#"><code>getContextStr()</code></a> Returns the application context string in the form [application]/[appInstance].
String	<a href="#"><code>getDateStarted()</code></a> Get date applicationInstance started
DvrApplicationContext	<a href="#"><code>getDvrApplicationContext()</code></a> Get live stream dvr application context
<a href="#"><u>WMSProperties</u></a>	<a href="#"><code>getDvrProperties()</code></a> Get the property collection of DVR settings that are specific to this application instance.
String	<a href="#"><code>getDvrRecorderList()</code></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="#"><code>getHTTPStreamerApplicationContext(String httpStreamName, boolean doCreate)</code></a> Get the HTTPStreamer application context for a given HTTPStreamer adapter
String	<a href="#"><code>getHTTPStreamerList()</code></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="#"><code>getHTTPStreamerProperties()</code></a> Get the property collection of HTTPStreamer settings that are specific to this application instance

int	<a href="#"><code>getHTTPStreamerSessionCount()</code></a> Get the current number of HTTPStreamerSessions associated with this application instance
int	<a href="#"><code>getHTTPStreamerSessionCount(int protocol)</code></a> Get the current number of HTTPStreamerSessions associated with this application instance by protocol.
int	<a href="#"><code>getHTTPStreamerSessionCount(int protocol, String streamName)</code></a> Get the current number of HTTPStreamerSessions associated with this application instance and stream name by protocol .
int	<a href="#"><code>getHTTPStreamerSessionCount(String streamName)</code></a> Get the current number of HTTPStreamerSessions associated with this application instance and stream name
java.util.Map	<a href="#"><code>getHTTPStreamerSessionCountsByName(int protocol)</code></a> Get a map of session counts by name for a given protocol
java.util.List	<a href="#"><code>getHTTPStreamerSessions()</code></a> Get the HTTPStreamerSessions associated with this application instance
java.util.List	<a href="#"><code>getHTTPStreamerSessions(int protocol)</code></a> Get the HTTPStreamerSessions associated with this application instance by protocol.
java.util.List	<a href="#"><code>getHTTPStreamerSessions(int protocol, String streamName)</code></a> Get the HTTPStreamerSessions associated with this application instance for a stream name by protocol.
java.util.List	<a href="#"><code>getHTTPStreamerSessions(String streamName)</code></a> Get the HTTPStreamerSessions associated with this application instance for a stream name
<a href="#"><u>IOPerformanceCounter</u></a>	<a href="#"><code>getIOPerformanceCounter()</code></a> Get the performance counter for applicationInstance
<a href="#"><u>IOPerformanceCounter</u></a>	<a href="#"><code>getIOPerformanceCounter(int counterIndex)</code></a> Get the performance counter for applicationInstance for a specific technology (see IVHost.COUNTER_*)
long	<a href="#"><code>getLastTouchTime()</code></a> Get the last time the instance was touched (milliseconds)
<a href="#"><u>ILiveStreamDvrRecorderControl</u></a>	<a href="#"><code>getLiveStreamDvrRecorderControl()</code></a> Get the Live Stream DVR Recorder Controller.
<a href="#"><u>ILiveStreamPacketizerControl</u></a>	<a href="#"><code>getLiveStreamPacketizerControl()</code></a> Get the Live Stream Packetizer Controller.
String	<a href="#"><code>getLiveStreamPacketizerList()</code></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="#"><code>getLiveStreamPacketizerProperties()</code></a> Get the property collection of LiveStreamPacketizer settings that are specific to this application instance
<a href="#"><u>ILiveStreamTranscoderControl</u></a>	<a href="#"><code>getLiveStreamTranscoderControl()</code></a> Get the Live Stream Transcoder Controller.
String	<a href="#"><code>getLiveStreamTranscoderList()</code></a> Get comma separated list of transcoders to use for this application instance

<a href="#"><u>WMSProperties</u></a>	<a href="#"><u>getManagerProperties()</u></a> Get Application's manager's properties collection
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
java.util.List	<a href="#"><u>getMediaCacheFilters()</u></a> Get the list of allowed MediaCache sources.
<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>getMediacasterRTPRTSPRTPTTransportMode()</u></a> RTP MediaCaster RTSP/RTP transport mode.
<a href="#"><u>MediaCasterStreamMap</u></a>	<a href="#"><u>getMediaCasterStreams()</u></a> Get the media caster streams attached to this application instance
<a href="#"><u>IMediaCasterValidateMediaCaster</u></a>	<a href="#"><u>getMediaCasterValidator()</u></a> Get the MediaCaster validator interface for this application instance
<a href="#"><u>IMediaListProvider</u></a>	<a href="#"><u>getMediaListProvider()</u></a> Get the current media list provider.
int	<a href="#"><u>getMediaReaderContentType(String mediaType)</u></a> Get the content type of a media stream name prefix (see IMediaReader.CONTENTTYPE_*)
<a href="#"><u>WMSProperties</u></a>	<a href="#"><u>getMediaReaderProperties()</u></a> Get the property collection of media reader settings that are specific to this application instance
<a href="#"><u>WMSProperties</u></a>	<a href="#"><u>getMediaWriterProperties()</u></a> Get the property collection of media writer settings that are specific to this application instance
<a href="#"><u>ModuleFunctions</u></a>	<a href="#"><u>getModFunctions()</u></a> Get list of application modules
Object	<a href="#"><u>getModuleInstance(String name)</u></a> Get the instance of the module class for this application instance.
<a href="#"><u>ModuleList</u></a>	<a href="#"><u>getModuleList()</u></a> Get the list of loaded modules.
String	<a href="#"><u>getName()</u></a> Get applicationInstance name
int	<a href="#"><u>getPingTimeout()</u></a> Get ping timeout (milliseconds)

int	<a href="#"><code>getPlayStreamCount(String streamName)</code></a> Get the number of Flash players playing a given stream name
java.util.Map	<a href="#"><code>getPlayStreamCountsByName()</code></a> Get a map of stream names to number of Flash players playing the stream name
java.util.List	<a href="#"><code>getPlayStreamsByName(String name)</code></a> Get a list of media streams that are listening for published stream.
<a href="#"><u>WMSProperties</u></a>	<a href="#"><code>getProperties()</code></a> Get applicationInstance properties
boolean[]	<a href="#"><code>getProtocolUsage()</code></a> Get the protocols in use by this application instance (see IApplicationInstance.PROTOCOLUSAGE_*)
void	<a href="#"><code>getProtocolUsage(boolean[] protocolsInUse)</code></a> Get the protocols in use by this application instance (see IApplicationInstance.PROTOCOLUSAGE_*)
int	<a href="#"><code>getPublisherCount()</code></a> Get the current number of server side publishers
java.util.List	<a href="#"><code>getPublishers()</code></a> Get the set of server side publishers
java.util.List	<a href="#"><code>getPublishStreamNames()</code></a> Get the list of live stream names currently being published.
int	<a href="#"><code>getPushPublishSessionCount()</code></a> Get the number of push publishing sessions.
java.util.List	<a href="#"><code>getPushPublishSessions()</code></a> Get all active push publish sessions
java.util.List	<a href="#"><code>getPushPublishSessions(IMediaStream stream)</code></a> Get the current push publish sessions for a given IMediaStream
String	<a href="#"><code>getRepeaterOriginUrl()</code></a> Get the Repeater Origin URL used by the Live Stream Repeater
String	<a href="#"><code>getRepeaterQueryString()</code></a> Get the Repeater query string that is used to connect to the origin.
String	<a href="#"><code>getRsoStorageDir()</code></a> Get remote shared object storage path
String	<a href="#"><code>getRsoStoragePath()</code></a> Get the resolved storage path to the shared objects
int	<a href="#"><code>getRTPAVSyncMethod()</code></a> Get RTP audio/video sync method (RTPStream.AVSNCMETHODS_SENDERREPORT, RTPStream.AVSNCMETHODS_SYSTEMCLOCK, RTPStream.AVSNCMETHODS_RTPTIMECODE)
int	<a href="#"><code>getRTPIidleFrequency()</code></a> Set the default RTP idle frequency (milliseconds)
int	<a href="#"><code>getRTPMaxRTCPWaitTime()</code></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="#"><code>getStreamAudioSampleAccess()</code></a> Get the default stream audio sample access
int	<a href="#"><code>getStreamCount()</code></a> Get the total number of open streams attached to this application instance
<a href="#"><u>IMediaStreamFileMapper</u></a>	<a href="#"><code>getStreamFileMapper()</code></a> Get the stream file mapper.
String	<a href="#"><code>getStreamKeyDir()</code></a> Get the stream key path
String	<a href="#"><code>getStreamKeyPath()</code></a> Get the resolved key path to the MediaStreams encryption keys
<a href="#"><u>IMediaStreamNameAliasProvider</u></a>	<a href="#"><code>getStreamNameAliasProvider()</code></a> Get the stream name alias provider
<a href="#"><u>WMSProperties</u></a>	<a href="#"><code>getStreamProperties()</code></a> Get the property collection of stream settings that are specific to this application instance
String	<a href="#"><code>getStreamReadAccess()</code></a> Get the default stream read access
<a href="#"><u>WMSProperties</u></a>	<a href="#"><code>getStreamRecorderProperties()</code></a> Get the property collection of StreamRecorder settings that are specific to this application instance
<a href="#"><u>MediaStreamMap</u></a>	<a href="#"><code>getStreams()</code></a> Get all the mediaStream objects attached to this applicationInstance
String	<a href="#"><code>getStreamStorageDir()</code></a> Get stream storage path
String	<a href="#"><code>getStreamStoragePath()</code></a> Get the resolved storage path to the MediaStreams
String	<a href="#"><code>getStreamType()</code></a> Get default streamType for application.
String	<a href="#"><code>getStreamVideoSampleAccess()</code></a> Get the default stream video sample access
String	<a href="#"><code>getStreamWriteAccess()</code></a> Get the default stream write access
<a href="#"><u>WMSProperties</u></a>	<a href="#"><code>getTimedTextProperties()</code></a> Get the property collection of timed text settings that are specific to this application instance.
String	<a href="#"><code>getTimeRunning()</code></a> Get time applicationInstance running
double	<a href="#"><code>getTimeRunningSeconds()</code></a> Get time running in seconds
<a href="#"><u>LiveStreamTranscoderApplicationContext</u></a>	<a href="#"><code>getTranscoderApplicationContext()</code></a> Get live stream transcoder application context
<a href="#"><u>WMSProperties</u></a>	<a href="#"><code>getTranscoderProperties()</code></a> Get the property collection of Transcoder settings that are specific to this application instance

int	<a href="#"><code>getValidationFrequency()</code></a> Get time between validation pings (milliseconds)
<a href="#"><code>IVHost</code></a>	<a href="#"><code>getVHost()</code></a> Get parent vHost
String	<a href="#"><code>getVODTimedTextProviderList()</code></a> Get the comma separated list of VODTimedTextProvider names being used by this application (see conf/TimedTextProviders.xml)
<code>java.util.List</code>	<a href="#"><code>getVODTimedTextProviderSet()</code></a>
void	<a href="#"><code>incClientCountTotal()</code></a> Increment the total number of connected client counter by one
boolean	<a href="#"><code>isAcceptConnection()</code></a> Is auto accept connection on/off
boolean	<a href="#"><code>isDebugEnabled()</code></a> If true appTimeout processing will be logged.
boolean	<a href="#"><code>isMediaCacheSourceAllowed(String sourceName)</code></a> Test to see if a MediaCache source name is allowed
boolean	<a href="#"><code>isRTPIncomingDatagramPortValid(int port)</code></a> Check a port number to be sure it is a valid RTP UDP port for this application instance
boolean	<a href="#"><code>isValidatingFMLEConnections()</code></a> Returns true if validating FMLE connection (default is false)
void	<a href="#"><code>notifyDvrStreamManagerCreate(IDvrStreamManager dvrStoreManager, String streamName)</code></a> Notify listeners that Dvr Application Store Manager has been created.
void	<a href="#"><code>notifyDvrStreamManagerDestroy(IDvrStreamManager dvrManager)</code></a> Notify listeners that Dvr Application Store Manager has been destroyed.
void	<a href="#"><code>notifyDvrStreamManagerInit(IDvrStreamManager dvrStoreManager)</code></a> Notify listeners that Dvr Application Store Manager has been initialized.
void	<a href="#"><code>notifyHTTPSessionCreate(IHTTPStreamerSession httpSession)</code></a> Notify HTTP session create
void	<a href="#"><code>notifyHTTPSessionDestroy(IHTTPStreamerSession httpSession)</code></a> Notify HTTP session destroy
void	<a href="#"><code>notifyLiveStreamDvrRecorderCreate(ILiveStreamDvrRecorder dvr, String streamName)</code></a> Notify Dvr Recorder Create
void	<a href="#"><code>notifyLiveStreamDvrRecorderDestroy(ILiveStreamDvrRecorder dvr)</code></a> Notify DVR Recorder has been destroyed.
void	<a href="#"><code>notifyLiveStreamDvrRecorderInit(ILiveStreamDvrRecorder dvr, String streamName)</code></a> Notify DVR Recorder has been initialized.
void	<a href="#"><code>notifyLiveStreamPacketizerCreate(ILiveStreamPacketizer liveStreamPacketizer, String streamName)</code></a> Notify Live Stream Packetizer Create

void	<a href="#"><code>notifyLiveStreamPacketizerDestroy(ILiveStreamPacketizer liveStreamPacketizer)</code></a> Notify Live Stream Packetizer Destory
void	<a href="#"><code>notifyLiveStreamPacketizerInit(ILiveStreamPacketizer liveStreamPacketizer, String streamName)</code></a> Notify Live Stream Packetizer Init
void	<a href="#"><code>notifyLiveStreamTranscoderCreate(ILiveStreamTranscoder liveStreamTranscoder, IMediaStream stream)</code></a> Notify live stream transcoder create
void	<a href="#"><code>notifyLiveStreamTranscoderDestroy(ILiveStreamTranscoder liveStreamTranscoder, IMediaStream stream)</code></a> Notify live stream transcoder destroy
void	<a href="#"><code>notifyLiveStreamTranscoderInit(ILiveStreamTranscoder liveStreamTranscoder, IMediaStream stream)</code></a> Notify live stream transcoder init
void	<a href="#"><code>notifyMediaReaderClose(IMediaReader mediaReader, IMediaStream stream)</code></a> Notify media reader notifyMediaReaderClose
void	<a href="#"><code>notifyMediaReaderCreate(IMediaReader mediaReader)</code></a> Notify media reader notifyMediaReaderCreate
void	<a href="#"><code>notifyMediaReaderExtractMetaData(IMediaReader mediaReader, IMediaStream stream)</code></a> Notify media reader notifyMediaReaderExtractMetaData
void	<a href="#"><code>notifyMediaReaderInit(IMediaReader mediaReader, IMediaStream stream)</code></a> Notify media reader notifyMediaReaderInit
void	<a href="#"><code>notifyMediaReaderOpen(IMediaReader mediaReader, IMediaStream stream)</code></a> Notify media reader notifyMediaReaderOpen
void	<a href="#"><code>notifyMediaWriterOnFLVAddMetadata(IMediaStream stream, java.util.Map extraMetadata)</code></a> Notify all MediaWriter listeners of onFLVAddMetadata
void	<a href="#"><code>notifyMediaWriterOnWriteComplete(IMediaStream stream, java.io.File file)</code></a> Notify all MediaWriter listeners of onWriteComplete
void	<a href="#"><code>notifyPushPublishSessionCreate(String streamName, IMediaStream stream, IPushPublishSession pushPublishSession)</code></a> Notify push publish session create
void	<a href="#"><code>notifyPushPublishSessionDestroy(String streamName, IMediaStream stream, IPushPublishSession pushPublishSession)</code></a> Notify push publish session destroy
void	<a href="#"><code>notifyRTPSessionCreate(RTPSession rtpSession)</code></a> Notify RTP session create
void	<a href="#"><code>notifyRTPSessionDestroy(RTPSession rtpSession)</code></a> Notify RTP session destory
void	<a href="#"><code>parseAllowDomains(String domainFilterStr)</code></a> Parse a comma delimited list of domain names used to control access to this application.

String	<a href="#"><code>readAppInstConfig(String sName)</code></a> Method to read xml config file..
void	<a href="#"><code>registerPlayRTPSession(RTPSession rtpSession)</code></a> Register an RTP session as a play session
void	<a href="#"><code>removeClientListener(IClientNotify clientListener)</code></a> Remove client listener.
void	<a href="#"><code>removeDvrRecorderListener(ILiveStreamDvrRecorderActionNotify listener)</code></a> Remove a Dvr Recorder listener (see: ILiveStreamDvrRecorderActionNotify)
void	<a href="#"><code>removeDvrStreamManagerListener(IDvrStreamManagerActionNotify listener)</code></a> Remove a Dvr Application Store Manager listener (see: IDvrStoreActionNotify)
void	<a href="#"><code>removeHTTPListener(IHTTPSessionNotify httpListener)</code></a> Remove listner to be notified of HTTPSession create and destroy.
void	<a href="#"><code>removeHTTPStreamerSession(IHTTPStreamerSession httpStreamerSession)</code></a> Remove a HTTPStreamerSession from this application instance
void	<a href="#"><code>removeLiveStreamPacketizerListener(ILiveStreamPacketizerActionNotify liveStreamPacketizerListener)</code></a> Remove a Live Stream Packetizer listener (see: ILiveStreamPacketizerActionNotify)
void	<a href="#"><code>removeLiveStreamTranscoderListener(ILiveStreamTranscoderNotify liveStreamTranscoderListener)</code></a> Remove a live stream transcoder listener
void	<a href="#"><code>removeMediaCasterListener(IMediaCasterNotify mediaCasterListener)</code></a> Remove mediaCaster listener.
void	<a href="#"><code>removeMediaReaderListener(IMediaReaderActionNotify mediaReaderListener)</code></a> Remove media reader listener.
void	<a href="#"><code>removeMediaStreamListener(IMediaStreamNotify mediaStreamListener)</code></a> Remove mediaStream listener.
void	<a href="#"><code>removeMediaWriterListener(IMediaWriterActionNotify listener)</code></a> remove MediaWriter listener class.
void	<a href="#"><code>removeModuleListener(IModuleNotify moduleListener)</code></a> Remove module listener
void	<a href="#"><code>removePlayStreamByName(IMediaStream stream)</code></a> Remove media stream from the list of streams that are listening for a published stream
void	<a href="#"><code>removePublisher(Publisher publisher)</code></a> Remove a server side publisher from this application instance
void	<a href="#"><code>removePushPublishSession(IMediaStream stream, IPushPublishSession pushPublishSession)</code></a> Remove a push publish session
void	<a href="#"><code>removePushPublishSessionListener(IPushPublishSessionNotify pushPublishSessionListener)</code></a> Remove a push publish session listener (see IPushPublishSessionNotify)
void	<a href="#"><code>removePushPublishSessions(IMediaStream stream)</code></a> Remove all push publish sessions for a give stream

void	<a href="#"><code>removeRTPListener(I RTPSessionNotify rtpListener)</code></a> Remove listner to be notified of RTPSession create and destroy.
void	<a href="#"><code>removeRTPSession(RTPSession rtpSession)</code></a> Remove an RTP session from this application instance
void	<a href="#"><code>removeSharedObjectListener( ISharedObjectNotify sharedObjectListener, boolean isPersistent)</code></a> Remove sharedObject listener.
boolean	<a href="#"><code>resetMediaCasterStream(String streamName)</code></a> Reset a media caster stream
boolean	<a href="#"><code>resetMediaCasterStream(String streamName, String streamExt)</code></a> Reset a media caster stream
void	<a href="#"><code>setAcceptConnection(boolean acceptConnection)</code></a> Set is auto accept connection
void	<a href="#"><code>setAllowDomains(String[] domainFilter)</code></a> Set the list of domain names used to control access to this application.
void	<a href="#"><code>setApplicationInstanceTouchTimeout(int applicationInstanceTouchTimeout)</code></a> Set the application instance touch timeout (milliseconds).
void	<a href="#"><code>setApplicationTimeout(int applicationTimeout)</code></a> Set application timeout (milliseconds)
void	<a href="#"><code>setClientIdleFrequency(int clientIdleFrequency)</code></a> Set default client idle frequency (milliseconds)
void	<a href="#"><code>setDebugAppTimeout(boolean debugAppTimeout)</code></a> If true appTimeout processing will be logged.
void	<a href="#"><code>setDvrRecorderList(String recorderList)</code></a> Set the comma separated list of Dvr Recorder names being used by this application (see conf/Dvr.xml)
void	<a href="#"><code>setHTTPStreamerList(String httpStreamerList)</code></a> Set the comma separated list of HTTPStreamer names being used by this application (see conf/HTTPStreamers.xml)
void	<a href="#"><code>setLiveStreamDvrRecorderControl(ILiveStreamDvrRecorderControl controller)</code></a> Set the Live Stream DVR Controller.
void	<a href="#"><code>setLiveStreamPacketizerControl(ILiveStreamPacketizerControl liveStreamPacketizerControl)</code></a> Set the Live Stream Packetizer Contoller.
void	<a href="#"><code>setLiveStreamPacketizerList(String liveStreamPacketizerList)</code></a> Set the comma separated list of LiveStreamPacketizers names being used by this application (see conf/LiveStreamPacketizers.xml)
void	<a href="#"><code>setLiveStreamTranscoderControl(ILiveStreamTranscoderControl liveStreamTranscoderControl)</code></a> Set the Live Stream Transcoder Contoller.
void	<a href="#"><code>setLiveStreamTranscoderList(String liveStreamTranscoderList)</code></a> 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>setMediacasterRTPRTSPRTPTTransportMode</u></a> (int mediacasterRTPRTSPRTPTTransportMode) RTP MediaCaster RTSP/RTP transport mode.
void	<a href="#"><u>setMediaCasterValidator</u></a> (IMediaCasterValidateMediaCaster mediaCasterValidator) Set the MediaCaster validator interface for this application instance
void	<a href="#"><u>setMediaListProvider</u></a> (IMediaListProvider 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="#"><code>setValidationFrequency(int validationFrequency)</code></a> Set time between validation pings (milliseconds)
void	<a href="#"><code>setVODTimedTextProviderList(String timedTextProviderList)</code></a> Set the comma separated list of VODTimedTextProvider names being used by this application (see conf/TimedTextProviders.xml)
void	<a href="#"><code>shutdown(boolean isServerShutdown, boolean isAppShutdown)</code></a> shutdown applicationInstance
void	<a href="#"><code>shutdownClient(IClient client)</code></a> shutdown a client connection immediately
boolean	<a href="#"><code>startMediaCasterStream(String streamName, String mediaCasterType)</code></a> Start a media caster stream
boolean	<a href="#"><code>startMediaCasterStream(String streamName, String streamExt, String mediaCasterType)</code></a> Start a media caster stream
void	<a href="#"><code>stopMediaCasterStream(String streamName)</code></a> Stop a media caster stream
void	<a href="#"><code>touch()</code></a> Touch the application instance so that it stays loaded for at least applicationInstanceTouchTimeout
boolean	<a href="#"><code>writeAppInstConfig(String sName, String data)</code></a> Method to write xml config file..

## Fields

### 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
```

(continued from last page)

Constant value: **3**

---

**PROTOCOLUSAGE\_RTMPE**

```
public static final int PROTOCOLUSAGE_RTMPE
```

Constant value: **4**

---

**PROTOCOLUSAGE\_RTMPTE**

```
public static final int PROTOCOLUSAGE_RTMPTE
```

Constant value: **5**

---

**PROTOCOLUSAGE\_RTP**

```
public static final int PROTOCOLUSAGE_RTP
```

Constant value: **6**

---

**PROTOCOLUSAGE\_CUPERTINO**

```
public static final int PROTOCOLUSAGE_CUPERTINO
```

Constant value: **7**

---

**PROTOCOLUSAGE\_SMOOTH**

```
public static final int PROTOCOLUSAGE_SMOOTH
```

Constant value: **8**

---

**PROTOCOLUSAGE\_SANJOSE**

```
public static final int PROTOCOLUSAGE_SANJOSE
```

Constant value: **9**

---

**PROTOCOLUSAGE\_MPEGDASH**

```
public static final int PROTOCOLUSAGE_MPEGDASH
```

Constant value: **10**

---

**PROTOCOLUSAGE\_WEBM**

```
public static final int PROTOCOLUSAGE_WEBM
```

Constant value: **11**

(continued from last page)

## PROTOCOLUSAGE\_WOWZ

```
public static final int PROTOCOLUSAGE_WOWZ
```

Constant value: **12**

## PROTOCOLUSAGE\_WOWZE

```
public static final int PROTOCOLUSAGE_WOWZE
```

Constant value: **13**

## PROTOCOLUSAGE\_WOWZS

```
public static final int PROTOCOLUSAGE_WOWZS
```

Constant value: **14**

## PROTOCOLUSAGE\_ORIGINCUPERTINO

```
public static final int PROTOCOLUSAGE_ORIGINCUPERTINO
```

Constant value: **15**

## PROTOCOLUSAGE\_ORIGINSMOOTH

```
public static final int PROTOCOLUSAGE_ORIGINSMOOTH
```

Constant value: **16**

## PROTOCOLUSAGE\_ORIGINSANJOSE

```
public static final int PROTOCOLUSAGE_ORIGINSANJOSE
```

Constant value: **17**

## PROTOCOLUSAGE\_ORIGINMPEGDASH

```
public static final int PROTOCOLUSAGE_ORIGINMPEGDASH
```

Constant value: **18**

## PROTOCOLUSAGE\_PUSHUBLISHING

```
public static final int PROTOCOLUSAGE_PUSHUBLISHING
```

Constant value: **19**

## PROTOCOLUSAGE\_MEDIACACHE

```
public static final int PROTOCOLUSAGE_MEDIACACHE
```

---

(continued from last page)

Constant value: **20**

---

### **PROTOCOLUSAGE\_RTMP\_IN**

```
public static final int PROTOCOLUSAGE_RTMP_IN
```

Constant value: **21**

---

### **PROTOCOLUSAGE\_RTMP\_OUT**

```
public static final int PROTOCOLUSAGE_RTMP_OUT
```

Constant value: **22**

---

### **PROTOCOLUSAGE\_RTMPT\_IN**

```
public static final int PROTOCOLUSAGE_RTMPT_IN
```

Constant value: **23**

---

### **PROTOCOLUSAGE\_RTMPT\_OUT**

```
public static final int PROTOCOLUSAGE_RTMPT_OUT
```

Constant value: **24**

---

### **PROTOCOLUSAGE\_RTMPS\_IN**

```
public static final int PROTOCOLUSAGE_RTMPS_IN
```

Constant value: **25**

---

### **PROTOCOLUSAGE\_RTMPS\_OUT**

```
public static final int PROTOCOLUSAGE_RTMPS_OUT
```

Constant value: **26**

---

### **PROTOCOLUSAGE\_RTMPTS\_IN**

```
public static final int PROTOCOLUSAGE_RTMPTS_IN
```

Constant value: **27**

---

### **PROTOCOLUSAGE\_RTMPTS\_OUT**

```
public static final int PROTOCOLUSAGE_RTMPTS_OUT
```

Constant value: **28**

---

---

**PROTOCOLUSAGE\_RTMPE\_IN**

```
public static final int PROTOCOLUSAGE_RTMPE_IN
```

Constant value: **29**

---

**PROTOCOLUSAGE\_RTMPE\_OUT**

```
public static final int PROTOCOLUSAGE_RTMPE_OUT
```

Constant value: **30**

---

**PROTOCOLUSAGE\_RTMPTE\_IN**

```
public static final int PROTOCOLUSAGE_RTMPTE_IN
```

Constant value: **31**

---

**PROTOCOLUSAGE\_RTMPTE\_OUT**

```
public static final int PROTOCOLUSAGE_RTMPTE_OUT
```

Constant value: **32**

---

**PROTOCOLUSAGE\_RTP\_IN**

```
public static final int PROTOCOLUSAGE_RTP_IN
```

Constant value: **33**

---

**PROTOCOLUSAGE\_RTP\_OUT**

```
public static final int PROTOCOLUSAGE_RTP_OUT
```

Constant value: **34**

---

**PROTOCOLUSAGE\_RTSP\_IN**

```
public static final int PROTOCOLUSAGE_RTSP_IN
```

Constant value: **35**

---

**PROTOCOLUSAGE\_RTSP\_OUT**

```
public static final int PROTOCOLUSAGE_RTSP_OUT
```

Constant value: **36**

---

(continued from last page)

## PROTOCOLUSAGE\_WOWZ\_IN

```
public static final int PROTOCOLUSAGE_WOWZ_IN
```

Constant value: **37**

## PROTOCOLUSAGE\_WOWZ\_OUT

```
public static final int PROTOCOLUSAGE_WOWZ_OUT
```

Constant value: **38**

## PROTOCOLUSAGE\_WOWZE\_IN

```
public static final int PROTOCOLUSAGE_WOWZE_IN
```

Constant value: **39**

## PROTOCOLUSAGE\_WOWZE\_OUT

```
public static final int PROTOCOLUSAGE_WOWZE_OUT
```

Constant value: **40**

## PROTOCOLUSAGE\_WOWZS\_IN

```
public static final int PROTOCOLUSAGE_WOWZS_IN
```

Constant value: **41**

## PROTOCOLUSAGE\_WOWZS\_OUT

```
public static final int PROTOCOLUSAGE_WOWZS_OUT
```

Constant value: **42**

## PROTOCOLUSAGE\_MC\_SHOUTCAST

```
public static final int PROTOCOLUSAGE_MC_SHOUTCAST
```

Constant value: **43**

## PROTOCOLUSAGE\_MC\_LIVEREPEATER

```
public static final int PROTOCOLUSAGE_MC_LIVEREPEATER
```

Constant value: **44**

## PROTOCOLUSAGE\_MC\_RTP

```
public static final int PROTOCOLUSAGE_MC_RTP
```

(continued from last page)

Constant value: **45**

---

## PROTOCOLUSAGE\_TOTAL

```
public static final int PROTOCOLUSAGE_TOTAL
```

Constant value: **46**

---

## 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

---

### getApplication

```
public IApplication getApplication()
```

Get parent application

**Returns:**

parent application

---

### getName

```
public String getName()
```

Get applicationInstance name

**Returns:**

applicationInstance name

(continued from last page)

## 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

**Returns:**

parent vHost

---

## getProperties

```
public WMSProperties getProperties()
```

Get applicationInstance properties

**Returns:**

applicationInstance properties

---

## getManagerProperties

```
public WMSProperties getManagerProperties()
```

Get Application's manager's properties collection

(continued from last page)

**Returns:**

manager's property collection

**getStreamType****public String getStreamType( )**

Get default streamType for application.

**Returns:**

streamType name

**setStreamType****public void setStreamType(String streamType)**

Set default stream type for application.

**Parameters:**

streamType - streamType name

**isAcceptConnection****public boolean isAcceptConnection()**

Is auto accept connection on/off

**Returns:**

auto accept connection

**setAcceptConnection****public void setAcceptConnection(boolean acceptConnection)**

Set is auto accept connection

**Parameters:**

acceptConnection - auto accept connection

**getClientCountTotal****public int getClientCountTotal()**

Get number of client connections in total that have connected to this applicationInstance

**Returns:**

number of client connections

**incClientCountTotal****public void incClientCountTotal()**

Increment the total number of connected client counter by one

**getClientCount****public int getClientCount()**

(continued from last page)

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

(continued from last page)

**Returns:**

collection of shared objects

**addClientListener****public void addClientListener([IClientNotify](#) clientListener)**

Add client listener. Listens for connects, disconnect, accepts and reject

**Add a Client Listener**

```
IAplicationInstance 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

(continued from last page)

## 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

```
IAplicationInstance 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)
```

Add sharedObject listener. Listens for create, destroy, clientConnect, clientDisconnect

### Add SharedObject Listener

```
IAplicationInstance 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

(continued from last page)

## removeSharedObjectListener

```
public void removeSharedObjectListener(ISharedObjectNotify sharedObjectListener,  
                                     boolean isPersistent)
```

Remove sharedObject listener. Listens for create, destroy, clientConnect, clientDisconnect

**Parameters:**

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:**

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)

---

## getRTPIIdleFrequency

```
public int getRTPIIdleFrequency()
```

Set the default RTP idle frequency (milliseconds)

**Returns:**

default RTP idle frequency (milliseconds)

---

## setRTPIIdleFrequency

```
public void setRTPIIdleFrequency(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 writer settings that are specific to this application instance

**Returns:**

property collection of media writer settings

---

## getRTTProperties

```
public WMSProperties getRTTProperties()
```

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 if this type of streaming is allowed

---

**containsLiveStreamPacketizer****public boolean containsLiveStreamPacketizer(String liveStreamPacketizer)**

Does this application instance contain a references to this live stream packetizer. If it is true we consider this a live stream source for the HTTP streamer. If false then we consider this a video on demand source.

**Parameters:**

liveStreamPacketizer - live stream packetizer name

**Returns:**

true if contains reference to it

---

**containsDvrRecorder****public boolean containsDvrRecorder(String dvrRecorder)**

Does this application instance contain a references to this DVR recorder. If it is true we consider this a DVR source for the HTTP streamer.

**Parameters:**

dvrRecorder - DVR recorder name

**Returns:**

true if contains reference to it

---

**getVODTimedTextProviderList****public String getVODTimedTextProviderList()**

Get the comma separated list of VODTimedTextProvider names being used by this application (see conf/TimedTextProviders.xml)

**Returns:**

comma separated list of VODTimedTextProvider names

---

**setVODTimedTextProviderList****public void setVODTimedTextProviderList(String timedTextProviderList)**

Set the comma separated list of VODTimedTextProvider names being used by this application (see conf/TimedTextProviders.xml)

**Parameters:**

timedTextProviderList - comma separated list of VODTimedTextProvider names

---

**getVODTimedTextProviderSet****public java.util.List getVODTimedTextProviderSet()**

(continued from last page)

## getTimedTextProperties

```
public WMSProperties getTimedTextProperties()
```

Get the property collection of timed text settings that are specific to this application instance. These are defined in Application/TimedText/Properties tag in Application.xml

**Returns:**

property collection of Timed Text settings

---

## getStreamNameAliasProvider

```
public IMediaStreamNameAliasProvider getStreamNameAliasProvider()
```

Get the stream name alias provider

**Returns:**

stream name alias provider

---

## setStreamNameAliasProvider

```
public void setStreamNameAliasProvider(IMediaStreamNameAliasProvider streamNameAliasProvider)
```

Set the stream name alias provider

**Parameters:**

streamNameAliasProvider - stream name alias provider

---

## getPublishers

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

Get the set of server side publishers

**Returns:**

set of server side publishers

---

## getPublisherCount

```
public int getPublisherCount()
```

Get the current number of server side publishers

**Returns:**

number of server side publishers

---

## addPublisher

```
public void addPublisher(Publisher publisher)
```

Add a server side publisher to this application instance

**Parameters:**

publisher - server side publisher

(continued from last page)

## removePublisher

```
public void removePublisher(Publisher publisher)
```

Remove a server side publisher from this application instance

**Parameters:**

publisher - server side publisher

---

## getHTTPStreamerSessions

```
public java.util.List getHTTPStreamerSessions(int protocol,  
String streamName)
```

Get the HTTPStreamerSessions associated with this application instance for a stream name by protocol. See ([IHTTPStreamerSession.SESSIONPROTOCOL\\_\\*](#)) for protocols

**Parameters:**

protocol - streaming protocol ([IHTTPStreamerSession.SESSIONPROTOCOL\\_\\*](#))  
streamName - stream name

**Returns:**

HTTPStreamerSessions associated with this application instance

---

## getHTTPStreamerSessions

```
public java.util.List getHTTPStreamerSessions(String streamName)
```

Get the HTTPStreamerSessions associated with this application instance for a stream name

**Parameters:**

streamName - stream name

**Returns:**

HTTPStreamerSessions associated with this application instance

---

## getHTTPStreamerSessionCountsByName

```
public java.util.Map getHTTPStreamerSessionCountsByName(int protocol)
```

Get a map of session counts by name for a given protocol

**Parameters:**

protocol - streaming protocol ([IHTTPStreamerSession.SESSIONPROTOCOL\\_\\*](#))

**Returns:**

map of session counts by name

---

## getHTTPStreamerSessionCount

```
public int getHTTPStreamerSessionCount(String streamName)
```

Get the current number of HTTPStreamerSessions associated with this application instance and stream name

**Parameters:**

streamName - stream name

**Returns:**

number of HTTPStreamerSessions associated with this application instance

## getHTTPStreamerSessionCount

```
public int getHTTPStreamerSessionCount(int protocol,  
String streamName)
```

Get the current number of HTTPStreamerSessions associated with this application instance and stream name by protocol . See (IHTTPStreamerSession.SESSIONPROTOCOL\_\*) for protocols

**Parameters:**

protocol - streaming protocol (IHTTPStreamerSession.SESSIONPROTOCOL\_\*)  
streamName - stream name

**Returns:**

HTTPStreamerSessions associated with this application instance

## getHTTPStreamerSessions

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

Get the HTTPStreamerSessions associated with this application instance

**Returns:**

HTTPStreamerSessions associated with this application instance

## getHTTPStreamerSessions

```
public java.util.List getHTTPStreamerSessions(int protocol)
```

Get the HTTPStreamerSessions associated with this application instance by protocol. See (IHTTPStreamerSession.SESSIONPROTOCOL\_\*) for protocols

**Parameters:**

protocol - streaming protocol (IHTTPStreamerSession.SESSIONPROTOCOL\_\*)

**Returns:**

HTTPStreamerSessions associated with this application instance

## getHTTPStreamerSessionCount

```
public int getHTTPStreamerSessionCount()
```

Get the current number of HTTPStreamerSessions associated with this application instance

**Returns:**

current number of HTTPStreamerSessions associated with this application instance

## getHTTPStreamerSessionCount

```
public int getHTTPStreamerSessionCount(int protocol)
```

Get the current number of HTTPStreamerSessions associated with this application instance by protocol. See (IHTTPStreamerSession.SESSIONPROTOCOL\_\*) for protocols

**Parameters:**

protocol - streaming protocol (IHTTPStreamerSession.SESSIONPROTOCOL\_\*)

**Returns:**

current number of HTTPStreamerSessions associated with this application instance

---

## addHTTPStreamerSession

```
public void addHTTPStreamerSession(IHTTPStreamerSession httpStreamerSession)
```

Add a HTTPStreamerSession to this application instance

**Parameters:**

httpStreamerSession - HTTPStreamerSession

---

## removeHTTPStreamerSession

```
public void removeHTTPStreamerSession(IHTTPStreamerSession httpStreamerSession)
```

Remove a HTTPStreamerSession from this application instance

**Parameters:**

httpStreamerSession - HTTPStreamerSession

---

## getHTTPStreamerApplicationContext

```
public IHTTPStreamerApplicationContext getHTTPStreamerApplicationContext(String httpStreamName,  
boolean doCreate)
```

Get the HTTPStreamer application context for a given HTTPStreamer adapter

**Parameters:**

httpStreamName - HTTPStreamer adapter name  
doCreate - create if it does not exist

**Returns:**

HTTPStreamer application context

---

## addRTPIncomingDatagramPortRange

```
public void addRTPIncomingDatagramPortRange(int startPort,  
int endPort)
```

Add a port range to the list of valid incoming RTP UDP ports

**Parameters:**

startPort - starting port number  
endPort - end port number

---

## addRTPIncomingDatagramPortAll

```
public void addRTPIncomingDatagramPortAll()
```

Allow all incoming RTP UDP ports for this application instance

---

## isRTPIncomingDatagramPortValid

```
public boolean isRTPIncomingDatagramPortValid(int port)
```

Check a port number to be sure it is a valid RTP UDP port for this application instance

**Parameters:**

port - port number

(continued from last page)

**Returns:**

true if the port is valid

**readAppInstConfig**

```
public String readAppInstConfig(String sName)
```

Method to read xml config file..

**writeAppInstConfig**

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

Method to write xml config file..

**getLiveStreamPacketizerControl**

```
public ILiveStreamPacketizerControl getLiveStreamPacketizerControl()
```

Get the Live Stream Packetizer Controller. This class will get called each time a stream is to be packetized using the LiveStreamPacketizer mechanism.

**Returns:**

Live Stream Packetizer Controller

**setLiveStreamPacketizerControl**

```
public void setLiveStreamPacketizerControl(ILiveStreamPacketizerControl  
liveStreamPacketizerControl)
```

Set the Live Stream Packetizer Controller. This class will get called each time a stream is to be packetized using the LiveStreamPacketizer mechanism.

**Parameters:**

liveStreamPacketizerControl - Live Stream Packetizer Controller

**resetMediaCasterStream**

```
public boolean resetMediaCasterStream(String streamName)
```

Reset a media caster stream

**Parameters:**

streamName - stream name

**Returns:**

true if successful

**resetMediaCasterStream**

```
public boolean resetMediaCasterStream(String streamName,  
String streamExt)
```

Reset a media caster stream

**Parameters:**

streamName - stream name

(continued from last page)

streamExt - stream extension

**Returns:**

true if successful

---

**startMediaCasterStream**

```
public boolean startMediaCasterStream(String streamName,  
        String streamExt,  
        String mediaCasterType)
```

Start a media caster stream

**Parameters:**

streamName - stream name  
streamExt - stream extension  
mediaCasterType - media caster stream type

**Returns:**

true if successful

---

**startMediaCasterStream**

```
public boolean startMediaCasterStream(String streamName,  
        String mediaCasterType)
```

Start a media caster stream

**Parameters:**

streamName - stream name  
mediaCasterType - media caster stream type

**Returns:**

true if successful

---

**stopMediaCasterStream**

```
public void stopMediaCasterStream(String streamName)
```

Stop a media caster stream

**Parameters:**

streamName - stream name

---

**getContextStr**

```
public String getContextStr()
```

Returns the application context string in the form [application]/[appInstance].

**Returns:**

application context string

---

**getPublishStreamNames**

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

Get the list of live stream names currently being published.

---

(continued from last page)

**Returns:**

list of live stream names currently being published

---

**addMediaWriterListener****public void addMediaWriterListener([IMediaWriterActionNotify](#) listener)**

Add a MediaWriter listener class. See IMediaWriterActionNotify

**Parameters:**

listener - MediaWriter listener class

---

**removeMediaWriterListener****public void removeMediaWriterListener([IMediaWriterActionNotify](#) listener)**

remove MediaWriter listener class. See IMediaWriterActionNotify

**Parameters:**

listener - MediaWriter listener class

---

**notifyMediaWriterOnWriteComplete****public void notifyMediaWriterOnWriteComplete([IMediaStream](#) stream,  
java.io.File file)**

Notify all MediaWriter listeners of onWriteComplete

**Parameters:**

stream - media stream

file - file that was written

---

**notifyMediaWriterOnFLVAddMetadata****public void notifyMediaWriterOnFLVAddMetadata([IMediaStream](#) stream,  
java.util.Map extraMetadata)**

Notify all MediaWriter listeners of onFLVAddMetadata

**Parameters:**

stream - media stream

extraMetadata - meta to add to the file

---

**getMediaCasterValidator****public [IMediaCasterValidateMediaCaster](#) getMediaCasterValidator()**

Get the MediaCaster validator interface for this application instance

**Returns:**

MediaCaster validator interface

---

**setMediaCasterValidator****public void setMediaCasterValidator([IMediaCasterValidateMediaCaster](#)  
mediaCasterValidator)**

Set the MediaCaster validator interface for this application instance

(continued from last page)

**Parameters:**

mediaCasterValidator - MediaCaster validator interface

**touch****public void touch()**

Touch the application instance so that it stays loaded for at least applicationInstanceTouchTimeout

**getLastTouchTime****public long getLastTouchTime()**

Get the last time the instance was touched (milliseconds)

**Returns:**

last time the instance was touched (milliseconds)

**getApplicationInstanceTouchTimeout****public int getApplicationInstanceTouchTimeout()**

Get the application instance touch timeout (milliseconds). Default is 5000.

**Returns:**

application instance touch timeout (milliseconds)

**setApplicationInstanceTouchTimeout****public void setApplicationInstanceTouchTimeout(int applicationInstanceTouchTimeout)**

Set the application instance touch timeout (milliseconds). Default is 5000.

**Parameters:**

applicationInstanceTouchTimeout - application instance touch timeout (milliseconds)

**getRTSPSessionTimeout****public int getRTSPSessionTimeout()**

Get the RTSP session timeout (milliseconds)

**Returns:**

RTSP session timeout (milliseconds)

**setRTSPSessionTimeout****public void setRTSPSessionTimeout(int rtspSessionTimeout)**

Set the RTSP session timeout (milliseconds)

**Parameters:**

rtspSessionTimeout - RTSP session timeout (milliseconds)

**getRTSPMaximumPendingWriteBytes****public int getRTSPMaximumPendingWriteBytes()**

(continued from last page)

Get the maximum number of pending write bytes for an RTSP session

**Returns:**

maximum number of pending write bytes for an RTSP session

---

**setRTSPMaximumPendingWriteBytes**

```
public void setRTSPMaximumPendingWriteBytes(int rtspMaximumPendingWriteBytes)
```

Set the maximum number of pending write bytes for an RTSP session

**Parameters:**

rtspMaximumPendingWriteBytes - maximum number of pending write bytes for an RTSP session

---

**addMediaReaderListener**

```
public void addMediaReaderListener(IMediaReaderActionNotify mediaReaderListener)
```

Add media reader listener. see IMediaReaderActionNotify

**Parameters:**

mediaReaderListener - media reader listener

---

**removeMediaReaderListener**

```
public void removeMediaReaderListener(IMediaReaderActionNotify mediaReaderListener)
```

Remove media reader listener. see IMediaReaderActionNotify

**Parameters:**

mediaReaderListener - media reader listener

---

**notifyMediaReaderCreate**

```
public void notifyMediaReaderCreate(IMediaReader mediaReader)
```

Notify media reader notifyMediaReaderCreate

**Parameters:**

mediaReader - media reader

---

**notifyMediaReaderInit**

```
public void notifyMediaReaderInit(IMediaReader mediaReader,  
                         IMediaStream stream)
```

Notify media reader notifyMediaReaderInit

**Parameters:**

mediaReader - media reader

stream - media stream

---

**notifyMediaReaderOpen**

```
public void notifyMediaReaderOpen(IMediaReader mediaReader,  
                         IMediaStream stream)
```

Notify media reader notifyMediaReaderOpen

(continued from last page)

**Parameters:**

mediaReader - media reader  
stream - media stream

---

**notifyMediaReaderExtractMetaData**

```
public void notifyMediaReaderExtractMetaData(IMediaReader mediaReader,  
          IMediaStream stream)
```

Notify media reader notifyMediaReaderExtractMetaData

**Parameters:**

mediaReader - media reader  
stream - media stream

---

**notifyMediaReaderClose**

```
public void notifyMediaReaderClose(IMediaReader mediaReader,  
          IMediaStream stream)
```

Notify media reader notifyMediaReaderClose

**Parameters:**

mediaReader - media reader  
stream - media stream

---

**getRTSPBindIpAddress**

```
public String getRTSPBindIpAddress()
```

Set the IP address to which UDP ports will be bound for RTSP/RTP sessions

**Returns:**

IP address to which UDP ports will be bound for RTSP/RTP sessions

---

**setRTSPBindIpAddress**

```
public void setRTSPBindIpAddress(String rtspBindIpAddress)
```

Get the IP address to which UDP ports will be bound for RTSP/RTP sessions

**Parameters:**

rtspBindIpAddress - IP address to which UDP ports will be bound for RTSP/RTP sessions

---

**getRTSPConnectionIpAddress**

```
public String getRTSPConnectionIpAddress()
```

Get the connection IP address to used in the Session Description Protocol data exchanged for an RTSP/RTP session

**Returns:**

connection IP address to used in the Session Description Protocol data exchanged for an RTSP/RTP session

---

**setRTSPConnectionIpAddress**

```
public void setRTSPConnectionIpAddress(String rtspConnectionIpAddress)
```

Set the connection IP address to used in the Session Description Protocol data exchanged for an RTSP/RTP session

(continued from last page)

**Parameters:**

`rtspConnectionIpAddress` - connection IP address to used in the Session Description Protocol data exchanged for an RTSP/RTP session

## getRTSPConnectionAddressType

```
public String getRTSPConnectionAddressType()
```

Get the connection IP address type (IP4) to used in the Session Description Protocol data exchanged for an RTSP/RTP session

**Returns:**

the connection IP address type (IP4) to used in the Session Description Protocol data exchanged for an RTSP/RTP session

## setRTSPConnectionAddressType

```
public void setRTSPConnectionAddressType(String rtspConnectionAddressType)
```

Set the connection IP address type (IP4) to used in the Session Description Protocol data exchanged for an RTSP/RTP session

**Parameters:**

`rtspConnectionAddressType`

## getRTSPOriginIpAddress

```
public String getRTSPOriginIpAddress()
```

Get the origin IP address to used in the Session Description Protocol data exchanged for an RTSP/RTP session

**Returns:**

origin IP address to used in the Session Description Protocol data exchanged for an RTSP/RTP session

## setRTSPOriginIpAddress

```
public void setRTSPOriginIpAddress(String rtspOriginIpAddress)
```

Set the origin IP address to used in the Session Description Protocol data exchanged for an RTSP/RTP session

**Parameters:**

`rtspOriginIpAddress` - origin IP address to used in the Session Description Protocol data exchanged for an RTSP/RTP session

## getRTSPOriginAddressType

```
public String getRTSPOriginAddressType()
```

Get the origin IP address type (IP4) to used in the Session Description Protocol data exchanged for an RTSP/RTP session

**Returns:**

origin IP address type (IP4) to used in the Session Description Protocol data exchanged for an RTSP/RTP session

## setRTSPOriginAddressType

```
public void setRTSPOriginAddressType(String rtspOriginAddressType)
```

Set the origin IP address type (IP4) to used in the Session Description Protocol data exchanged for an RTSP/RTP session

**Parameters:**

(continued from last page)

`rtspOriginAddressType` - origin IP address type (IP4) to used in the Session Description Protocol data exchanged for an RTSP/RTP session

## **addLiveStreamPacketizerListener**

```
public void addLiveStreamPacketizerListener(ILiveStreamPacketizerActionNotify  
liveStreamPacketizerListener)
```

Add a Live Stream Packetizer listener (see: `ILiveStreamPacketizerActionNotify`)

**Parameters:**

`liveStreamPacketizerListener` - Live Stream Packetizer listener

## **removeLiveStreamPacketizerListener**

```
public void removeLiveStreamPacketizerListener(ILiveStreamPacketizerActionNotify  
liveStreamPacketizerListener)
```

Remove a Live Stream Packetizer listener (see: `ILiveStreamPacketizerActionNotify`)

**Parameters:**

`liveStreamPacketizerListener` - Live Stream Packetizer listener

## **notifyLiveStreamPacketizerCreate**

```
public void notifyLiveStreamPacketizerCreate(ILiveStreamPacketizer  
liveStreamPacketizer,  
String streamName)
```

Notify Live Stream Packetizer Create

**Parameters:**

`liveStreamPacketizer` - Live Stream Packetizer listener

## **notifyLiveStreamPacketizerDestroy**

```
public void notifyLiveStreamPacketizerDestroy(ILiveStreamPacketizer  
liveStreamPacketizer)
```

Notify Live Stream Packetizer Destory

**Parameters:**

`liveStreamPacketizer` - Live Stream Packetizer listener

## **notifyLiveStreamPacketizerInit**

```
public void notifyLiveStreamPacketizerInit(ILiveStreamPacketizer liveStreamPacketizer,  
String streamName)
```

Notify Live Stream Packetizer Init

**Parameters:**

`liveStreamPacketizer` - Live Stream Packetizer listener

## **isValidFMLEConnections**

```
public boolean isValidFMLEConnections()
```

Returns true if validating FMLE connection (default is false)

(continued from last page)

**Returns:**

true if validating FMLE connection

---

**setValidateFMLEConnections**

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

Returns true if validating FMLE connection (default is false)

**Parameters:**

validateFMLEConnections - true if validating FMLE connection

---

**addLiveStreamTranscoderListener**

```
public void addLiveStreamTranscoderListener(ILiveStreamTranscoderNotify liveStreamTranscoderListener)
```

Add a live stream transcoder listener

**Parameters:**

liveStreamTranscoderListener - live stream transcoder listener

---

**removeLiveStreamTranscoderListener**

```
public void removeLiveStreamTranscoderListener(ILiveStreamTranscoderNotify liveStreamTranscoderListener)
```

Remove a live stream transcoder listener

**Parameters:**

liveStreamTranscoderListener - live stream transcoder listener

---

**notifyLiveStreamTranscoderCreate**

```
public void notifyLiveStreamTranscoderCreate(ILiveStreamTranscoder liveStreamTranscoder, IMediaStream stream)
```

Notify live stream transcoder create

**Parameters:**liveStreamTranscoder - live stream transcoder  
stream - stream

---

**notifyLiveStreamTranscoderDestroy**

```
public void notifyLiveStreamTranscoderDestroy(ILiveStreamTranscoder liveStreamTranscoder, IMediaStream stream)
```

Notify live stream transcoder destroy

**Parameters:**liveStreamTranscoder - live stream transcoder  
stream - stream

(continued from last page)

## notifyLiveStreamTranscoderInit

```
public void notifyLiveStreamTranscoderInit(ILiveStreamTranscoder liveStreamTranscoder,  
          IMediaStream stream)
```

Notify live stream transcoder init

**Parameters:**

liveStreamTranscoder - live stream transcoder  
stream - stream

---

## containsLiveStreamTranscoder

```
public boolean containsLiveStreamTranscoder(String liveStreamTranscoder)
```

Return true if this application instance contains the transcoder name

**Parameters:**

liveStreamTranscoder - transcoder name

**Returns:**

true if this application instance contains the transcoder name

---

## getLiveStreamTranscoderList

```
public String getLiveStreamTranscoderList()
```

Get comma separated list of transcoders to use for this application instance

**Returns:**

comma separated list of transcoders

---

## setLiveStreamTranscoderList

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

Set comma separated list of transcoders to use for this application instance

**Parameters:**

liveStreamTranscoderList - comma separated list of transcoders

---

## getLiveStreamTranscoderControl

```
public ILiveStreamTranscoderControl getLiveStreamTranscoderControl()
```

Get the Live Stream Transcoder Contoller. This class will get called each time a stream is to be transcoded using the LiveStreamTranscoder mechanism.

**Returns:**

Live Stream Transcoder Contoller

---

## setLiveStreamTranscoderControl

```
public void setLiveStreamTranscoderControl(ILiveStreamTranscoderControl  
          liveStreamTranscoderControl)
```

Set the Live Stream Transcoder Contoller. This class will get called each time a stream is to be transcoded using the LiveStreamTranscoder mechanism.

(continued from last page)

**Parameters:**

liveStreamTranscoderControl - Live Stream Transcoder Controller

---

**getTranscoderApplicationContext**

```
public LiveStreamTranscoderApplicationContext getTranscoderApplicationContext()
```

Get live stream transcoder application context

**Returns:**

live stream transcoder application context

---

**getDvrProperties**

```
public WMSProperties getDvrProperties()
```

Get the property collection of DVR settings that are specific to this application instance. These are defined in Application/DVR/Properties tag in Application.xml

**Returns:**

property collection of DVR settings

---

**getDvrApplicationContext**

```
public DvrApplicationContext getDvrApplicationContext()
```

Get live stream dvr application context

**Returns:**

live stream dvr application context

---

**getLiveStreamDvrRecorderControl**

```
public ILiveStreamDvrRecorderControl getLiveStreamDvrRecorderControl()
```

Get the Live Stream DVR Recorder Controller. This class will get called each time a stream is to be DVR-ed.

**Returns:**

Live Stream DVR Controller

---

**setLiveStreamDvrRecorderControl**

```
public void setLiveStreamDvrRecorderControl(ILiveStreamDvrRecorderControl controller)
```

Set the Live Stream DVR Controller.

**Parameters:**

controller - Live Stream DVR Controller

---

**getDvrRecorderList**

```
public String getDvrRecorderList()
```

Get the comma separated list of Dvr Recorder names being used by this application (see conf/Dvr.xml)

**Returns:**

comma separated list of Dvr Recorder names

---

---

## setDvrRecorderList

```
public void setDvrRecorderList(String recorderList)
```

Set the comma separated list of Dvr Recorder names being used by this application (see conf/Dvr.xml)

**Parameters:**

recorderList - comma separated list of Dvr Recorder names

---

## addDvrRecorderListener

```
public void addDvrRecorderListener(ILiveStreamDvrRecorderActionNotify listener)
```

Add a Dvr Recorder listener (see: ILiveStreamDvrRecorderActionNotify)

**Parameters:**

listener - Dvr Recorder listener

---

## removeDvrRecorderListener

```
public void removeDvrRecorderListener(ILiveStreamDvrRecorderActionNotify listener)
```

Remove a Dvr Recorder listener (see: ILiveStreamDvrRecorderActionNotify)

**Parameters:**

listener - Dvr Recorder listener

---

## notifyLiveStreamDvrRecorderCreate

```
public void notifyLiveStreamDvrRecorderCreate(ILiveStreamDvrRecorder dvr,  
String streamName)
```

Notify Dvr Recorder Create

**Parameters:**

dvr - DVR Recorder listener

streamName - stream Name

---

## notifyLiveStreamDvrRecorderInit

```
public void notifyLiveStreamDvrRecorderInit(ILiveStreamDvrRecorder dvr,  
String streamName)
```

Notify DVR Recorder has been initialized.

**Parameters:**

dvr - DVR Recorder listener \* @param streamName stream Name

---

## notifyLiveStreamDvrRecorderDestroy

```
public void notifyLiveStreamDvrRecorderDestroy(ILiveStreamDvrRecorder dvr)
```

Notify DVR Recorder has been destroyed.

**Parameters:**

dvr - DVR Recorder listener

---

(continued from last page)

## addDvrStreamManagerListener

```
public void addDvrStreamManagerListener(IDvrStreamManagerActionNotify listener)
```

Add a Dvr Application Store Manager listener (see: IDvrStoreActionNotify)

**Parameters:**

listener - Dvr Application Store Manager listener

---

## removeDvrStreamManagerListener

```
public void removeDvrStreamManagerListener(IDvrStreamManagerActionNotify listener)
```

Remove a Dvr Application Store Manager listener (see: IDvrStoreActionNotify)

**Parameters:**

listener - Dvr Application Store Manager listener

---

## notifyDvrStreamManagerCreate

```
public void notifyDvrStreamManagerCreate(IDvrStreamManager dvrStoreManager,  
String streamName)
```

Notify listeners that Dvr Application Store Manager has been created.

**Parameters:**

dvrStoreManager - Dvr Application Store Manager  
streamName

---

## notifyDvrStreamManagerInit

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

Notify listeners that Dvr Application Store Manager has been initialized.

**Parameters:**

dvrStoreManager - Dvr Application Store Manager

---

## notifyDvrStreamManagerDestroy

```
public void notifyDvrStreamManagerDestroy(IDvrStreamManager dvrManager)
```

Notify listeners that Dvr Application Store Manager has been destroyed.

**Parameters:**

dvrManager - Dvr Application Store Manager

---

## getMediaReaderContentType

```
public int getMediaReaderContentType(String mediaType)
```

Get the content type of a media stream name prefix (see IMediaReader.CONTENTTYPE\_\*)

**Parameters:**

mediaType - mediaType (such as flv or smil)

**Returns:**

content type (see IMediaReader.CONTENTTYPE\_\*)

## getMediaListProvider

```
public IMediaListProvider getMediaListProvider()
```

Get the current media list provider. The media list provider is used to resolve amlst:streamname requests to a media list (equivalent to a SMIL file).

**Returns:**

media list provider

## setMediaListProvider

```
public void setMediaListProvider(IMediaListProvider mediaListProvider)
```

Set the current media list provider. The media list provider is used to resolve amlst:streamname requests to a media list (equivalent to a SMIL file).

**Parameters:**

mediaListProvider - media list provider

## getMediacasterRTPRTSPRTPTTransportMode

```
public int getMediacasterRTPRTSPRTPTTransportMode()
```

RTP MediaCaster RTSP/RTP transport mode. See RTPMediaCaster.RTSPTTRANSPORTMODE\_\*

**Returns:**

RTP MediaCaster RTSP/RTP transport mode

## setMediacasterRTPRTSPRTPTTransportMode

```
public void setMediacasterRTPRTSPRTPTTransportMode(int  
mediacasterRTPRTSPRTPTTransportMode)
```

RTP MediaCaster RTSP/RTP transport mode. See RTPMediaCaster.RTSPTTRANSPORTMODE\_\*

**Parameters:**

mediacasterRTPRTSPRTPTTransportMode - RTP MediaCaster RTSP/RTP transport mode

## getProtocolUsage

```
public boolean[ ] getProtocolUsage()
```

Get the protocols in use by this application instance (see IApplicationInstance.PROTCOLUSAGE\_\*)

**Returns:**

protocols in use by this application instance (see IApplicationInstance.PROTCOLUSAGE\_\*)

## getProtocolUsage

```
public void getProtocolUsage(boolean[ ] protocolsInUse)
```

Get the protocols in use by this application instance (see IApplicationInstance.PROTCOLUSAGE\_\*)

## isDebugAppTimeout

```
public boolean isDebugAppTimeout()
```

---

(continued from last page)

If true appTimeout processing will be logged.

**Returns:**

true appTimeout processing will be logged

---

## setDebugAppTimeout

```
public void setDebugAppTimeout(boolean debugAppTimeout)
```

If true appTimeout processing will be logged.

**Parameters:**

debugAppTimeout - true appTimeout processing will be logged

---

## addRTPLListener

```
public void addRTPLListener(IRTPSessionNotify rtpListener)
```

Add listner to be notified of RTPSession create and destroy.

**Parameters:**

rtpListener - RTP listener

---

## removeRTPLListener

```
public void removeRTPLListener(IRTPSessionNotify rtpListener)
```

Remove listner to be notified of RTPSession create and destroy.

**Parameters:**

rtpListener - RTP listener

---

## notifyRTPSessionCreate

```
public void notifyRTPSessionCreate(RTPSession rtpSession)
```

Notify RTP session create

**Parameters:**

rtpSession - rtp session

---

## notifyRTPSessionDestroy

```
public void notifyRTPSessionDestroy(RTPSession rtpSession)
```

Notify RTP session destory

**Parameters:**

rtpSession - rtp session

---

## addHTTPListener

```
public void addHTTPListener(IHTTPSessionNotify httpListener)
```

Add listner to be notified of HTTPSession create and destroy.

**Parameters:**

httpListener - http listener

---

## removeHTTPListener

```
public void removeHTTPListener(IHTTPSessionNotify httpListener)
```

Remove listner to be notified of HTTPSession create and destroy.

**Parameters:**

httpListener - http listener

---

## notifyHTTPSessionCreate

```
public void notifyHTTPSessionCreate(IHTTPStreamerSession httpSession)
```

Notify HTTP session create

**Parameters:**

httpSession - HTTP session

---

## notifyHTTPSessionDestroy

```
public void notifyHTTPSessionDestroy(IHTTPStreamerSession httpSession)
```

Notify HTTP session destroy

**Parameters:**

httpSession - HTTP session

---

## getMediaCacheFilters

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

Get the list of allowed MediaCache sources.

**Returns:**

list of allowed MediaCache sources

---

## isMediaCacheSourceAllowed

```
public boolean isMediaCacheSourceAllowed(String sourceName)
```

Test to see if a MediaCache source name is allowed

**Parameters:**

sourceName - MediaCache source name

**Returns:**

true if allowed

---

## getPushPublishSessions

```
public java.util.List getPushPublishSessions(IMediaStream stream)
```

Get the current push publish sessions for a given IMediaStream

**Parameters:**

stream - stream

(continued from last page)

**Returns:**

push publish sessions

---

**getPushPublishSessions**

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

Get all active push publish sessions

**Returns:**

all active push publish sessions

---

**addPushPublishSession**

```
public void addPushPublishSession(IMediaStream stream,  
IPushPublishSession pushPublishSession)
```

Add a push publish session to the list of active push publish sessions

**Parameters:**

stream - stream

pushPublishSession - push publish session

---

**removePushPublishSession**

```
public void removePushPublishSession(IMediaStream stream,  
IPushPublishSession pushPublishSession)
```

Remove a push publish session

**Parameters:**

stream - stream

pushPublishSession - push publish session

---

**removePushPublishSessions**

```
public void removePushPublishSessions(IMediaStream stream)
```

Remove all push publish sessions for a give stream

**Parameters:**

stream - stream

---

**addPushPublishSessionListener**

```
public void addPushPublishSessionListener(IPushPublishSessionNotify  
pushPublishSessionListener)
```

Add a push publish session listener (see IPushPublishSessionNotify)

**Parameters:**

pushPublishSessionListener - push publish session

---

**removePushPublishSessionListener**

```
public void removePushPublishSessionListener(IPushPublishSessionNotify  
pushPublishSessionListener)
```

Remove a push publish session listener (see IPushPublishSessionNotify)

(continued from last page)

**Parameters:**

pushPublishSessionListener - push publish session

**notifyPushPublishSessionCreate**

```
public void notifyPushPublishSessionCreate(String streamName,  
    IMediaStream stream,  
    IPushPublishSession pushPublishSession)
```

Notify push publish session create

**Parameters:**

streamName - stream name  
stream - stream  
pushPublishSession - push publish session

**notifyPushPublishSessionDestroy**

```
public void notifyPushPublishSessionDestroy(String streamName,  
    IMediaStream stream,  
    IPushPublishSession pushPublishSession)
```

Notify push publish session destroy

**Parameters:**

streamName - stream name  
stream - stream  
pushPublishSession - push publish session

**getPushPublishSessionCount**

```
public int getPushPublishSessionCount()
```

Get the number of push publishing sessions.

**Returns:**

number of push publishing sessions

**getStreamRecorderProperties**

```
public WMSProperties getStreamRecorderProperties()
```

Get the property collection of StreamRecorder settings that are specific to this application instance

**Returns:**

property collection of StreamRecorder settings

## com.wowza.wms.application Interface IApplicationInstanceNotify

public interface IApplicationInstanceNotify  
extends

IApplicationInstanceNotify: listener interface used by IApplication addApplicationInstanceListener

### Method Summary

void	<a href="#">onApplicationInstanceCreate(IApplicationInstance applicationInstance)</a> Triggered when applicationInstance created
void	<a href="#">onApplicationInstanceDestroy(IApplicationInstance applicationInstance)</a> 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="#"><u>onApplicationCreate(IApplication application)</u></a> Triggered when application created
void	<a href="#"><u>onApplicationDestroy(IApplication application)</u></a> 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(WMSProperties from, WMSProperties to)</a> 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(String name)</a> Get property value as generic object.
boolean	<a href="#">getPropertyBoolean(String name, boolean defaultVal)</a> Get property as boolean, return default value if does not exist.
double	<a href="#">getPropertyDouble(String name, double defaultVal)</a> Get property as double, return default value if does not exist.
int	<a href="#">getPropertyInt(String name, int defaultVal)</a> Get property as int, return default value if does not exist.
long	<a href="#">getPropertyLong(String name, long defaultVal)</a> Get property as long, return default value if does not exist.
String	<a href="#">getPropertyStr(String name)</a> Get property as String
String	<a href="#">getPropertyStr(String name, String defaultVal)</a> Get property as String, return default value if does not exist.
void	<a href="#">putAll(java.util.Map m)</a>

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

**Methods inherited from interface java.util.Map**

clear, containsKey, containsValue, entrySet, equals, get, hashCode, isEmpty, keySet, put, putAll, remove, size, values

**Methods inherited from interface java.util.Map**

clear, containsKey, containsValue, entrySet, equals, get, hashCode, isEmpty, keySet, put, putAll, remove, size, values

## Constructors

### WMSProperties

```
public WMSProperties()
```

## Methods

### cloneProperties

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

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

**Parameters:**

from - source properties  
to - destination properties

### setProperty

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

(continued from last page)

Set property to generic object.

**Parameters:**

name - property name  
value - value

---

**getProperty**

```
public Object getProperty(String name)
```

Get property value as generic object.

**Parameters:**

name - property name

**Returns:**

value, null if does not exist

---

**getPropertyStr**

```
public String getPropertyStr(String name)
```

Get property as String

**Parameters:**

name - property name

**Returns:**

value as String, null if does not exist

---

**getPropertyStr**

```
public String getPropertyStr(String name,  
                           String defaultValue)
```

Get property as String, return default value if does not exist.

**Parameters:**

name - property name  
defaultValue - default value

**Returns:**

value as String, defaultValue if does not exist

---

**getPropertyBoolean**

```
public boolean getPropertyBoolean(String name,  
                                boolean defaultValue)
```

Get property as boolean, return default value if does not exist.

**Parameters:**

name - property name  
defaultValue - default value

**Returns:**

value as boolean, defaultValue if does not exist

---

## getPropertyInt

```
public int getPropertyInt(String name,  
                         int defaultVal)
```

Get property as int, return default value if does not exist.

**Parameters:**

name - property name  
defaultVal - default value

**Returns:**

value as boolean, defaultVal if does not exist

---

## getPropertyLong

```
public long getPropertyLong(String name,  
                           long defaultVal)
```

Get property as long, return default value if does not exist.

**Parameters:**

name - property name  
defaultVal - default value

**Returns:**

value as long, defaultVal if does not exist

---

## getPropertyDouble

```
public double getPropertyDouble(String name,  
                               double defaultVal)
```

Get property as double, return default value if does not exist.

**Parameters:**

name - property name  
defaultVal - default value

**Returns:**

value as double, defaultVal if does not exist

---

## getAllAsStrings

```
public String[] getAllAsStrings()
```

Return all properties as String[]. Format is "key=value".

**Returns:**

all properties as String[]

---

## toString

```
public String toString()
```

---

(continued from last page)

## **putAll**

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

---

Package

**com.wowza.wms.authentication**

## com.wowza.wms.authentication Class AuthenticateUsernamePasswordProviderBase

```
java.lang.Object
+-com.wowza.wms.authentication.AuthenticateUsernamePasswordProviderBase
```

### All Implemented Interfaces:

[IAuthenticateUsernamePasswordProvider](#)

```
public abstract class AuthenticateUsernamePasswordProviderBase
extends Object
implements IAuthenticateUsernamePasswordProvider
```

Base class for implementing HTTP and RTSP based custom authentication class.

### Field Summary

protected	<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`

**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(IApplicationInstance appInstance, AuthenticationItem authenticationItem)</a>
	Initialize authentication class when instantiated as part of an application instance
void	<a href="#">init(IVHost vhost, AuthenticationItem authenticationItem)</a>
	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(IVHost vhost, IHttpRequest req, IHttpResponse resp)</a>
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	<a href="#"><b>authenticateRTSP</b>(<b>RTPSession</b> rtspSession, com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp)</a>
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	<a href="#"><b>authenticateSIP</b>(RTPSession rtspSession, com.wowza.wms.sip.SIPRequestMessage req, com.wowza.wms.sip.SIPResponseMessages resp)</a>
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="#"><code>getClient()</code></a> Get client
String	<a href="#"><code>getPassword(String username)</code></a> Get password for a given user
<a href="#">RTPSession</a>	<a href="#"><code>getRTPSession()</code></a> Get RTP session
<a href="#">IVHost</a>	<a href="#"><code>getVHost()</code></a> Get vhost
void	<a href="#"><code>setClient(IClient client)</code></a> Set client
void	<a href="#"><code>setRTPSession(RTPSession rtpSession)</code></a> Set RTP session
void	<a href="#"><code>setVHost(IVHost vhost)</code></a> Set vhost
boolean	<a href="#"><code>userExists(String username)</code></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 accepeted conenction to this object.
String	<a href="#">getLastConnectAcceptedTimeString()</a>
	Get time (milliseconds) of the last accepeted conenction to this object as formatted string.
long	<a href="#">getLastConnectRejectedByReasonStamp(int reason)</a>
	Get time (milliseconds) of the last rejected conenction by reason to this object.

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

boolean	<a href="#"><b>validateNewConnection</b></a> (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 occured  
stamp - time stamp connection occured (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 connection to this object.

**Returns:**

time (milliseconds) of the last connection to this object

---

**getLastConnectAcceptedStampString**

```
public String getLastConnectAcceptedStampString()
```

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

**Returns:**

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

---

**getLastConnectAcceptedTime**

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

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

**Returns:**

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

---

**getLastConnectAcceptedTimeString**

```
public String getLastConnectAcceptedTimeString()
```

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

**Returns:**

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

---

**getLastConnectRejectedStamp**

```
public long getLastConnectRejectedStamp()
```

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

**Returns:**

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

---

**getLastConnectRejectedStampString**

```
public String getLastConnectRejectedStampString()
```

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

**Returns:**

time (milliseconds) of the last rejected connection 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 formated string.

**Returns:**

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

---

**getLastConnectRejectedByReasonStampString**

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

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

**Parameters:**

reason - reason, see REJECTREASON\_\*

**Returns:**

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

---

**getLastConnectRejectedByReasonStamp**

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

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

**Parameters:**

reason - reason, see REJECTREASON\_\*

**Returns:**

time (milliseconds) of the last rejected conenction 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 formated string.

**Parameters:**

reason - reason, see REJECTREASON\_\*

(continued from last page)

**Returns:**

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

**getLastDisconnectStampString****public String getLastDisconnectStampString( )**

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

**Returns:**

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

**getLastDisconnectStamp****public long getLastDisconnectStamp( )**

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

**Returns:**

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

void	<a href="#"><code>addAcceptConnectionAttribute</code></a> (String key, String item) Add and attribute to the resultObj that gets passed back to the client on successful connection
void	<a href="#"><code>call</code></a> (String handlerName) Simplified call client method/handler call.
void	<a href="#"><code>call</code></a> (String handlerName, <a href="#"><code>IModuleCallResult</code></a> resultObj, Object[] params) Call client method/handler.
void	<a href="#"><code>clearFastPlaySettings</code></a> () Force clear the fastPlay settings
void	<a href="#"><code>fcSubscribe</code></a> (String streamName) Subscribe to a live stream (for live stream repeater to start start from edge to origin)
void	<a href="#"><code>fcSubscribe</code></a> (String streamName, String mediaCasterType) Subscribe to a live stream (use a particular mediaCasterType)
void	<a href="#"><code>fcUnSubscribe</code></a> (String streamName) UnSubscribe from a stream
void	<a href="#"><code>fcUnSubscribeAll</code></a> () UnSubscribe to all streams that this client is current subscribed to
<a href="#"><code>IApplicationInstance</code></a>	<a href="#"><code>getAppInstance</code></a> () Get parent applicationInstance.
<a href="#"><code>IApplication</code></a>	<a href="#"><code>getApplication</code></a> () Get parent application.
int	<a href="#"><code>getBufferTime</code></a> () Get default buffer time for newly created mediaStream objects
int	<a href="#"><code>getClientId</code></a> () Get client id.
long	<a href="#"><code>getConnectTime</code></a> () Get time in milliseconds the client connected to the server.
String	<a href="#"><code>getDateStarted</code></a> () Get date and time of client connection
<a href="#"><code>ElapsedTimer</code></a>	<a href="#"><code>getElapsedTime</code></a> () Get elapsed time client has been connected.
<a href="#"><code>FastPlaySettings</code></a>	<a href="#"><code>getFastPlaySettings</code></a> () Get the current fastPlay settings.
String	<a href="#"><code>getFlashVer</code></a> () Get client flash version (same as FMS getAgent())
int	<a href="#"><code>getIdleFrequency</code></a> () Get client idle frequency (milliseconds)
String	<a href="#"><code>getIp</code></a> () Client ip address
long	<a href="#"><code>getLastValidateTime</code></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)
int	<a href="#"><u>getPlayStreamCount()</u></a> Get the number of play streams
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)
int	<a href="#"><u>getPublishStreamCount()</u></a> Get the number of publish streams
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="#">AMFObj</a>	<a href="#"><code>getRespAMFAudioObj(IMediaStream stream)</code></a> Get the audio response AMFObj for a given mediaStream
<a href="#">AMFObj</a>	<a href="#"><code>getRespAMFDataObj(IMediaStream stream)</code></a> Get the data response AMFObj for a given mediaStream
<a href="#">AMFObj</a>	<a href="#"><code>getRespAMFVideoObj(IMediaStream stream)</code></a> Get the video response AMFObj for a given mediaStream
<a href="#">ResponseFunctions</a>	<a href="#"><code>getRespFunctions()</code></a> Get client responseFunctions object.
<a href="#">AMFObj</a>	<a href="#"><code>getResponseAMFObj(int index)</code></a> Get the response channel AMFObj for channel index.
<a href="#">RTPStream</a>	<a href="#"><code>getRTPStream()</code></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="#"><code>getServerHostPort()</code></a> Get the hostPort object for the connection that is servicing this client
<a href="#">String</a>	<a href="#"><code>getSharedObjectReadAccess()</code></a> Get the shared object read access value.
<a href="#">String</a>	<a href="#"><code>getSharedObjectWriteAccess()</code></a> Get the shared object write access value.
<a href="#">String</a>	<a href="#"><code>getStreamAudioSampleAccess()</code></a> Get the audio sample access value.
<a href="#">java.io.File</a>	<a href="#"><code>getStreamFile(String streamName)</code></a> Get File object for stream with given name.
<a href="#">java.io.File</a>	<a href="#"><code>getStreamFile(String streamName, String streamExt)</code></a> Get File object for stream with given name and extension.
<a href="#">java.io.File</a>	<a href="#"><code>getStreamFile(String streamName, String streamExt, boolean doCreateFolder)</code></a> Get File object for stream with given name and extension.
<a href="#">String</a>	<a href="#"><code>getStreamReadAccess()</code></a> Get the stream read access value.
<a href="#">String</a>	<a href="#"><code>getStreamType()</code></a> Get default streamType
<a href="#">String</a>	<a href="#"><code>getStreamVideoSampleAccess()</code></a> Get the video sample access value.
<a href="#">String</a>	<a href="#"><code>getStreamWriteAccess()</code></a> Get the stream write access value.
<a href="#">String</a>	<a href="#"><code>getTimeRunning()</code></a> Get elapsed time of connection
<a href="#">double</a>	<a href="#"><code>getTimeRunningSeconds()</code></a> Get time running in seconds

<a href="#"><u>IOPerformanceCounter</u></a>	<a href="#"><u>getTotalIOPerformanceCounter()</u></a> Get performance counter for all bytes transferred to this client.
String	<a href="#"><u>getUri()</u></a> Get the full URI of the connection string
<a href="#"><u>IVHost</u></a>	<a href="#"><u>getVHost()</u></a> Get parent vHost
com.wowza.wms.protocol.wowz.WOWZSession	<a href="#"><u>getWowzSession()</u></a>
ClientWriteListener	<a href="#"><u>getWriteListener()</u></a> Object that tracks write operations
boolean	<a href="#"><u>isAcceptConnection()</u></a> Is auto accept connection
boolean	<a href="#"><u>isConnected()</u></a> Is this client connected
boolean	<a href="#"><u>isEncrypted()</u></a> Is this connection encrypted (RTMPE or RTMPTE)
boolean	<a href="#"><u>isFlashMediaLiveEncoder()</u></a> Returns true if this connection is the Flash Media Live Encoder
boolean	<a href="#"><u>isFlashVersion10()</u></a> Returns true if the Flash version is equal or greater than 10.x.x.x
boolean	<a href="#"><u>isFlashVersion90115()</u></a> Returns true if the Flash version is equal or greater than 9.0.115.x
boolean	<a href="#"><u>isFlashVersionH264Capable()</u></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="#"><u>isLiveRepeater()</u></a> Returns true if this connection is from the live stream repeater
boolean	<a href="#"><u>isObjectEncodingAMF0()</u></a> Is the object encoding for this client AMF0
boolean	<a href="#"><u>isObjectEncodingAMF3()</u></a> Is the object encoding for this client AMF3
boolean	<a href="#"><u>isSecure()</u></a> Is this connection protected by either SSL or encryption (RTMPE, RTMPTE, RTMPS)
boolean	<a href="#"><u>isSendCloseOnRejectConnection()</u></a>
boolean	<a href="#"><u>isSSL()</u></a> Is this connection SSL (RTMPS)
boolean	<a href="#"><u>isValidatingFMLEConnections()</u></a> Returns true if validating FMLE connection (default is false)
int	<a href="#"><u>ping(IModulePingResult pingResult)</u></a> Ping client.

void	<a href="#"><u>redirectConnection(String url)</u></a> Redirection connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected and info.ex.code of 302)
void	<a href="#"><u>redirectConnection(String url, String description)</u></a> Redirection connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected and info.ex.code of 302)
void	<a href="#"><u>redirectConnection(String url, String description, AMFData errorObj)</u></a> Redirection connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected and info.ex.code of 302).
void	<a href="#"><u>redirectConnection(String url, String description, String errorStr)</u></a> 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(AMFData errorObj)</u></a> Reject connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected)
void	<a href="#"><u>rejectConnection(String errorStr)</u></a> Reject connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected)
void	<a href="#"><u>rejectConnection(String description, AMFData errorObj)</u></a> Reject connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected)
void	<a href="#"><u>rejectConnection(String description, String errorStr)</u></a> Reject connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected)
void	<a href="#"><u>reparentClient(IVHost vhost)</u></a> Move a client object to a new vhost.
void	<a href="#"><u>setAcceptConnection(boolean acceptConnection)</u></a> Set the default for accept connection
void	<a href="#"><u>setAcceptConnectionDescription(String description)</u></a> Call this method from onConnect to set the info.description property returned in NetConnection onStatus handler
void	<a href="#"><u>setAcceptConnectionExObj(AMFDataObj acceptConnectionExObj)</u></a> Call this method from onConnect to set the info.ex property returned in NetConnection onStatus handler
void	<a href="#"><u>setAcceptConnectionObj(AMFData acceptConnectionObj)</u></a> 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(int bufferTime)</u></a> Set default buffer time for newly created mediaStream objects
void	<a href="#"><u>setFastPlaySettings(FastPlaySettings fastPlaySettings)</u></a> 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>setSendCloseOnRejectConnection</u></a> (boolean sendCloseOnRejectConnection)
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>setWowzSession</u></a> (com.wowza.wms.protocol.wowz.WOWZSession wowzSession)

void	<a href="#"><u>shutdownClient()</u></a> Gracefully shutdown a client.
int	<a href="#"><u>testFlashVersion(int[] version)</u></a> 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

### **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**

(continued from last page)

Constant value: \*

## WRITE\_ACCESS\_NONE

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

Constant value:

## Methods

### getClientId

```
public int getClientId()
```

Get client id. Assigned by server and connection time.

**Returns:**

client id

### getFlashVer

```
public String getFlashVer()
```

Get client flash version (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

(continued from last page)

## getAppInstance

```
public IApplicationInstance getAppInstance( )
```

Get parent applicationInstance. Is null if connection reject or before accepted.

**Returns:**

parent applicationInstance

---

## getApplication

```
public IApplication getApplication( )
```

Get parent application. Is null if connection reject or before accepted.

**Returns:**

application

---

## getPlayStreams

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

Get a collection of all play streams. Play streams are streams that are created due to a call to play.

### Iterate Play Streams

```
IClient client;

List playStreams = client.getPlayStreams();
Iterator iter = playStreams.iterator();
while(iter.hasNext())
{
    IMediaStream stream = (IMediaStream)iter.next();
    WMSLoggerFactory.getLogger(null).debug("stream: "+stream.getName());
}
```

**Returns:**

collection of play streams

---

## getPublishStreams

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

(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

## getPlayStreamCount

```
public int getPlayStreamCount()
```

Get the number of play streams

### Returns:

number of play streams

## getPublishStreamCount

```
public int getPublishStreamCount()
```

Get the number of publish streams

### Returns:

number of publish streams

## shutdownClient

```
public void shutdownClient()
```

Gracefully shutdown a client. Only use this method to shutdown a client if you know the client is connected to Wowza Pro. If the client may be disconnected from Wowza Pro, use IClient.setShutdownClient(true)

## touch

```
public void touch()
```

Update the last touch time for client

(continued from last page)

## isConnected

```
public boolean isConnected()
```

Is this client connected

**Returns:**

is connected

---

## getProperties

```
public WMSProperties getProperties()
```

Get client properties

**Returns:**

collection of client properties

---

## getStreamType

```
public String getStreamType()
```

Get default streamType

**Returns:**

streamType

---

## setStreamType

```
public void setStreamType(String streamType)
```

Set default streamType

**Parameters:**

streamType - streamType

---

## isAcceptConnection

```
public boolean isAcceptConnection()
```

Is auto accept connection

**Returns:**

auto accept connection

---

## acceptConnection

```
public void acceptConnection()
```

Accept connection

---

## acceptConnection

```
public void acceptConnection(String successStr)
```

Accept connection

**Parameters:**

(continued from last page)

successStr - application property value added to NetConnection.Connect.Success event object

---

## acceptConnection

```
public void acceptConnection(AMFData successObj)
```

Accept connection

**Parameters:**

successObj - application property value added to NetConnection.Connect.Success event object

---

## rejectConnection

```
public void rejectConnection()
```

Reject connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected)

---

## rejectConnection

```
public void rejectConnection(String errorStr)
```

Reject connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected)

**Parameters:**

errorStr - error (returned in onStatus handler in info.application parameter)

---

## rejectConnection

```
public void rejectConnection(AMFData errorObj)
```

Reject connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected)

**Parameters:**

errorObj - error AMFData (returned in onStatus handler in info.application parameter)

---

## rejectConnection

```
public void rejectConnection(String description,  
                           String errorStr)
```

Reject connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected)

**Parameters:**

description - description (returned in onStatus handler in info.description parameter)

errorStr - error (returned in onStatus handler in info.application parameter)

---

## rejectConnection

```
public void rejectConnection(String description,  
                           AMFData errorObj)
```

Reject connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected)

**Parameters:**

description - description (returned in onStatus handler in info.description parameter)

errorObj - error AMFData (returned in onStatus handler in info.application parameter)

---

---

(continued from last page)

## redirectConnection

```
public void redirectConnection(String url)
```

Redirection connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected and info.ex.code of 302)

**Parameters:**

url - url (returned in onStatus handler in info.ex.redirect parameter)

---

## redirectConnection

```
public void redirectConnection(String url,  
    String description)
```

Redirection connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected and info.ex.code of 302)

**Parameters:**

url - url (returned in onStatus handler in info.ex.redirect parameter)

description - description (returned in onStatus handler in info.description parameter)

---

## redirectConnection

```
public void redirectConnection(String url,  
    String description,  
    String errorStr)
```

Redirection connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected and info.ex.code of 302)

**Parameters:**

url - url (returned in onStatus handler in info.ex.redirect parameter)

description - description (returned in onStatus handler in info.description parameter)

errorStr - error (returned in onStatus handler in info.application parameter)

---

## redirectConnection

```
public void redirectConnection(String url,  
    String description,  
    AMFData errorObj)
```

(continued from last page)

Redirection connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected and info.ex.code of 302). Code must be added client side to capture the NetConnection.Connect.Rejected message, look for the info.ex.code value of 302 and attempt a reconnect to the info.ex.redirect url. Client side example:

```
public function ncOnStatus(infoObject:NetStatusEvent)
{
    if (infoObject.info.code == "NetConnection.Connect.Rejected")
    {
        if (infoObject.info.ex.code == 302)
        {
            trace("Connection redirected: "+infoObject.info.ex.redirect);
            nc.connect(infoObject.info.ex.redirect);
        }
        else
            trace("Connection rejected");
    }
}
nc.addEventListener(NetStatusEvent.NET_STATUS, ncOnStatus);
```

**Parameters:**

url - url (returned in onStatus handler in info.ex.redirect parameter)  
 description - description (returned in onStatus handler in info.description parameter)  
 errorObj - error AMFData (returned in onStatus handler in info.application parameter)

**getVHost**

```
public IVHost getVHost()
```

Get parent vHost

**Returns:**

vHost object

**call**

```
public void call(String handlerName,
                 IModuleCallResult resultObj,
                 Object[] params)
```

(continued from last page)

Call client method/handler. You can call any method/handler attached to NetConnection object. Such as netConnection.myFunc

## Call Client Handler Method With Result Object

```
IClient client;

class ReturnObj implements IModuleCallResult
{
    public void onResult(IClient client, RequestFunction function,
                        AMFDataList params)
    {
        WMSLoggerFactory.getLogger(null).debug("onResult");
    }
}

client.call("onCustomMethod", new ReturnObj(), "param1", 1.2345, false, new Date());
```

### Parameters:

handlerName - handler name  
resultObj - if client method returns data this object will receive callback, null if don't care or no return on client method/handler.  
params - variable list of argument. Will be automatically wrapped in AMFData objects

---

## call

```
public void call(String handlerName)
```

Simplified call client method/handler call. No resultObj. No parameters

### Parameters:

handlerName - handler name

---

## ping

```
public int ping(IModulePingResult pingResult)
```

(continued from last page)

Ping client. Unlike FMS this ping is asynchronous. pingResult object will be notified of successful response by client or timeout.

## Ping Client and Kill Connection If Client Has Died

```
class PingResult implements IModulePingResult
{
    public void onResult(IClient client, long pingTime, int pingId, boolean result)
    {
        WMSLogger log = WMSLoggerFactory.getLogger(null);
        log.debug("onResult: result:" + result);
        if (!result)
        {
            // client has died lets kill it
            client.getAppInstance().shutdownClient(client);
        }
        else
            log.debug("lastPingTime: " + client.getPingRoundTripTime());
    }
}
client.ping(new PingResult());
```

**Parameters:**

pingResult - ping result object. Will receive callback on result of ping or timeout. Can be null.

**Returns:**

ping id

### getIp

```
public String getIp()
```

Client ip address

**Returns:**

client ip address

### getFastPlaySettings

```
public FastPlaySettings getFastPlaySettings()
```

Get the current fastPlay settings. FastPlay settings when set from the client are immediately cleared on the next call to seek/unpause/play.

**Returns:**

fastPlay settings

(continued from last page)

## setFastPlaySettings

```
public void setFastPlaySettings(FastPlaySettings fastPlaySettings)
```

Set fastPlay settings

**Parameters:**

fastPlaySettings - fastPlay settings

---

## clearFastPlaySettings

```
public void clearFastPlaySettings()
```

Force clear the fastPlay settings

---

## getRespFunctions

```
public ResponseFunctions getRespFunctions()
```

Get client responseFunctions object. The responseFunctions object is the method by which events are sent to the client. This code snippet shows how to send an onStatus event for a mediaStream (stream) to the client.

### Send Status Message to Client

```
ResponseFunctions respFunctions = client.getRespFunctions();

ResponseFunction resp = new ResponseFunction();
resp.createDefaultMessage("onStatus", 0.0);
resp.setRetAMFObj(stream.getRespAMFDataObj());
resp.setSrc(stream.getSrc());

AMFDataObj data = new AMFDataObj();
data.put("level", new AMFDataItem("status"));
data.put("code", new AMFDataItem("NetStream.Publish.Success"));
data.put("description", new AMFDataItem(stream.getName()+" is now unpublished."));
data.put("clientid", new AMFDataItem(clientID));
resp.addBody(data);

respFunctions.add(resp);
```

**Returns:**

responseFunctions object

---

## getConnectTime

```
public long getConnectTime()
```

(continued from last page)

Get time in milliseconds the client connected to the server. To get the duration of time in milliseconds that the client has been connected to the server us the following formula: (System.currentTimeMillis() - client.getConnectTime())

**Returns:**

duration connected (milliseconds)

---

**getPingRoundTripTime**

```
public long getPingRoundTripTime()
```

Get turn around time (milliseconds) of last ping request

**Returns:**

time (milliseconds)

---

**getDateStarted**

```
public String getDateStarted()
```

Get date and time of client connection

**Returns:**

date and time of connection

---

**getTimeRunning**

```
public String getTimeRunning()
```

Get elapsed time of connection

**Returns:**

elapsed time of connection

---

**getTimeRunningSeconds**

```
public double getTimeRunningSeconds()
```

Get time running in seconds

**Returns:**

time running in seconds

---

**getStreamFile**

```
public java.io.File getStreamFile(String streamName)
```

Get File object for stream with given name. Assumes file extension is .flv.

**Parameters:**

streamName - stream name

**Returns:**

file

---

**getStreamFile**

```
public java.io.File getStreamFile(String streamName,
                                 String streamExt)
```

(continued from last page)

Get File object for stream with given name and extension. For example test.flv would be getStreamFile("test", "flv");

## Get File Descriptor

```
IClient client;
File file = client.getStreamFile("test", "flv");
WMSLogger log = WMSLoggerFactory.getLogger(null);
if (file != null)
{
    if (file.exists())
        log.debug("getStreamFile: " + file.getPath());
}
```

**Parameters:**

- streamName - stream name
- streamExt - stream extension

**Returns:**

file

### getStreamFile

```
public java.io.File getStreamFile(String streamName,
    String streamExt,
    boolean doCreateFolder)
```

Get File object for stream with given name and extension. If doCreateFolder is true, create folder necessary to make path exist. For example test.flv would be getStreamFile("test", "flv");

**Parameters:**

- streamName - stream name
- streamExt - stream extension
- doCreateFolder - create folders if needed

**Returns:**

file

### getBufferTime

```
public int getBufferTime()
```

Get default buffer time for newly created mediaStream objects

**Returns:**

default buffer time (milliseconds)

### setBufferTime

```
public void setBufferTime(int bufferTime)
```

---

(continued from last page)

Set default buffer time for newly created mediaStream objects

**Parameters:**

bufferTime - bufer time (milliseconds)

---

## getResponseAMFObj

```
public AMFObj getResponseAMFObj(int index)
```

Get the response channel AMFObj for channel index.

**Parameters:**

index - channel index

**Returns:**

AMFObj

---

## getRespAMFAudioObj

```
public AMFObj getRespAMFAudioObj(IMediaStream stream)
```

Get the audio response AMFObj for a given mediaStream

**Parameters:**

stream - mediaStream

**Returns:**

response channel AMFObj

---

## getRespAMFVideoObj

```
public AMFObj getRespAMFVideoObj(IMediaStream stream)
```

Get the video response AMFObj for a given mediaStream

**Parameters:**

stream - mediaStream

**Returns:**

response channel AMFObj

---

## getRespAMFDataObj

```
public AMFObj getRespAMFDataObj(IMediaStream stream)
```

Get the data response AMFObj for a given mediaStream

**Parameters:**

stream - mediaStream

**Returns:**

response channel AMFObj

---

## getQueryStr

```
public String getQueryStr()
```

(continued from last page)

Get the query string part of the connection string. If the connection string where:  
`rtmp://localhost/app/appInst?data1=myData1&data2=myData2`, then query string would be  
`data1=myData1&data2=myData2`

**Returns:**

query string

**getReferrer****public String getReferrer()**

Get the referrer data for this connection. The referrer in Wowza Streaming Engine terms is the full url to the Flash movie that connected to the server. Checking referrer 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("mycoolfash.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-colon 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-colon delimited list is compared against incoming stream play requests. If any item in the list completely matches the stream name or the start of the stream name, then access is allowed. For example, if the streamWriteAccess list is set to "testa/testb;testc" then the following play requests would be granted the following access:

- testc: Granted Access
- testc/test: Granted Access
- testC/test: Denied Access (incorrect case)
- testa/testb: Granted Access
- testa/testb123: Granted Access
- testa/testb/file123: Granted Access
- testa/test: Denied Access (incomplete match)

Values in this list are always case sensitive.

**Parameters:**

streamWriteAccess - shared object read access value

## getWriteListener

```
public ClientWriteListener getWriteListener()
```

Object that tracks write operations

**Returns:**

Object that tracks write operations

## addAcceptConnectionAttribute

```
public void addAcceptConnectionAttribute(String key,  
AMFDataObj item)
```

Add and attribute to the resultObj that gets passed back to the client on successful connection

**Parameters:**

key - key  
item - item to add

## addAcceptConnectionAttribute

```
public void addAcceptConnectionAttribute(String key,  
String item)
```

Add and attribute to the resultObj that gets passed back to the client on successful connection

(continued from last page)

**Parameters:**

key - key  
item - item to add

---

**getRepeaterOriginUrl**

```
public String getRepeaterOriginUrl()
```

Get the origin URL used by the Live Stream Repeater

**Returns:**

URL used by the Live Stream Repeater

---

**setRepeaterOriginUrl**

```
public void setRepeaterOriginUrl(String repeaterOriginUrl)
```

Set the origin URL used by the Live Stream Repeater

**Parameters:**

repeaterOriginUrl - URL used by the Live Stream Repeater

---

**getLastValidateTime**

```
public long getLastValidateTime()
```

Get last time (millisecond) the connection was validated with a ping

**Returns:**

last time (millisecond) the connection was validated with a ping

---

**setLastValidateTime**

```
public void setLastValidateTime(long lastValidateTime)
```

Set the last time (milliseconds) the connection was validated with ping

**Parameters:**

lastValidateTime - last time (milliseconds) the connection was validated with ping

---

**getPingTimeout**

```
public int getPingTimeout()
```

Get the ping timeout (milliseconds)

**Returns:**

ping timeout (milliseconds)

---

**isLiveRepeater**

```
public boolean isLiveRepeater()
```

Returns true if this connection is from the live stream repeater

**Returns:**

true if this connection is from the live stream repeater

---

## isFlashVersionH264Capable

```
public boolean isFlashVersionH264Capable()
```

Returns true if the connected client is capable of playing H.264 video (Flash player 9.0.45.x or greater)

**Returns:**

returns true if the connected client is capable of playing H.264 video (Flash player 9.0.45.x or greater)

---

## isFlashVersion90115

```
public boolean isFlashVersion90115()
```

Returns true if the Flash version is equal or greater than 9.0.115.x

**Returns:**

returns true if the Flash version is equal or greater than 9.0.115.x

---

## isFlashVersion10

```
public boolean isFlashVersion10()
```

Returns true if the Flash version is equal or greater than 10.x.x.x

**Returns:**

returns true if the Flash version is equal or greater than 10.x.x.x

---

## isFlashMediaLiveEncoder

```
public boolean isFlashMediaLiveEncoder()
```

Returns true if this connection is the Flash Media Live Encoder

**Returns:**

true if this connection is the Flash Media Live Encoder

---

## testFlashVersion

```
public int testFlashVersion(int[] version)
```

Test to see if the connected client flash version is equal to or greater than a given value. Example: int isGood = testFlashVersion( { 9, 0, 5, 12 } );

**Parameters:**

version - array of version values

**Returns:**

1 if greater, 0 if equal, -1 if less than

---

## isObjectEncodingAMF3

```
public boolean isObjectEncodingAMF3()
```

Is the object encoding for this client AMF3

**Returns:**

true if the object encoding for this client is AMF3

---

---

## isObjectEncodingAMF0

```
public boolean isObjectEncodingAMF0()
```

Is the object encoding for this client AMF0

**Returns:**

true if 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 propert of NetConnection onStatus handler

---

(continued from last page)

## getRTPStream

```
public RTPStream getRTPStream()
```

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

**Returns:**

underlying RTPStream object

---

## setAcceptConnection

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

Set the default for accept connection

**Parameters:**

acceptConnection - default for accept connection

---

## setShutdownClient

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

Gracefully and forcefully shutdown a client.

**Parameters:**

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

---

## reparentClient

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

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

**Parameters:**

vhost - new vhost

---

## getMaximumSetBufferTime

```
public int getMaximumSetBufferTime()
```

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

**Returns:**

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

---

## getMaximumPendingWriteBytes

```
public int getMaximumPendingWriteBytes()
```

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

**Returns:**

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

(continued from last page)

## getLiveStreamPacketizerList

```
public String getLiveStreamPacketizerList()
```

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

**Returns:**

comma separated list of LiveStreamPacketizers names

---

## setLiveStreamPacketizerList

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

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

**Parameters:**

liveStreamPacketizerList - comma separated list of LiveStreamPacketizers names

---

## getLiveStreamTranscoderList

```
public String getLiveStreamTranscoderList()
```

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

**Returns:**

comma separated list of LiveStreamTranscoders names

---

## setLiveStreamTranscoderList

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

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

**Parameters:**

liveStreamTranscoderList - comma separated list of LiveStreamTranscoders names

---

## getElapsedTime

```
public ElapsedTimer getElapsedTime()
```

Get elapsed time client has been connected.

**Returns:**

elapsed time client has been connected.

---

## getLiveRepeaterCapabilities

```
public int getLiveRepeaterCapabilities()
```

Get the live repeater capabilities of this connection

**Returns:**

live repeater capabilities of this connection

---

## setLiveRepeaterCapabilities

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

(continued from last page)

Set the live repeater capabilities of this connection

**Parameters:**

liveRepeaterCapabilities - live repeater capabilities of this connection

---

**fcSubscribe**

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

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

**Parameters:**

streamName - stream name

---

**fcSubscribe**

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

Subscribe to a live stream (use a particular mediaCasterType)

**Parameters:**

streamName - stream name

mediaCasterType - media caster type name

---

**fcUnSubscribeAll**

```
public void fcUnSubscribeAll()
```

UnSubscribe to all streams that this client is current subscribed to

---

**fcUnSubscribe**

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

UnSubscribe from a stream

**Parameters:**

streamName - stream name

---

**isValidFMLEConnections**

```
public boolean isValidFMLEConnections()
```

Returns true if validating FMLE connection (default is false)

**Returns:**

true if validating FMLE connection

---

**setValidateFMLEConnections**

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

Returns true if validating FMLE connection (default is false)

**Parameters:**

validateFMLEConnections - true if validating FMLE connection

(continued from last page)

**setThreadContext**

```
public void setThreadContext()
```

Set the thread logging context to this client

---

**getWowzSession**

```
public com.wowza.wms.protocol.wowz.WOWZSession getWowzSession()
```

---

**setWowzSession**

```
public void setWowzSession(com.wowza.wms.protocol.wowz.WOWZSession wowzSession)
```

---

**isSendCloseOnRejectConnection**

```
public boolean isSendCloseOnRejectConnection()
```

---

**setSendCloseOnRejectConnection**

```
public void setSendCloseOnRejectConnection(boolean sendCloseOnRejectConnection)
```

## com.wowza.wms.client Interface IClientNotify

public interface IClientNotify  
extends

IClientNotify: listener interface used by [IAplicationInstance.addClientListener\(IClientNotify\)](#)  
See Also:

[IAplicationInstance.addClientListener\(IClientNotify\)](#)

### Method Summary

void	<a href="#">onClientAccept(IClient client)</a> Triggered when client connection accepted
void	<a href="#">onClientConnect(IClient client)</a> Triggered when client attempt connection
void	<a href="#">onClientDisconnect(IClient client)</a> Triggered when client disconnected
void	<a href="#">onClientReject(IClient client)</a> 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, <a href="#">java.util.Date</a> 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, <a href="#">java.util.Date</a> date, long stamp) Triggered when client disconnected
void	<a href="#">onRejectConnection</a> ( <a href="#">ConnectionCounter</a> connectionCounter, <a href="#">ConnectionHolder</a> connectionHolder, int reason, <a href="#">java.util.Date</a> date, long stamp) Triggered when client connection rejected

### Methods

#### onAcceptConnection

```
public void onAcceptConnection(ConnectionCounter connectionCounter,  
    ConnectionHolder connectionHolder,  
    java.util.Date date,  
    long stamp)
```

Triggered when client connection accepted

**Parameters:**

- connectionCounter
- connectionHolder
- date
- stamp

#### onRejectConnection

```
public void onRejectConnection(ConnectionCounter connectionCounter,  
    ConnectionHolder connectionHolder,  
    int reason,  
    java.util.Date date,  
    long stamp)
```

Triggered when client connection rejected

**Parameters:**

- connectionCounter
- connectionHolder
- reason

(continued from last page)

date  
stamp

---

## onDisconnect

```
public void onDisconnect(ConnectionCounter connectionCounter,  
    ConnectionHolder connectionHolder,  
    boolean isValid,  
    java.util.Date date,  
    long stamp)
```

Triggered when client disconnected

**Parameters:**

connectionCounter  
connectionHolder  
isValid  
date  
stamp

---

Package

**com.wowza.wms.closedcaption.cea608**

## com.wowza.wms.closedcaption-cea608

### Class ClosedCaptionCEA608Utils

```
java.lang.Object
+-com.wowza.wms.timedtext-cea608.CEA608Constants
  +-com.wowza.wms.timedtext-cea608.ClosedCaptionCEA608Utils
    +-com.wowza.wms.closedcaption-cea608.ClosedCaptionCEA608Utils
```

---

public class **ClosedCaptionCEA608Utils**  
extends [ClosedCaptionCEA608Utils](#)

Fields inherited from class [com.wowza.wms.timedtext-cea608.ClosedCaptionCEA608Utils](#)

[CEA608\\_EXTENDED\\_FR\\_SP\\_MAP](#), [CEA608\\_EXTENDED\\_NA\\_MAP](#),  
[CEA608\\_EXTENDED\\_PORT\\_GERMAN\\_DANISH\\_MAP](#), [CEA608\\_MUSICAL\\_NOTE](#), [CEA608\\_SOLID\\_BLOCK](#), [CHANNEL1](#),  
[CHANNEL2](#), [COL\\_0](#), [COL\\_12](#), [COL\\_16](#), [COL\\_20](#), [COL\\_24](#), [COL\\_28](#), [COL\\_4](#), [COL\\_5](#), [COLO\\_BLUE](#), [COLO\\_CYAN](#),  
[COLO\\_GREEN](#), [COLO\\_HIBYTE](#), [COLO\\_LOWBYTE](#), [COLO\\_MAGENTA](#), [COLO\\_RED](#), [COLO\\_WHITE](#), [COLO\\_YELLOW](#),  
[COLN\\_LOWBYTE](#), [CONTROLCODES\\_AOF](#), [CONTROLCODES\\_AON](#), [CONTROLCODES\\_BAO](#), [CONTROLCODES\\_BAS](#),  
[CONTROLCODES\\_BBO](#), [CONTROLCODES\\_BBS](#), [CONTROLCODES\\_BCO](#), [CONTROLCODES\\_BCS](#), [CONTROLCODES\\_BGO](#),  
[CONTROLCODES\\_BGS](#), [CONTROLCODES\\_BMO](#), [CONTROLCODES\\_BMS](#), [CONTROLCODES\\_BRO](#), [CONTROLCODES\\_BRS](#),  
[CONTROLCODES\\_BS](#), [CONTROLCODES\\_BT](#), [CONTROLCODES\\_BWO](#), [CONTROLCODES\\_BWS](#), [CONTROLCODES\\_BYO](#),  
[CONTROLCODES\\_BYS](#), [CONTROLCODES\\_CR](#), [CONTROLCODES\\_DER](#), [CONTROLCODES\\_EDM](#), [CONTROLCODES\\_ENM](#),  
[CONTROLCODES\\_EOC](#), [CONTROLCODES\\_FA](#), [CONTROLCODES\\_FAU](#), [CONTROLCODES\\_FON](#), [CONTROLCODES\\_RCL](#),  
[CONTROLCODES\\_RDC](#), [CONTROLCODES\\_RTD](#), [CONTROLCODES\\_RU2](#), [CONTROLCODES\\_RU3](#), [CONTROLCODES\\_RU4](#),  
[CONTROLCODES\\_TO1](#), [CONTROLCODES\\_TO2](#), [CONTROLCODES\\_TO3](#), [CONTROLCODES\\_TR](#), [COUNT\\_COLS](#),  
[COUNT\\_ROWS](#), [DONOTHING\\_BYTE](#), [DONOTHING\\_SHORT](#), [FORMAT\\_COLOR](#), [FORMAT\\_FLASH](#), [FORMAT\\_ITALICS](#),  
[FORMAT\\_ITALICS\\_UNDERLINE](#), [LANG\\_ASCII](#), [LANG\\_ASCII\\_DBL](#), [LANG\\_GB231280](#), [LANG\\_KSC56011987](#),  
[LANG\\_PRIV1](#), [LANG\\_PRIV2](#), [LANG\\_REG1](#), [MAX\\_CCCOUNT](#), [ODDPARITY](#), [SEI\\_PAYLOADTYPE](#), [SEI\\_STARTCODE](#),  
[SEI\\_USERDATATYPECODE](#), [UTF8\\_BRITISH\\_POUND](#), [UTF8\\_BULLET](#), [UTF8\\_CENTS](#),  
[UTF8\\_CLOSING\\_DOUBLE\\_QUOTE](#), [UTF8\\_CLOSING\\_GUILLEMET](#), [UTF8\\_CONNECTING\\_VERTICAL\\_BAR](#),  
[UTF8\\_COPYRIGHT](#), [UTF8\\_CURRENCY](#), [UTF8\\_DASH](#), [UTF8\\_DEGREE](#), [UTF8\\_DOUBLEQUOTE](#), [UTF8\\_EM\\_DASH](#),  
[UTF8\\_EXCLAMATION](#), [UTF8\\_GREATERTHAN](#), [UTF8\\_INVERTED\\_EXCLAMATION](#), [UTF8\\_LEFT\\_SINGLE\\_QUOTATION](#),  
[UTF8\\_LEFTPAREN](#), [UTF8\\_LESSTHAN](#), [UTF8\\_LOWER\\_A](#), [UTF8\\_LOWER\\_A\\_WITH\\_CIRCUMFLEX](#),  
[UTF8\\_LOWER\\_A\\_WITH\\_DIAERESIS](#), [UTF8\\_LOWER\\_A\\_WITH\\_GRAVE](#), [UTF8\\_LOWER\\_A\\_WITH\\_RING](#),  
[UTF8\\_LOWER\\_A\\_WITH\\_TILDE](#), [UTF8\\_LOWER\\_B](#), [UTF8\\_LOWER\\_C](#), [UTF8\\_LOWER\\_D](#), [UTF8\\_LOWER\\_E](#),  
[UTF8\\_LOWER\\_E\\_WITH\\_CIRCUMFLEX](#), [UTF8\\_LOWER\\_E\\_WITH\\_DIAERESIS](#), [UTF8\\_LOWER\\_E\\_WITH\\_GRAVE](#),  
[UTF8\\_LOWER\\_F](#), [UTF8\\_LOWER\\_G](#), [UTF8\\_LOWER\\_H](#), [UTF8\\_LOWER\\_I](#), [UTF8\\_LOWER\\_I\\_WITH\\_CIRCUMFLEX](#),  
[UTF8\\_LOWER\\_I\\_WITH\\_DIAERESIS](#), [UTF8\\_LOWER\\_I\\_WITH\\_GRAVE](#), [UTF8\\_LOWER\\_J](#), [UTF8\\_LOWER\\_K](#),  
[UTF8\\_LOWER\\_L](#), [UTF8\\_LOWER\\_M](#), [UTF8\\_LOWER\\_N](#), [UTF8\\_LOWER\\_O](#), [UTF8\\_LOWER\\_O\\_WITH\\_CIRCUMFLEX](#),  
[UTF8\\_LOWER\\_O\\_WITH\\_DIAERESIS](#), [UTF8\\_LOWER\\_O\\_WITH\\_GRAVE](#), [UTF8\\_LOWER\\_O\\_WITH\\_STROKE](#),  
[UTF8\\_LOWER\\_O\\_WITH\\_TILDE](#), [UTF8\\_LOWER\\_P](#), [UTF8\\_LOWER\\_Q](#), [UTF8\\_LOWER\\_R](#), [UTF8\\_LOWER\\_S](#),  
[UTF8\\_LOWER\\_T](#), [UTF8\\_LOWER\\_U](#), [UTF8\\_LOWER\\_U\\_WITH\\_CIRCUMFLEX](#), [UTF8\\_LOWER\\_U\\_WITH\\_DIAERESIS](#),  
[UTF8\\_LOWER\\_U\\_WITH\\_GRAVE](#), [UTF8\\_LOWER\\_V](#), [UTF8\\_LOWER\\_W](#), [UTF8\\_LOWER\\_X](#), [UTF8\\_LOWER\\_Y](#),  
[UTF8\\_LOWER\\_Z](#), [UTF8\\_MINUSYPHEN](#), [UTF8\\_MUSICAL\\_NOTE](#), [UTF8\\_NBSP](#), [UTF8\\_NOTMATCH](#), [UTF8\\_ONE\\_HALF](#),  
[UTF8\\_OPENING\\_DOUBLE\\_QUOTE](#), [UTF8\\_OPENING\\_GUILLEMET](#), [UTF8\\_PERIOD](#), [UTF8\\_PIPE](#), [UTF8\\_QUESTION](#),  
[UTF8\\_REGISTERED\\_MARK](#), [UTF8\\_REVERSE\\_QUERY](#), [UTF8\\_RIGHTPAREN](#), [UTF8\\_SERVICE\\_MARK](#),  
[UTF8\\_SINGLEQUOTE](#), [UTF8\\_SPACE](#), [UTF8\\_TRADEMARK](#), [UTF8\\_UNDERSCORE](#), [UTF8\\_UPPER\\_A](#),  
[UTF8\\_UPPER\\_A\\_WITH\\_ACUTE](#), [UTF8\\_UPPER\\_A\\_WITH\\_CIRCUMFLEX](#), [UTF8\\_UPPER\\_A\\_WITH\\_DIAERESIS](#),  
[UTF8\\_UPPER\\_A\\_WITH\\_GRAVE](#), [UTF8\\_UPPER\\_A\\_WITH\\_RING](#), [UTF8\\_UPPER\\_A\\_WITH\\_TILDE](#), [UTF8\\_UPPER\\_B](#),  
[UTF8\\_UPPER\\_C](#), [UTF8\\_UPPER\\_C\\_WITH\\_CEDILLA](#), [UTF8\\_UPPER\\_D](#), [UTF8\\_UPPER\\_E](#),  
[UTF8\\_UPPER\\_E\\_WITH\\_ACUTE](#), [UTF8\\_UPPER\\_E\\_WITH\\_CIRCUMFLEX](#), [UTF8\\_UPPER\\_E\\_WITH\\_DIAERESIS](#),  
[UTF8\\_UPPER\\_E\\_WITH\\_GRAVE](#), [UTF8\\_UPPER\\_F](#), [UTF8\\_UPPER\\_G](#), [UTF8\\_UPPER\\_H](#), [UTF8\\_UPPER\\_I](#),  
[UTF8\\_UPPER\\_I\\_WITH\\_CIRCUMFLEX](#), [UTF8\\_UPPER\\_I\\_WITH\\_DIAERESIS](#), [UTF8\\_UPPER\\_I\\_WITH\\_GRAVE](#),  
[UTF8\\_UPPER\\_I\\_WITH\\_TILDE](#), [UTF8\\_UPPER\\_J](#), [UTF8\\_UPPER\\_K](#), [UTF8\\_UPPER\\_L](#), [UTF8\\_UPPER\\_M](#),  
[UTF8\\_UPPER\\_N](#), [UTF8\\_UPPER\\_O](#), [UTF8\\_UPPER\\_O\\_WITH\\_ACUTE](#), [UTF8\\_UPPER\\_O\\_WITH\\_CIRCUMFLEX](#),  
[UTF8\\_UPPER\\_O\\_WITH\\_DIAERESIS](#), [UTF8\\_UPPER\\_O\\_WITH\\_GRAVE](#), [UTF8\\_UPPER\\_O\\_WITH\\_STROKE](#),  
[UTF8\\_UPPER\\_O\\_WITH\\_TILDE](#), [UTF8\\_UPPER\\_P](#), [UTF8\\_UPPER\\_Q](#), [UTF8\\_UPPER\\_R](#), [UTF8\\_UPPER\\_S](#),  
[UTF8\\_UPPER\\_T](#), [UTF8\\_UPPER\\_U](#), [UTF8\\_UPPER\\_U\\_WITH\\_ACUTE](#), [UTF8\\_UPPER\\_U\\_WITH\\_CIRCUMFLEX](#),  
[UTF8\\_UPPER\\_U\\_WITH\\_DIAERESIS](#), [UTF8\\_UPPER\\_U\\_WITH\\_GRAVE](#), [UTF8\\_UPPER\\_V](#), [UTF8\\_UPPER\\_W](#),  
[UTF8\\_UPPER\\_X](#), [UTF8\\_UPPER\\_Y](#), [UTF8\\_UPPER\\_Z](#), [UTF8\\_YEN](#), [UTF8MAP](#), [UTF8MAP\\_INDEX\\_OFFSET](#)

#### Fields inherited from class com.wowza.wms.timedtext.CEA608.CEA608Constants

[CC\\_CHANNEL\\_CC1](#), [CC\\_CHANNEL\\_CC2](#), [CC\\_CHANNEL\\_CC3](#), [CC\\_CHANNEL\\_CC4](#), [CC\\_TYPE\\_DTVCC\\_PACKET\\_DATA](#),  
[CC\\_TYPE\\_DTVCC\\_PACKET\\_START](#), [CC\\_TYPE\\_NTSC\\_FIELD\\_1](#), [CC\\_TYPE\\_NTSC\\_FIELD\\_2](#)

## Constructor Summary

public	<a href="#">ClosedCaptionCEA608Utils()</a>
--------	--

Methods inherited from class [com.wowza.wms.timedtext-cea608.ClosedCaptionCEA608Utils](#)

[calcParity](#), [calcParity](#), [cea608CharToCharUTF8](#), [cea608ExtendedCharacterToUTF8](#),  
[charUTF8ToCAE608](#), [charUTF8ToCAE608](#), [checkParity](#), [checkParity](#)

Methods inherited from class [java.lang.Object](#)

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

## Constructors

### **ClosedCaptionCEA608Utils**

**public ClosedCaptionCEA608Utils()**

---

**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
+- com.wowza.wms.dvr.DvrManifestEntry
  +- com.wowza.wms.dvr.DvrManifestChunkEntry
```

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

## 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
+- com.wowza.wms.dvr.DvrManifestEntry
  +- com.wowza.wms.dvr.DvrManifestCodecEntry
```

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

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

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

## Fields

### MANIFESTFILE\_KEY\_INDEX

public static final java.lang.String **MANIFESTFILE\_KEY\_INDEX**

Constant value: **index**

### MANIFESTFILE\_KEY\_TYPE

public static final java.lang.String **MANIFESTFILE\_KEY\_TYPE**

Constant value: **type**

### MANIFESTFILE\_KEY\_START

public static final java.lang.String **MANIFESTFILE\_KEY\_START**

Constant value: **start**

### MANIFESTFILE\_KEY\_STOP

public static final java.lang.String **MANIFESTFILE\_KEY\_STOP**

(continued from last page)

Constant value: **stop**

---

**MANIFESTFILE\_KEY\_SIZE**public static final java.lang.String **MANIFESTFILE\_KEY\_SIZE**Constant value: **size**

---

**MANIFESTFILE\_KEY\_ARTIFACT**public static final java.lang.String **MANIFESTFILE\_KEY\_ARTIFACT**Constant value: **artifact**

---

**MANIFESTFILE\_KEY\_DVRTIME**public static final java.lang.String **MANIFESTFILE\_KEY\_DVRTIME**Constant value: **dvrTime**

---

**MANIFESTFILE\_KEY\_CHUNKINDEX**public static final java.lang.String **MANIFESTFILE\_KEY\_CHUNKINDEX**Constant value: **chunkIndex**

---

**MANIFESTFILE\_KEY\_PACKETTIME**public static final java.lang.String **MANIFESTFILE\_KEY\_PACKETTIME**Constant value: **packetTime**

---

**MANIFESTFILE\_KEY\_UTCTIME**public static final java.lang.String **MANIFESTFILE\_KEY\_UTCTIME**Constant value: **utcTime**

---

**MANIFESTFILE\_KEY\_NAME**public static final java.lang.String **MANIFESTFILE\_KEY\_NAME**Constant value: **name**

---

**MANIFESTFILE\_KEY\_AUDIO\_CODEC**public static final java.lang.String **MANIFESTFILE\_KEY\_AUDIO\_CODEC**Constant value: **aCodec**

(continued from last page)

## MANIFESTFILE\_KEY\_VIDEO\_CODEC

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

Constant value: **vCodec**

---

## MANIFESTFILE\_KEY\_ENCRYPTIONS

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

Constant value: **enc**

---

## MANIFESTFILE\_KEY\_METADATA

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

Constant value: **metadata**

---

## SERIALIZE\_CURRENT\_VERSION

```
protected static final byte SERIALIZE_CURRENT_VERSION
```

Constant value: **2**

---

## dvrStart

```
protected long dvrStart
```

---

## dvrStop

```
protected long dvrStop
```

---

## packetTime

```
protected long packetTime
```

---

## utcTime

```
protected long utcTime
```

---

## index

```
protected int index
```

(continued from last page)

**type**

```
protected int type
```

## Constructors

### DvrManifestEntry

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

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

**Parameters:**

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

## Methods

### getStartTimecode

```
public long getStartTimecode()
```

Get start timecode. In milliseconds, DVR time base.

**Returns:**

start time

### getStopTimecode

```
public long getStopTimecode()
```

Get stop timecode. In milliseconds, DVR time base.

**Returns:**

stop time

### getPacketStartTime

```
public long getPacketStartTime()
```

Get chunk start timecode in packetTime units. In milliseconds.

**Returns:**

packet start time

(continued from last page)

## getUtcStartTime

```
public long getUtcStartTime()
```

Get chunk start timecode in UTC units. In milliseconds.

**Returns:**

UTC start time

---

## getDuration

```
public long getDuration()
```

Get duration

**Returns:**

duration in ms

---

## getType

```
public int getType()
```

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

**Returns:**

type

---

## getIndex

```
public int getIndex()
```

Get manifest index.

**Returns:**

index

---

## serialize

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

Serialize manifest record.

**Parameters:**

out - output stream

---

## serialize

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

Serialize manifest record.

**Returns:**

serialized data

(continued from last page)

## **getManifestRepresentation**

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

Get textual representation of record for textual manifest usage.

**Returns:**

textual representation of entry

---

## **getCommonInitialTextRepString**

```
protected String getCommonInitialTextRepString()
```

---

## **encodeBytes**

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

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

**Parameters:**

bytes - buffer to encode.

**Returns:**

Base64 encoding or "".

## com.wowza.wms.dvr Class DvrManifestEntryFactory

```
java.lang.Object
+-com.wowza.wms.dvr.DvrManifestEntryFactory
```

```
public class DvrManifestEntryFactory
extends Object
```

Deserialize manifest entry blobs.

### Constructor Summary

public	<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

### Constructors

#### DvrManifestEntryFactory

```
public DvrManifestEntryFactory()
```

### Methods

#### deserialize

```
public static DvrManifestEntry deserialize(byte[ ] data)
```

#### deserialize

```
public static DvrManifestEntry deserialize(java.nio.ByteBuffer buffer)
```

## com.wowza.wms.dvr

### Class DvrManifestEntryRange

```
java.lang.Object
+-com.wowza.wms.dvr.DvrManifestEntryRange
```

---

public class **DvrManifestEntryRange**  
extends Object

Class that compactly represents a range of indices of a certain manifest type.

#### Constructor Summary

public	<a href="#">DvrManifestEntryRange</a> (int type, int startIndex, int endIndex)
	Construct a range of a given type.

#### Method Summary

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

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

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

(continued from last page)

## Constructors

### DvrManifestEntryRange

```
public DvrManifestEntryRange(int type,  
                            int startIndex,  
                            int endIndex)
```

Construct a range of a given type.

#### Parameters:

type - manifest type  
startIndex - initial member of range  
endIndex - final member of range

## Methods

### getType

```
public int getType()
```

Get range type

#### Returns:

type.

---

### getRange

```
public IndexRange getRange()
```

Get index range.

#### Returns:

index range

---

### setRange

```
public void setRange(IndexRange range)
```

Set index range.

#### Parameters:

range - index range.

---

### getstartIndex

```
public int getstartIndex()
```

Get start index.

#### Returns:

start index (or -1 if undefined)

---

### getEndIndex

```
public int getEndIndex()
```

Get end index.

(continued from last page)

**Returns:**

end index (or -1 if undefined)

**isInRange**

```
public boolean isInRange(int index)
```

Is index in range.

**Parameters:**

index - index in question.

**Returns:**

true if part of range.

**serialize**

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

Serialize range.

**Returns:**

out buffer containing serialized range.

**deserialize**

```
public void deserialize(byte[] data)
```

Deserialize. Called after no-op constructor.

**Parameters:**

data - buffer to deserialize.

**isEmpty**

```
public boolean isEmpty()
```

**toString**

```
public String toString()
```

## com.wowza.wms.dvr Class DvrManifestEntryRangeGroup

```
java.lang.Object
+-com.wowza.wms.dvr.DvrManifestEntryRangeGroup
```

---

```
public class DvrManifestEntryRangeGroup
extends Object
```

A group of Manifest Ranges. The group may contain ranges of different types.

### Field Summary

public	<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
--

### 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 if 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
+- com.wowza.wms.dvr.DvrManifestEntry
  +- com.wowza.wms.dvr.DvrManifestOnMetadataEntry
```

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, DvrChunkArtifact artifact, byte[] data)
	Constructor

**Method Summary**

String	<a href="#">getArtifactsTextRepresentation()</a>
byte[]	<a href="#">getData()</a> Get onMetadata information.
DvrChunkArtifact	<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](#)

```
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
```

---

### 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, SERIALIZED_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
```

## 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)
DvrManifestEntry	<a href="#"><u>getRecordedEntryByIndex</u></a> (int index)
DvrManifestEntry	<a href="#"><u>getRecordedEntryByTimeKey</u></a> (long t)
DvrManifestEntry	<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="#">calculateChunkGroupTime(long dvrTime)</a>
String	<a href="#">determineChunkGroupIdentifier(long dvrTime)</a>
int	<a href="#">getChunkGroupDuration()</a>
void	<a href="#">setChunkGroupDuration(int chunkGrouping)</a>

### 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(DvrManifestChunkEntry entry, DvrChunk chunk)</a> Provide in-memory cache the opportunity to add the DVR chunk to its cache.
void	<a href="#">init(IDvrStreamStore store)</a> Called to initialize the in-memory cache.
DvrChunk	<a href="#">retrieveRawChunk(DvrManifestChunkEntry entry)</a>

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="#"><b>determineUtcTime</b></a> (long currentUtc, long aChunkStartUtc, long aFirstPacketUtc, <a href="#">AMFPacket</a> aFirstPacket, long vChunkStartUtc, long vFirstPacketUtc, <a href="#">AMFPacket</a> vFirstPacket)
void	<a href="#"><b>init</b></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

All Superinterfaces:  
[IDvrPrivateConstants](#)

---

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_PLAYLIST_REQUEST_DELEGATE</a>  Value: <b>com.wowza.wms.dvr.impl.DvrStartDurationPlaylistRequestDelegate</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="#"><u>DEFAULT_PROPERTY_BREAK_ON PTS</u></a> Default value for DVR Property "dvrChunkBreakOnPTS". Value: <b>true</b>
public static final	<a href="#"><u>DEFAULT_PROPERTY_CHUNK_CACHE_CLASS</u></a> Default value for DVR Property "dvrChunkMemoryCacheClass". Value: <b>com.wowza.wms.dvr.impl.DvrDefaultChunkMemoryCache</b>
public static final	<a href="#"><u>DEFAULT_PROPERTY_CHUNK_DURATION_MINIMUM</u></a> Default value for DVR Property "dvrChunkDurationMinimum". Value: <b>1500</b>
public static final	<a href="#"><u>DEFAULT_PROPERTY_CHUNK_GROUPING_SECONDS</u></a> Default value for DVR Property "dvrChunkGroupingSeconds". Value: <b>600</b>
public static final	<a href="#"><u>DEFAULT_PROPERTY_CHUNK_READER_CLASS</u></a> Default value for DVR Property "dvrChunkReaderClass". Value: <b>com.wowza.wms.dvr.impl.io.DvrFileChunkReader</b>
public static final	<a href="#"><u>DEFAULT_PROPERTY_CHUNK_WRITER_CLASS</u></a> Default value for DVR Property "dvrChunkWriterClass". Value: <b>com.wowza.wms.dvr.impl.io.DvrFileChunkWriter</b>
public static final	<a href="#"><u>DEFAULT_PROPERTY_DEBUG_MAX_INVALID_CHUNKS_LOGGED</u></a> Default value for DVR Property "dvrMaxInvalidChunksLogged". Value: <b>10</b>
public static final	<a href="#"><u>DEFAULT_PROPERTY_DEBUG_MAX_RAW_PACKETS</u></a> Default value for DVR Property "dvrDebugMaximumRawPackets". Value: <b>200</b>
public static final	<a href="#"><u>DEFAULT_PROPERTY_DEBUG_MAX_VALID_CHUNKS_LOGGED</u></a> Default value for DVR Property "dvrMaxValidChunksLogged". Value: <b>10</b>
public static final	<a href="#"><u>DEFAULT_PROPERTY_DEBUG_RAW_PACKETS</u></a> Default value for DVR Property "dvrDebugRawPackets". Value: <b>false</b>
public static final	<a href="#"><u>DEFAULT_PROPERTY_DVR_MAX_CHUNK_LOG</u></a> Default value for DVR Property "dvrMaxChunkLogCount". Value: <b>10</b>
public static final	<a href="#"><u>DEFAULT_PROPERTY_FILE_SYSTEM_CLASS</u></a> Default value for DVR Property "dvrFileSystemClass". Value: <b>com.wowza.wms.dvr.impl.io.DvrDefaultFileSystem</b>
public static final	<a href="#"><u>DEFAULT_PROPERTY_MANIFEST_PERSISTENT_CLASS</u></a> Default value for DVR Property "dvrManifestPersisterClass". Value: <b>com.wowza.wms.dvr.impl.DvrManifestPersister</b>
public static final	<a href="#"><u>DEFAULT_PROPERTY_MAX_RECALC_DURATION_LOG</u></a> Default value for DVR Property "dvrMaxRecountDurationLogCount". Value: <b>30</b>

public static final	<a href="#"><u>DEFAULT_PROPERTY_MAX_RECORDING_LENGTH</u></a> Default value for DVR Property "dvrMaximumRecordingLength". Value: <b>108000</b>
public static final	<a href="#"><u>DEFAULT_PROPERTY_MBR_ALTERNATIVE_MATCH_DELTA</u></a> Default value for DVR Property "dvrMbrAlternateMatchDelta". Value: <b>1000</b>
public static final	<a href="#"><u>DEFAULT_PROPERTY_MBR_MINIMUM_PACKETTIME_GAP_SIZE</u></a> Default value for DVR Property "dvrMbrMinimumPacketTimeGapSize". Value: <b>100</b>
public static final	<a href="#"><u>DEFAULT_PROPERTY_MBR_MINIMUM_UTCTIME_GAP_SIZE</u></a> Default value for DVR Property "dvrMbrMinimumPacketTimeGapSize". Value: <b>750</b>
public static final	<a href="#"><u>DEFAULT_PROPERTY_MEDIACACHE_READER_CLASS</u></a> Default value for DVR Property "dvrMediaCacheReaderClass". Value: <b>com.wowza.wms.mediocache.impl.MediaCacheRandomAccessReader</b>
public static final	<a href="#"><u>DEFAULT_PROPERTY_PACKET_DELTA_TO_NOTIFY</u></a> Value: <b>200</b>
public static final	<a href="#"><u>DEFAULT_PROPERTY_PACKET_DELTA_TO_RESET_TIME</u></a> Default value for DVR Property "dvrResetTimePacketDelta". Value: <b>200</b>
public static final	<a href="#"><u>DEFAULT_PROPERTY_PACKET_DURATION_MAXIMUM</u></a> Default value for DVR Property "dvrChunkDurationMinimum". Value: <b>5000</b>
public static final	<a href="#"><u>DEFAULT_PROPERTY_PACKET_SORT_TIME</u></a> Default value for DVR Property "dvrPacketSortTime". Value: <b>0</b>
public static final	<a href="#"><u>DEFAULT_PROPERTY_RECORDINGS_LOADER_CLASS</u></a> Default value for DVR Property "dvrRecordingsLoaderClass". Value: <b>com.wowza.wms.dvr.DvrRecordingsLoader</b>
public static final	<a href="#"><u>DEFAULT_PROPERTY_REPEATERS_HEARTBEAT_DURATION</u></a> Value: <b>4000</b>
public static final	<a href="#"><u>DEFAULT_PROPERTY_SANJOSE_ABST_DURATION_TOLERANCE</u></a> Default value for property "dvrSanJosePlaylistAbstDurationTolerance" #see <a href="#"><u>PROPERTY_SANJOSE_ABST_DURATION_TOLERANCE</u></a> Value: <b>50</b>
public static final	<a href="#"><u>DEFAULT_PROPERTY_SANJOSE_ABST_TIMESCALE</u></a> Default value for property "dvrSanJosePlaylistAbstTimescale" #see <a href="#"><u>PROPERTY_SANJOSE_ABST_TIMESCALE</u></a> Value: <b>1000</b>
public static final	<a href="#"><u>DEFAULT_PROPERTY_SMOOTH_MANIFEST_LIVE_CAN_PAUSE</u></a> Value: <b>true</b>

public static final	<a href="#"><u>DEFAULT_PROPERTY_SMOOTH_MANIFEST_LIVE_CAN_SEEK</u></a> Value: <b>true</b>
public static final	<a href="#"><u>DEFAULT_PROPERTY_SMOOTH_MANIFEST_MAJOR_VERSION</u></a> Value: <b>2</b>
public static final	<a href="#"><u>DEFAULT_PROPERTY_SMOOTH_MANIFEST_MINOR_VERSION</u></a> Value: <b>1</b>
public static final	<a href="#"><u>DEFAULT_PROPERTY_SMOOTH_MANIFEST_RECORDED_CAN_PAUSE</u></a> Value: <b>true</b>
public static final	<a href="#"><u>DEFAULT_PROPERTY_SMOOTH_MANIFEST_RECORDED_CAN_SEEK</u></a> Value: <b>true</b>
public static final	<a href="#"><u>DEFAULT_PROPERTY_SMOOTH_MANIFEST_RECORDED_SPECIFY_DURATION</u></a> Value: <b>true</b>
public static final	<a href="#"><u>DEFAULT_PROPERTY_STORAGE_DIRECTORY</u></a> Default value for DVR Property "dvrStorageDirectory". Value: <b> \${com.wowza.wms.context.VHostConfigHome} /dvr</b>
public static final	<a href="#"><u>DEFAULT_PROPERTY_TEXT_READER_CLASS</u></a> Default value for DVR Property "dvrTextReaderClass". Value: <b>com.wowza.wms.dvr.impl.io.DvrTextFileReader</b>
public static final	<a href="#"><u>DEFAULT_PROPERTY_TEXT_WRITER_CLASS</u></a> Default value for DVR Property "dvrTextWriterClass". Value: <b>com.wowza.wms.dvr.impl.io.DvrTextFileWriter</b>
public static final	<a href="#"><u>DVR_DEFAULT_FILESTORE</u></a> The default DVR store ID: "dvrfilestorage". Value: <b>dvrfilestorage</b>
public static final	<a href="#"><u>DVR_DEFAULT_RECORDER_ID</u></a> The default DVR recorder ID: "dvrrecorder". Value: <b>dvrrecorder</b>
public static final	<a href="#"><u>DVR_REPEAT_PACKETIZER_ID</u></a> The default DVR streaming repeater ID: "dvrstreamingrepeater". Value: <b>dvrstreamingrepeater</b>
public static final	<a href="#"><u>DVR_STREAMING_PACKETIZER_ID</u></a> The default DVR streaming packetizer ID: "dvrstreamingpacketizer". Value: <b>dvrstreamingpacketizer</b>
public static final	<a href="#"><u>DVR_WINDOW_DURATION_UNLIMITED</u></a> Value: <b>0</b>
public static final	<a href="#"><u>MEDIACACHE_PREFIX</u></a> The MediaCache prefix for dvr repeater Value: <b>dvrorigin</b>

public static final	<a href="#"><u>MIMETYPE_VIDEO_MP4</u></a> Constant for mime type "video/mp4" Value: <b>video/mp4</b>
public static final	<a href="#"><u>PROPERTY_ALLOWABLE_AV_PACKET_DELTA</u></a> DVR Property "dvrAllowableAVPacketDelta": for specifying how much audio and video packets may diverge before triggering an error. Value: <b>dvrAllowableAVPacketDelta</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_ALLOW_CACHING</u></a> DVR Property "dvrCupertinoPlaylistAllowCaching": used to force Cupertino playlist to set #EXT-X-ALLOW-CACHE: value Value: <b>dvrCupertinoPlaylistAllowCaching</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 "dvrCupertinoPlaylistForceNonLive": 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_CUPERTINO_PUT_END_TAG_BEFORE_CHUNKS</u></a> DVR Property "dvrCupertinoPutEndTagBeforeChunks": if set to true, puts the EXT-X-ENDLIST before the DVR chunks. Value: <b>dvrCupertinoPutEndTagBeforeChunks</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> HTTP Streamer or 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_DETAILS_WHEN_RECORD_DNE</u></a> HTTP Streamer or DVR Property "dvrDebugMbrDetailsWhenRecordDNE": for turning on DVR MBR Player Adapter debug logging. Value: <b>dvrDebugMbrDetailsWhenRecordDNE</b>
public static final	<a href="#"><u>PROPERTY_DEBUG_MBR_PLAYER_ADAPTER</u></a> HTTP Streamer or DVR Property "dvrDebugMbrPlayerAdapter": for turning on DVR MBR Player Adapter debug logging. Value: <b>dvrDebugMbrPlayerAdapter</b>
public static final	<a href="#"><u>PROPERTY_DEBUG_MBR_PLAYER_ADAPTER_IF_SHIFTED_EMPTY</u></a> HTTP Streamer or DVR Property "dvrDebugMbrPlayerAdapter": for turning on DVR MBR Player Adapter debug logging when null shifted entries is returned. Value: <b>dvrDebugMbrPlayerAdapterWhenShiftedEmpty</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	<u><a href="#">PROPERTY_DEBUG_PLAYLIST_REQUEST</a></u> DVR Property "dvrDebugPlaylistRequest": for turning on logging of DVR playlist requests. Value: <b>dvrDebugPlaylistRequest</b>
public static final	<u><a href="#">PROPERTY_DEBUG_RAW_PACKETS</a></u> DVR Property "dvrDebugRawPackets": for turning on logging of incoming raw packets. Value: <b>dvrDebugRawPackets</b>
public static final	<u><a href="#">PROPERTY_DEBUG_RAW_PACKETS_MATCHER</a></u> DVR Property "dvrDebugRawPacketsMatcher": for matching stream names that will dump raw packet. Value: <b>dvrDebugRawPacketsMatcher</b>
public static final	<u><a href="#">PROPERTY_DEBUG_REPEATERS</a></u> DVR Property "dvrDebugRepeater": for turning on logging of DVR repeater Value: <b>dvrDebugRepeater</b>
public static final	<u><a href="#">PROPERTY_DEBUG_SANJOSE_PLAYER_ADAPTER</a></u> HTTP Streamer Property "dvrDebugSanJosePlayerAdapter": for turning on DVR Player San Jose Adapter debug logging. Value: <b>dvrDebugSanJosePlayerAdapter</b>
public static final	<u><a href="#">PROPERTY_DEBUG_SMOOTH_PLAYER_ADAPTER</a></u> HTTP Streamer Property "dvrDebugSmoothPlayerAdapter": for turning on DVR Player Smooth Adapter debug logging. Value: <b>dvrDebugSmoothPlayerAdapter</b>
public static final	<u><a href="#">PROPERTY_DEBUG_STATE_CHANGE</a></u> DVR Property "dvrDebugStateChange": for logging state changes of DVR store. Value: <b>dvrDebugStateChange</b>
public static final	<u><a href="#">PROPERTY_DEBUG_TOSSED_HOLDERS</a></u> DVR Property "dvrDebugTossedHolders": for turning on logging of packets that are being tossed. Value: <b>dvrDebugTossedHolders</b>
public static final	<u><a href="#">PROPERTY_ENCRYPTION_INFO_DELEGATE</a></u> DVR Property "dvrPlaylistEncryptionInfoDelegate": used to over-ride the encryption info on the playback side. Value: <b>dvrPlaylistEncryptionInfoDelegate</b>
public static final	<u><a href="#">PROPERTY_FILE_SYSTEM_CLASS</a></u> DVR Property "dvrFileSystemClass": for controlling the class responsible for managing the DVR File System. Value: <b>dvrFileSystemClass</b>
public static final	<u><a href="#">PROPERTY_MANIFEST_PERSISTENT_CLASS</a></u> DVR Property "dvrManifestPersisterClass": for controlling the class responsible for persisting the manifest files. Value: <b>dvrManifestPersisterClass</b>
public static final	<u><a href="#">PROPERTY_MAX_CHUNK_LOG</a></u> 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_ALTERNATIVE_MATCH_DELTA</u></a> DVR Property "dvrMbrAlternateMatchDelta": when finding alternative for a MBR gap, an alternate is considered a match if it is within this many ms of the gap in question. Value: <b>dvrMbrAlternateMatchDelta</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="#"><u>PROPERTY_MEDIACACHE_READER_CLASS</u></a> DVR Property "dvrMediaCacheReaderClass" Value: <b>dvrMediaCacheReaderClass</b>
public static final	<a href="#"><u>PROPERTY_PACKET_DELTA_TO_NOTIFY</u></a> Value: <b>dvrPacketDeltaToNotify</b>
public static final	<a href="#"><u>PROPERTY_PACKET_DELTA_TO_RESET_TIME</u></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="#"><u>PROPERTY_PACKET_DURATION_MAXIMUM</u></a> DVR Property "dvrPacketDurationMaximum": for maximum packet duration, in milliseconds. Value: <b>dvrPacketDurationMaximum</b>
public static final	<a href="#"><u>PROPERTY_PACKET_SORT_TIME</u></a> DVR Property "dvrPacketSortTime": for sorting incoming packets. Value: <b>dvrPacketSortTime</b>

public static final	<a href="#"><u>PROPERTY_PLAYLIST_REQUEST_DELEGATE</u></a> DVR Property "dvrPlaylistRequestDelegate": used to over-ride the class that generates a playlist request. Value: <b>dvrPlaylistRequestDelegate</b>
public static final	<a href="#"><u>PROPERTY_RECORD_AUDIO</u></a> DVR Property "recordAudio": for determining if audio should be recorded. Value: <b>recordAudio</b>
public static final	<a href="#"><u>PROPERTY_RECORD_DATA</u></a> DVR Property "recordData": for determining if data should be recorded. Value: <b>recordData</b>
public static final	<a href="#"><u>PROPERTY_RECORD_VIDEO</u></a> DVR Property "recordVideo": for determining if video should be recorded. Value: <b>recordVideo</b>
public static final	<a href="#"><u>PROPERTY_RECORDINGS_LOADER_CLASS</u></a> DVR Property "dvrRecordingsLoaderClass": for controlling the class responsible for loading DVR recordings. Value: <b>dvrRecordingsLoaderClass</b>
public static final	<a href="#"><u>PROPERTY_REPEATERT_HEARTBEAT_DURATION</u></a> DVR Property "dvrRepeaterHeartbeatDuration": for time in ms that origin pings edges Value: <b>dvrRepeaterHeartbeatDuration</b>
public static final	<a href="#"><u>PROPERTY_REPEATERT_SHARED_SECRET</u></a> DVR Property "dvrEncryptionSharedSecret": for encryption shared secret between origins and edges. Value: <b>dvrEncryptionSharedSecret</b>
public static final	<a href="#"><u>PROPERTY_SANJOSE_ABST_DURATION_TOLERANCE</u></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 if 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 if 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 if 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	<u><a href="#">PROPERTY_STORAGE_DIRECTORY</a></u> DVR Property "dvrStorageDirectory": for overriding the DVR storage directory location Typically this is defined application-wide in Application.xml under Application/DVR/StorageDirectory. Value: <b>dvrStorageDirectory</b>
public static final	<u><a href="#">PROPERTY_STREAM_DATA</a></u> Value: <b>dvrStreamRecordedData</b>
public static final	<u><a href="#">PROPERTY_TEXT_READER_CLASS</a></u> DVR Property "dvrTextReaderClass": for controlling the class responsible for reading DVR text files. Value: <b>dvrTextReaderClass</b>
public static final	<u><a href="#">PROPERTY_TEXT_WRITER_CLASS</a></u> DVR Property "dvrTextWriterClass": for controlling the class responsible for writing DVR text files. Value: <b>dvrTextWriterClass</b>
public static final	<u><a href="#">PROPERTY_WINDOW_DURATION</a></u> DVR Property "dvrWindowDuration": for specifying the DVR window duration, in seconds. Value: <b>dvrWindowDuration</b>
public static final	<u><a href="#">SANJOSE_F4M_STREAMTYPE_STREAMING</a></u> Constant for San Jose streaming type "streaming" Value: <b>streaming</b>
public static final	<u><a href="#">SANJOSE_F4M_STREAMTYPE_DVR</a></u> Constant for San Jose stream type "dvr". Value: <b>dvr</b>
public static final	<u><a href="#">SANJOSE_F4M_STREAMTYPE_LIVE</a></u> Constant for San Jose stream type "live". Value: <b>live</b>
public static final	<u><a href="#">SANJOSE_F4M_STREAMTYPE_LIVEORRECORDED</a></u> Constant for San Jose stream type "liveOrRecorded". Value: <b>liveOrRecorded</b>
public static final	<u><a href="#">SANJOSE_F4M_STREAMTYPE_RECORDED</a></u> Constant for San Jose stream type "recorded". Value: <b>recorded</b>
public static final	<u><a href="#">SANJOSE_F4M_VERSION_1_0</a></u> Constant for San Jose f4m version "1.0". Value: <b>1.0</b>
public static final	<u><a href="#">SANJOSE_F4M_VERSION_2_0</a></u> 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, CHUNK_FOLDER_FORMAT, DEFAULT_PROPERTY_AUDIO_GROUP_COUNT,
DEFAULT_PROPERTY_MANIFEST_PURGE_SIZE, DEFAULT_PROPERTY_MAX_ALLOWABLE_CHUNK_DURATION,
DEFAULT_PROPERTY_MAX_SAVE HOLDER_SIZE, DEFAULT_PROPERTY_MINIMUM_AVAILABLE_CHUNKS,
DEFAULT_PROPERTY_PURGE_CONTROL_CLASS,
DEFAULT_PROPERTY_RECALC_DURATION_MAX_AV_DIFFERENCE_TRIGGER,
DEFAULT_PROPERTY_RECALC_DURATION_TRIGGER_SIZE, DEFAULT_PROPERTY_STREAM_IDLE_TIMEOUT,
DEFAULT_PROPERTY_STREAM_STARTUP_TIMEOUT, DEFAULT_PROPERTY_UTC_ALIGN_TO_AUDIO,
DEFAULT_PROPERTY_UTC_USE_WOWZA_PACKET_ARRIVAL, DEFAULT_PROPERTY_WAIT_FOR_CODEC_TIME,
PROPERTY_AUDIO_GROUP_COUNT, PROPERTY_DEBUG_MBR_RESOLVER,
PROPERTY_DEBUG_MBR_RESOLVER_GAP_DETAILS,
PROPERTY_DEBUG_MBR_RESOLVER_INTERSECTION_DETAILS, PROPERTY_INTERCEPTOR_CHUNK_DELEGATE,
PROPERTY_MANIFEST_PURGE_CACHE_SIZE, PROPERTY_MAX_ALLOWABLE_CHUNK_DURATION,
PROPERTY_MAX_SAVE HOLDER_SIZE, PROPERTY_MBR_USE_SIMPLEALIGNMENT,
PROPERTY_MINIMUM_AVAILABLE_CHUNKS, PROPERTY_PURGE_CONTROL_CLASS,
PROPERTY_RECALC_DURATION_MAX_AV_DIFFERENCE_TRIGGER,
PROPERTY_RECALC_DURATION_TRIGGER_SIZE, PROPERTY_UTC_ALIGN_TO_AUDIO,
PROPERTY_UTC_DELEGATE_CLASS, PROPERTY_UTC_USE_WOWZA_PACKET_ARRIVAL,
PROPERTY_VERIFY_CHUNK_EXISTENCE_ON_MANIFEST_LOAD, PROPERTY_WAIT_FOR_CODEC_TIME
```

---

## Fields

### DVR\_STREAMING\_PACKETIZER\_ID

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

The default DVR streaming packetizer ID: "dvrstreamingpacketizer".

See DVR.xml and Application.xml.

Constant value: **dvrstreamingpacketizer**

---

### DVR\_REPEATERT\_PACKETIZER\_ID

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

The default DVR streaming repeater ID: "dvrstreamingrepeater".

See DVR.xml and Application.xml.

Constant value: **dvrstreamingrepeater**

---

### DVR\_DEFAULT\_RECORDER\_ID

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

The default DVR recorder ID: "dvrrecorder".

See DVR.xml and Application.xml.

Constant value: **dvrrecorder**

---

### DVR\_DEFAULT\_FILESTORE

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

The default DVR store ID: "dvrfilestorage".

See DVR.xml and Application.xml.

Constant value: **dvrfilestorage**

## MEDIACACHE\_PREFIX

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

The MediaCache prefix for dvr repeater

See MediaCache.xml

Constant value: **dvrorigin**

---

## PROPERTY\_CHUNK\_GROUPING\_SECONDS

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

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

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

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

See Also:

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

---

## DEFAULT\_PROPERTY\_CHUNK\_GROUPING\_SECONDS

```
public static final int DEFAULT_PROPERTY_CHUNK_GROUPING_SECONDS
```

Default value for DVR Property "dvrChunkGroupingSeconds".

Default value is 600 seconds (10 minutes).

Constant value: **600**

See Also:

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

---

## PROPERTY\_APPEND\_DISCONTINUITY\_DELTA

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

DVR Property "dvrAppendDiscontinuityDelta": for specifying how many milliseconds of empty space are put between individual recordings that when in append mode.

Valid values are integers greater than or equal to 0. Add the property to Application/DVR/Properties section of Application.xml.

Constant value: **dvrAppendDiscontinuityDelta**

See Also:

[DEFAULT\\_PROPERTY\\_APPEND\\_DISCONTINUITY\\_DELTA](#)

---

## DEFAULT\_PROPERTY\_APPEND\_DISCONTINUITY\_DELTA

```
public static final int DEFAULT_PROPERTY_APPEND_DISCONTINUITY_DELTA
```

Default value for DVR Property "dvrAppendDiscontinuityDelta".

Default value is 0.

Constant value: **0**

See Also:

(continued from last page)

---

[PROPERTY\\_APPEND\\_DISCONTINUITY\\_DELTA](#)

## PROPERTY\_WINDOW\_DURATION

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

DVR Property "dvrWindowDuration": for specifying the DVR window duration, in seconds.

Typically this is defined application-wide in Application.xml under Application/DVR/WindowDuration. However, to override this on a per stream basis, the property may be set on the IDvrStreamManager after creation and before initialization. Valid values are integers greater than or equal to 0. Add the property to Application/DVR/Properties section of Application.xml. The default is 0, meaning the window size is unlimited.

Constant value: **dvrWindowDuration**

See Also:

[DVR\\_WINDOW\\_DURATION\\_UNLIMITED](#)

---

## DVR\_WINDOW\_DURATION\_UNLIMITED

```
public static final int DVR_WINDOW_DURATION_UNLIMITED
```

Constant value: **0**

---

## PROPERTY\_STORAGE\_DIRECTORY

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

DVR Property "dvrStorageDirectory": for overriding the DVR storage directory location. Typically this is defined application-wide in Application.xml under Application/DVR/StorageDirectory. However, to override this on a per stream basis, the property may be set on the IDvrStreamManager after creation and before initialization.

Constant value: **dvrStorageDirectory**

See Also:

[DEFAULT\\_PROPERTY\\_STORAGE\\_DIRECTORY](#)

---

## DEFAULT\_PROPERTY\_STORAGE\_DIRECTORY

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

Default value for DVR Property "dvrStorageDirectory".

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

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

See Also:

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

---

## PROPERTY\_ARCHIVE\_STRATEGY

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

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

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

Constant value: **dvrArchiveStrategy**

See Also:

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

(continued from last page)

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

---

## ARCHIVE\_STRATEGY\_DELETE

public static final java.lang.String **ARCHIVE\_STRATEGY\_DELETE**

Constant value: **delete**

---

## ARCHIVE\_STRATEGY\_VERSION

public static final java.lang.String **ARCHIVE\_STRATEGY\_VERSION**

Constant value: **version**

---

## ARCHIVE\_STRATEGY\_APPEND

public static final java.lang.String **ARCHIVE\_STRATEGY\_APPEND**

Constant value: **append**

---

## PROPERTY\_CHUNK\_READER\_CLASS

public static final java.lang.String **PROPERTY\_CHUNK\_READER\_CLASS**

DVR Property "dvrChunkReaderClass": for controlling the class responsible for reading DVR Chunks.

The class must implement IDvrChunkReader

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

**See Also:**

[com.wowza.wms.dvr.io.IDvrChunkReader](#)  
[DEFAULT\\_PROPERTY\\_CHUNK\\_READER\\_CLASS](#)

---

## DEFAULT\_PROPERTY\_CHUNK\_READER\_CLASS

public static final java.lang.String **DEFAULT\_PROPERTY\_CHUNK\_READER\_CLASS**

Default value for DVR Property "dvrChunkReaderClass".

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

**See Also:**

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

---

## PROPERTY\_CHUNK\_WRITER\_CLASS

public static final java.lang.String **PROPERTY\_CHUNK\_WRITER\_CLASS**

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

The class must implement IDvrChunkWriter

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

(continued from last page)

**See Also:**

[com.wowza.wms.dvr.io.IDvrChunkWriter](#)  
[DEFAULT\\_PROPERTY\\_CHUNK\\_WRITER\\_CLASS](#)

---

## DEFAULT\_PROPERTY\_CHUNK\_WRITER\_CLASS

public static final java.lang.String **DEFAULT\_PROPERTY\_CHUNK\_WRITER\_CLASS**

Default value for DVR Property "dvrChunkWriterClass".

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

**See Also:**

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

---

## PROPERTY\_FILE\_SYSTEM\_CLASS

public static final java.lang.String **PROPERTY\_FILE\_SYSTEM\_CLASS**

DVR Property "dvrFileSystemClass": for controlling the class responsible for managing the DVR File System.

The class must implement [com.wowza.wms.dvr.io.IDvrFileSystem](#) and may sub-class  
[com.wowza.wms.dvr.impl.io.DvrDefaultFileSystem](#)

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

**See Also:**

[com.wowza.wms.dvr.io.IDvrFileSystem](#)  
[DEFAULT\\_PROPERTY\\_FILE\\_SYSTEM\\_CLASS](#)

---

## DEFAULT\_PROPERTY\_FILE\_SYSTEM\_CLASS

public static final java.lang.String **DEFAULT\_PROPERTY\_FILE\_SYSTEM\_CLASS**

Default value for DVR Property "dvrFileSystemClass".

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

**See Also:**

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

---

## PROPERTY\_MANIFEST\_PERSISTER\_CLASS

public static final java.lang.String **PROPERTY\_MANIFEST\_PERSISTER\_CLASS**

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

The class must implement [com.wowza.wms.dvr.io.IDvrManifestPersister](#)

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

**See Also:**

[com.wowza.wms.dvr.io.IDvrManifestPersister](#)  
[DEFAULT\\_PROPERTY\\_MANIFEST\\_PERSISTER\\_CLASS](#)

(continued from last page)

## DEFAULT\_PROPERTY\_MANIFEST\_PERSISTER\_CLASS

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

Default value for DVR Property "dvrManifestPersisterClass".

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

See Also:

[PROPERTY\\_MANIFEST\\_PERSISTER\\_CLASS](#)

---

## PROPERTY\_CHUNK\_CACHE\_CLASS

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

DVR Property "dvrChunkMemoryCacheClass": for controlling the class responsible for caching DVR chunks in memory.

The class must implement [IDvrChunkMemoryCache](#) and may sub-class  
`com.wowza.wms.dvr.impl.DvrDefaultChunkMemoryCache`

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

See Also:

[IDvrChunkMemoryCache](#)

[DEFAULT\\_PROPERTY\\_CHUNK\\_CACHE\\_CLASS](#)

---

## DEFAULT\_PROPERTY\_CHUNK\_CACHE\_CLASS

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

Default value for DVR Property "dvrChunkMemoryCacheClass".

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

See Also:

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

---

## PROPERTY\_RECORDINGS\_LOADER\_CLASS

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

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

The class must implement [IDvrRecordingsLoader](#)

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

See Also:

[IDvrRecordingsLoader](#)

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

---

## DEFAULT\_PROPERTY\_RECORDINGS\_LOADER\_CLASS

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

Default value for DVR Property "dvrRecordingsLoaderClass".

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

---

(continued from last page)

Constant value: **com.wowza.wms.dvr.DvrRecordingsLoader**

See Also:

[PROPERTY\\_RECORDINGS\\_LOADER\\_CLASS](#)

---

## PROPERTY\_TEXT\_WRITER\_CLASS

public static final java.lang.String **PROPERTY\_TEXT\_WRITER\_CLASS**

DVR Property "dvrTextWriterClass": for controlling the class responsible for writing DVR text files.

The class must implement [IDvrTextWriter](#)

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

See Also:

[IDvrTextWriter](#)

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

---

## DEFAULT\_PROPERTY\_TEXT\_WRITER\_CLASS

public static final java.lang.String **DEFAULT\_PROPERTY\_TEXT\_WRITER\_CLASS**

Default value for DVR Property "dvrTextWriterClass".

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

See Also:

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

---

## PROPERTY\_TEXT\_READER\_CLASS

public static final java.lang.String **PROPERTY\_TEXT\_READER\_CLASS**

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

The class must implement [IDvrTextReader](#)

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

See Also:

[IDvrTextReader](#)

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

---

## DEFAULT\_PROPERTY\_TEXT\_READER\_CLASS

public static final java.lang.String **DEFAULT\_PROPERTY\_TEXT\_READER\_CLASS**

Default value for DVR Property "dvrTextReaderClass".

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

See Also:

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

---

(continued from last page)

## PROPERTY\_ALLOWABLE\_AV\_PACKET\_DELTA

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

DVR Property "dvrAllowableAVPacketDelta": for specifying how much audio and video packets may diverge before triggering an error.

Wowza nDVR expects the incoming audio and video to be aligned. Through this parameter, Wowza nDVR will try to compensate for out of alignment issues, but it cannot resolve them. This setting safeguards against audio and data video packets that are not closely aligned. The units are in ms. The default value is 2000 ms. If audio and video are out of alignment by more than this value, the audio and video chunks will be ignored in an attempt to get the streams back into alignment. If you already have an out of alignment issue, increasing the default value is likely to cause more problems. Increasing this value will increase the number of chunks not recorded which may cause the overall quality to be unacceptable.

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

See Also:

[DEFAULT\\_PROPERTY\\_ALLOWABLE\\_AV\\_PACKET\\_DELTA](#)

## DEFAULT\_PROPERTY\_ALLOWABLE\_AV\_PACKET\_DELTA

```
public static final int DEFAULT_PROPERTY_ALLOWABLE_AV_PACKET_DELTA
```

Default value for DVR Property "dvrAllowableAVPacketDelta".

Default value is 2000 ms.  
Constant value: **2000**

See Also:

[PROPERTY\\_ALLOWABLE\\_AV\\_PACKET\\_DELTA](#)

## PROPERTY\_PACKET\_DELTA\_TO\_RESET\_TIME

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

DVR Property "dvrResetTimePacketDelta": for controlling how much Chunk start times must diverge from last chunk's end time before the DVR time gets reset.

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

See Also:

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

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

```
public static final int DEFAULT_PROPERTY_PACKET_DELTA_TO_RESET_TIME
```

Default value for DVR Property "dvrResetTimePacketDelta".

Default value is 200 ms.  
Constant value: **200**

See Also:

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

## PROPERTY\_PACKET\_DELTA\_TO\_NOTIFY

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

(continued from last page)

Constant value: **dvrPacketDeltaToNotify**

---

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

```
public static final int DEFAULT_PROPERTY_PACKET_DELTA_TO_NOTIFY
```

Constant value: **200**

---

## PROPERTY\_RECORD\_DATA

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

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

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

Constant value: **recordData**

---

## PROPERTY\_RECORD\_VIDEO

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

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

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

Constant value: **recordVideo**

---

## PROPERTY\_RECORD\_AUDIO

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

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

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

Constant value: **recordAudio**

---

## PROPERTY\_START\_RECORDING\_ON\_STARTUP

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

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

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

Constant value: **startRecordingOnStartup**

---

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

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

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

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

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

See Also:

(continued from last page)

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

```
public static final int DEFAULT_PROPERTY_AUDIO_ONLY_CHUNK_TARGET_DURATION
```

Default value for DVR Property "dvrAudioOnlyChunkTargetDuration".

Default value is 2000 ms.

Constant value: **2000**

See Also:

[PROPERTY\\_AUDIO\\_ONLY\\_CHUNK\\_TARGET\\_DURATION](#)**PROPERTY\_PACKET\_SORT\_TIME**

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

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

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

Constant value: **dvrPacketSortTime**

See Also:

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

```
public static final int DEFAULT_PROPERTY_PACKET_SORT_TIME
```

Default value for DVR Property "dvrPacketSortTime".

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

Constant value: **0**

See Also:

[PROPERTY\\_PACKET\\_SORT\\_TIME](#)**PROPERTY\_BREAK\_ON PTS**

```
public static final java.lang.String PROPERTY_BREAK_ON PTS
```

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

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

See Also:

[DEFAULT\\_PROPERTY\\_BREAK\\_ON PTS](#)**DEFAULT\_PROPERTY\_BREAK\_ON PTS**

```
public static final boolean DEFAULT_PROPERTY_BREAK_ON PTS
```

Default value for DVR Property "dvrChunkBreakOnPTS".

Default value is true.

Constant value: **true**

See Also:

(continued from last page)

---

[PROPERTY\\_BREAK\\_ON PTS](#)

## PROPERTY\_REPEATERSHARESECRET

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

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\_REPEATERTHEARTBEATDURATION

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

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\_REPEATERTHEARTBEATDURATION

```
public static final int DEFAULT_PROPERTY_REPEATERTHEARTBEATDURATION
```

Constant value: **4000**

---

## PROPERTY\_CHUNKDURATIONMINIMUM

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

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\_CHUNKDURATIONMINIMUM

```
public static final int DEFAULT_PROPERTY_CHUNKDURATIONMINIMUM
```

Default value for DVR Property "dvrChunkDurationMinimum".

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

See Also:

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

---

## PROPERTY\_CHUNKMEMORYCACHESIZE

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

(continued from last page)

DVR Property "chunkMemoryCacheSize": used by DvrDefaultChunkMemoryCache to set number of chunks stored in DVR in-memory cache.

Default value is [DEFAULT\\_CHUNK\\_MEMORY\\_CACHESIZE](#)

Add this to Application/DVR/Properties section of Application.xml  
Constant value: **chunkMemoryCacheSize**

---

## DEFAULT\_CHUNK\_MEMORY\_CACHESIZE

public static final int **DEFAULT\_CHUNK\_MEMORY\_CACHESIZE**

Default value for DVR Property "chunkMemoryCacheSize".

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

See Also:

[PROPERTY\\_CHUNK\\_MEMORY\\_CACHESIZE](#)

---

## PROPERTY\_MBR\_USE\_UTC\_FOR\_ALIGNMENT

public static final java.lang.String **PROPERTY\_MBR\_USE\_UTC\_FOR\_ALIGNMENT**

DVR Property "dvrMbrUseUtcForAlignment": when doing mbr alignment use utc time for alignment

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

---

## PROPERTY\_MBR\_MINIMUM\_PACKETTIME\_GAP\_SIZE

public static final java.lang.String **PROPERTY\_MBR\_MINIMUM\_PACKETTIME\_GAP\_SIZE**

DVR Property "dvrMbrMinimumPacketTimeGapSize": when doing mbr alignment gaps smaller than this are ignored.

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

See Also:

[DEFAULT\\_PROPERTY\\_MBR\\_MINIMUM\\_PACKETTIME\\_GAP\\_SIZE](#)

---

## PROPERTY\_MBR\_ALTERNATIVE\_MATCH\_DELTA

public static final java.lang.String **PROPERTY\_MBR\_ALTERNATIVE\_MATCH\_DELTA**

DVR Property "dvrMbrAlternateMatchDelta": when finding alternative for a MBR gap, an alternate is considered a match if it is within this many ms of the gap in question.

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

See Also:

[DEFAULT\\_PROPERTY\\_MBR\\_ALTERNATIVE\\_MATCH\\_DELTA](#)

---

## DEFAULT\_PROPERTY\_MBR\_ALTERNATIVE\_MATCH\_DELTA

public static final int **DEFAULT\_PROPERTY\_MBR\_ALTERNATIVE\_MATCH\_DELTA**

Default value for DVR Property "dvrMbrAlternateMatchDelta".

Default value is 1000 ms.  
Constant value: **1000**

(continued from last page)

**See Also:**[PROPERTY\\_MBR\\_ALTERNATIVE\\_MATCH\\_DELTA](#)

---

**PROPERTY\_PACKET\_DURATION\_MAXIMUM****public static final java.lang.String PROPERTY\_PACKET\_DURATION\_MAXIMUM**

DVR Property "dvrPacketDurationMaximum": for maximum packet duration, in milliseconds.

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

**See Also:**[DEFAULT\\_PROPERTY\\_PACKET\\_DURATION\\_MAXIMUM](#)

---

**DEFAULT\_PROPERTY\_PACKET\_DURATION\_MAXIMUM****public static final int DEFAULT\_PROPERTY\_PACKET\_DURATION\_MAXIMUM**

Default value for DVR Property "dvrChunkDurationMinimum".

Default value is 5000 ms.  
Constant value: **5000**

**See Also:**[PROPERTY\\_PACKET\\_DURATION\\_MAXIMUM](#)

---

**DEFAULT\_PROPERTY\_MBR\_MINIMUM\_PACKETTIME\_GAP\_SIZE****public static final int DEFAULT\_PROPERTY\_MBR\_MINIMUM\_PACKETTIME\_GAP\_SIZE**

Default value for DVR Property "dvrMbrMinimumPacketTimeGapSize".

Default value is 100 ms.  
Constant value: **100**

**See Also:**[PROPERTY\\_MBR\\_MINIMUM\\_PACKETTIME\\_GAP\\_SIZE](#)

---

**PROPERTY\_MBR\_MINIMUM\_UTCTIME\_GAP\_SIZE****public static final java.lang.String PROPERTY\_MBR\_MINIMUM\_UTCTIME\_GAP\_SIZE**

DVR Property "dvrMbrMinimumUtcTimeGapSize": when doing mbr alignment gaps smaller than this are ignored.

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

**See Also:**[DEFAULT\\_PROPERTY\\_MBR\\_MINIMUM\\_UTCTIME\\_GAP\\_SIZE](#)

---

**DEFAULT\_PROPERTY\_MBR\_MINIMUM\_UTCTIME\_GAP\_SIZE****public static final int DEFAULT\_PROPERTY\_MBR\_MINIMUM\_UTCTIME\_GAP\_SIZE**

Default value for DVR Property "dvrMbrMinimumPacketTimeGapSize".

Default value is 750 ms.  
Constant value: **750**

(continued from last page)

**See Also:**[PROPERTY\\_MBR\\_MINIMUM\\_UTCTIME\\_GAP\\_SIZE](#)

## PROPERTY\_MAX\_RECORDING\_LENGTH

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

DVR Property "dvrMaximumRecordingLength": The maximum recording length in seconds. Recording stops when it reaches this value.

Add the property to Application/DVR/Properties section of Application.xml A value of zero means no maximum recording length is enforced. If used in conjunction with DVRWindow, should be slightly larger (1-2%) than DVRWindow.  
Constant value: **dvrMaximumRecordingLength**

**See Also:**[DEFAULT\\_PROPERTY\\_MAX\\_RECORDING\\_LENGTH](#)

## DEFAULT\_PROPERTY\_MAX\_RECORDING\_LENGTH

```
public static final long DEFAULT_PROPERTY_MAX_RECORDING_LENGTH
```

Default value for DVR Property "dvrMaximumRecordingLength".

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

**See Also:**[PROPERTY\\_MAX\\_RECORDING\\_LENGTH](#)

## PROPERTY\_MEDIACACHE\_ENABLED

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

DVR Property "dvrMediaCacheEnabled"

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

## PROPERTY\_MEDIACACHE\_READER\_CLASS

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

DVR Property "dvrMediaCacheReaderClass"

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

**See Also:**[DEFAULT\\_PROPERTY\\_MEDIACACHE\\_READER\\_CLASS](#)

## DEFAULT\_PROPERTY\_MEDIACACHE\_READER\_CLASS

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

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

**See Also:**[PROPERTY\\_MEDIACACHE\\_READER\\_CLASS](#)

---

(continued from last page)

## PROPERTY\_DEBUG\_LOG\_INVALID\_CHUNK\_DETAILS

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

DVR Property "dvrLogInvalidChunkDetails": to control detailed logging information of DVR invalid chunks

The default is false, meaning do not log invalid chunks. Add the property to Application/DVR/Properties section of Application.xml

Constant value: **dvrLogInvalidChunkDetails**

See Also:

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

---

## PROPERTY\_DEBUG\_LOG\_INVALID\_CHUNK\_MATCHER

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

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

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

Constant value: **dvrLogInvalidChunkMatcher**

See Also:

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

---

## PROPERTY\_DEBUG\_MAX\_INVALID\_CHUNKS\_LOGGED

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

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

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

Constant value: **dvrMaxInvalidChunksLogged**

See Also:

[DEFAULT\\_PROPERTY\\_DEBUG\\_MAX\\_INVALID\\_CHUNKS\\_LOGGED](#)

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

---

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

```
public static final int DEFAULT_PROPERTY_DEBUG_MAX_INVALID_CHUNKS_LOGGED
```

Default value for DVR Property "dvrMaxInvalidChunksLogged".

Default value is 10.

Constant value: **10**

See Also:

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

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

---

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

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

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

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

Constant value: **dvrLogValidChunkDetails**

(continued from last page)

**See Also:**[PROPERTY\\_DEBUG\\_MAX\\_VALID\\_CHUNKS\\_LOGGED](#)

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

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

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

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

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

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

```
public static final int DEFAULT_PROPERTY_DEBUG_MAX_VALID_CHUNKS_LOGGED
```

Default value for DVR Property "dvrMaxValidChunksLogged".

Default value is 10.

Constant value: **10**

**See Also:**[PROPERTY\\_DEBUG\\_LOG\\_VALID\\_CHUNK\\_DETAILS](#)  
[PROPERTY\\_DEBUG\\_MAX\\_VALID\\_CHUNKS\\_LOGGED](#)

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

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

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

Only affects logging if [PROPERTY\\_DEBUG\\_MAX\\_VALID\\_CHUNKS\\_LOGGED](#) is true.  
Constant value: **dvrLogValidChunkMatcher**

**See Also:**[PROPERTY\\_DEBUG\\_MAX\\_VALID\\_CHUNKS\\_LOGGED](#)

## PROPERTY\_DEBUG\_TOSSED\_HOLDERS

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

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

Valid values are "true" or "false". Default is false.  
Constant value: **dvrDebugTossedHolders**

## PROPERTY\_DEBUG\_RAW\_PACKETS

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

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

Valid values are "true" or "false".  
Constant value: **dvrDebugRawPackets**

**See Also:**

(continued from last page)

[PROPERTY\\_DEBUG\\_MAXIMUM\\_RAW\\_PACKETS](#)**DEFAULT\_PROPERTY\_DEBUG\_RAW\_PACKETS****public static final boolean DEFAULT\_PROPERTY\_DEBUG\_RAW\_PACKETS**

Default value for DVR Property "dvrDebugRawPackets".

Default value is false.

Constant value: **false**

See Also:

[PROPERTY\\_DEBUG\\_RAW\\_PACKETS](#)[PROPERTY\\_DEBUG\\_RAW\\_PACKETS\\_MATCHER](#)**PROPERTY\_DEBUG\_RAW\_PACKETS\_MATCHER****public static final java.lang.String PROPERTY\_DEBUG\_RAW\_PACKETS\_MATCHER**

DVR Property "dvrDebugRawPacketsMatcher": for matching stream names that will dump raw packet.

Only affects logging if [PROPERTY\\_DEBUG\\_RAW\\_PACKETS](#) is true.Constant value: **dvrDebugRawPacketsMatcher**

See Also:

[PROPERTY\\_DEBUG\\_RAW\\_PACKETS](#)**PROPERTY\_DEBUG\_MAXIMUM\_RAW\_PACKETS****public static final java.lang.String PROPERTY\_DEBUG\_MAXIMUM\_RAW\_PACKETS**

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

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

Only affects logging if [PROPERTY\\_DEBUG\\_RAW\\_PACKETS](#) is true.Constant value: **dvrDebugMaximumRawPackets**

See Also:

[PROPERTY\\_DEBUG\\_RAW\\_PACKETS](#)**DEFAULT\_PROPERTY\_DEBUG\_MAX\_RAW\_PACKETS****public static final int DEFAULT\_PROPERTY\_DEBUG\_MAX\_RAW\_PACKETS**

Default value for DVR Property "dvrDebugMaximumRawPackets".

Default value is 200 log statements.

Constant value: **200**

See Also:

[PROPERTY\\_DEBUG\\_MAXIMUM\\_RAW\\_PACKETS](#)**PROPERTY\_DEBUG\_STATE\_CHANGE****public static final java.lang.String PROPERTY\_DEBUG\_STATE\_CHANGE**

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

Constant value: **dvrDebugStateChange**

---

(continued from last page)

## PROPERTY\_DEBUG\_CHUNK\_RETRIEVALS

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

DVR Property "dvrDebugChunkRetrievals": for logging each chunk retrieval  
Constant value: **dvrDebugChunkRetrievals**

---

## PROPERTY\_DEBUG\_FAILED\_CHUNK\_RETRIEVALS

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

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

---

## PROPERTY\_MAX\_CHUNK\_LOG

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

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

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

See Also:

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

---

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

```
public static final int DEFAULT_PROPERTY_DVR_MAX_CHUNK_LOG
```

Default value for DVR Property "dvrMaxChunkLogCount".

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

See Also:

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

---

## PROPERTY\_MAX\_RECALC\_DURATION\_LOG

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

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

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

See Also:

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

---

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

```
public static final int DEFAULT_PROPERTY_MAX_RECALC_DURATION_LOG
```

Default value for DVR Property "dvrMaxRecountDurationLogCount".

Default value is 30 logging statements.

(continued from last page)

Constant value: **30****See Also:**[PROPERTY\\_MAX\\_RECALC\\_DURATION\\_LOG](#)

---

## PROPERTY\_DEBUG\_METHODS

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

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

Valid values are "true" or "false".

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

---

## PROPERTY\_DEBUG\_REPEATERS

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

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

Valid values are "true" or "false".

Constant value: **dvrDebugRepeater**

---

## PROPERTY\_DEBUG\_PLAYER\_ADAPTER

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

HTTP Streamer Property "dvrDebugPlayerAdapter": for turning on DVR Player Adapter debug logging for all streamer types.

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

Add this property to Application/HTTPStreamer/Properties.  
Constant value: **dvrDebugPlayerAdapter**

---

## PROPERTY\_DEBUG\_SMOOTH\_PLAYER\_ADAPTER

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

HTTP Streamer Property "dvrDebugSmoothPlayerAdapter": for turning on DVR Player Smooth Adapter debug logging.

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

Add this property to Application/HTTPStreamer/Properties.  
Constant value: **dvrDebugSmoothPlayerAdapter**

---

## PROPERTY\_DEBUG\_SANJOSE\_PLAYER\_ADAPTER

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

HTTP Streamer Property "dvrDebugSanJosePlayerAdapter": for turning on DVR Player San Jose Adapter debug logging.

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

Add this property to Application/HTTPStreamer/Properties.  
Constant value: **dvrDebugSanJosePlayerAdapter**

---

(continued from last page)

## PROPERTY\_DEBUG\_CUPERTINO\_PLAYER\_ADAPTER

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

HTTP Streamer Property "dvrDebugCupertinoPlayerAdapter": for turning on DVR Player Cupertino Adapter debug logging.

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

Add this property to Application/HTTPStreamer/Properties.  
Constant value: **dvrDebugCupertinoPlayerAdapter**

## PROPERTY\_DEBUG\_MBR\_PLAYER\_ADAPTER

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

HTTP Streamer or DVR Property "dvrDebugMbrPlayerAdapter": for turning on DVR MBR Player Adapter debug logging.

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

Add this property to Application/HTTPStreamer/Properties.  
Constant value: **dvrDebugMbrPlayerAdapter**

## PROPERTY\_DEBUG\_MBR\_PLAYER\_ADAPTER\_IF\_SHIFTED\_EMPTY

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

HTTP Streamer or DVR Property "dvrDebugMbrPlayerAdapter": for turning on DVR MBR Player Adapter debug logging when null shifted entries is returned.

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

Add this property to Application/HTTPStreamer/Properties.  
Constant value: **dvrDebugMbrPlayerAdapterWhenShiftedEmpty**

## PROPERTY\_DEBUG\_MBR\_DETAILS\_WHEN\_RECORD\_DNE

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

HTTP Streamer or DVR Property "dvrDebugMbrDetailsWhenRecordDNE": for turning on DVR MBR Player Adapter debug logging.

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

Add this property to Application/HTTPStreamer/Properties.  
Constant value: **dvrDebugMbrDetailsWhenRecordDNE**

## PROPERTY\_DEBUG\_MBR\_ALIGNMENT

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

HTTP Streamer or DVR Property "dvrDebugMbrAlignment": for turning on logging of mbr alignment

Valid values are "true" or "false". Default is false.  
Constant value: **dvrDebugMbrAlignment**

## PROPERTY\_DEBUG\_MBR\_ALIGNMENT\_RESOLUTION

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

---

(continued from last page)

DVR Property "dvrDebugMbrAlignmentResolution": for turning on logging of mbr alignment resolution (requested to actual)

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

Constant value: **dvrDebugMbrAlignmentResolution**

---

## PROPERTY\_DEBUG\_PLAYLIST\_REQUEST

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

DVR Property "dvrDebugPlaylistRequest": for turning on logging of DVR playlist requests.

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

Constant value: **dvrDebugPlaylistRequest**

---

## PROPERTY\_PLAYLIST\_REQUEST\_DELEGATE

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

DVR Property "dvrPlaylistRequestDelegate": used to over-ride the class that generates a playlist request.

The delegate should extend DvrBasePlaylistRequestFactory

If unspecified, the standard DVR-time based delegate [DEFAULT\\_PLAYLIST\\_REQUEST\\_DELEGATE](#) is used

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

Constant value: **dvrPlaylistRequestDelegate**

---

## DEFAULT\_PLAYLIST\_REQUEST\_DELEGATE

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

Constant value: **com.wowza.wms.dvr.impl.DvrStartDurationPlaylistRequestDelegate**

---

## PROPERTY\_ENCRYPTION\_INFO\_DELEGATE

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

DVR Property "dvrPlaylistEncryptionInfoDelegate": used to over-ride the encryption info on the playback side.

The specified delegate should extend com.wowza.wms.dvr.DvrBaseEncryptionInfoDelegate

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

Constant value: **dvrPlaylistEncryptionInfoDelegate**

---

## PROPERTY\_SANJOSE\_PLAYLIST\_MIMETYPE

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

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

Default value is "video/mp4"

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

Constant value: **dvrSanJosePlaylistMimeType**

---

## PROPERTY\_SANJOSE\_PLAYLIST\_DELIVERYTYPE

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

(continued from last page)

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

Default value is "streaming"

Add this to Application/DVR/Properties section of Application.xml  
Constant value: **dvrSanJosePlaylistDeliveryType**

## **PROPERTY\_SANJOSE\_PLAYLIST\_VERSION**

**public static final java.lang.String PROPERTY\_SANJOSE\_PLAYLIST\_VERSION**

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

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

Constant value: **dvrSanJosePlaylistVersion**

**See Also:**

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

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

## **PROPERTY\_SANJOSE\_PLAYLIST\_RECORDED\_STREAMTYPE**

**public static final java.lang.String PROPERTY\_SANJOSE\_PLAYLIST\_RECORDED\_STREAMTYPE**

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

Default value is #SANJOSE\_F4M\_STREAMTYPE\_RECORDED.\*  
Constant value: **dvrSanJosePlaylistRecordedStreamType**

**See Also:**

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

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

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

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

## **PROPERTY\_SANJOSE\_PLAYLIST\_LIVE\_STREAMTYPE**

**public static final java.lang.String PROPERTY\_SANJOSE\_PLAYLIST\_LIVE\_STREAMTYPE**

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

Default value is #SANJOSE\_F4M\_STREAMTYPE\_DVR.  
Constant value: **dvrSanJosePlaylistLiveStreamType**

**See Also:**

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

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

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

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

## **SANJOSE\_F4M\_STREAMTYPE\_DVR**

**public static final java.lang.String SANJOSE\_F4M\_STREAMTYPE\_DVR**

Constant for San Jose stream type "dvr".

(continued from last page)

Constant value: **dvr**

---

## SANJOSE\_F4M\_STREAMTYPE\_RECORDED

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

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

---

## SANJOSE\_F4M\_STREAMTYPE\_LIVE

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

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

---

## SANJOSE\_F4M\_STREAMTYPE\_LIVEORRECORDED

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

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

---

## SANJOSE\_F4M\_VERSION\_1\_0

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

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

---

## SANJOSE\_F4M\_VERSION\_2\_0

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

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

---

## SANJOSE\_F4M\_STREAMINGTYPE\_STREAMING

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

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

---

## MIMETYPE\_VIDEO\_MP4

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

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

---

## PROPERTY\_SANJOSE\_ABST\_TIMESCALE

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

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

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

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

(continued from last page)

Constant value: **dvrSanJosePlaylistAbstTimescale**

---

## PROPERTY\_SANJOSE\_ABST\_DURATION\_TOLERANCE

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

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

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

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

Constant value: **dvrSanJosePlaylistAbstDurationTolerance**

---

## DEFAULT\_PROPERTY\_SANJOSE\_ABST\_TIMESCALE

```
public static final int DEFAULT_PROPERTY_SANJOSE_ABST_TIMESCALE
```

Default value for property "dvrSanJosePlaylistAbstTimescale" #see [PROPERTY\\_SANJOSE\\_ABST\\_TIMESCALE](#)  
Constant value: **1000**

---

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

```
public static final int DEFAULT_PROPERTY_SANJOSE_ABST_DURATION_TOLERANCE
```

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

---

## PROPERTY\_CUPERTINO\_PLAYLIST\_ALLOW\_CACHING

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

DVR Property "dvrCupertinoPlaylistAllowCaching": used to force Cupertino playlist to set #EXT-X-ALLOW-CACHE: value

Default value is false

Add this to Application/DVR/Properties section of Application.xml  
Constant value: **dvrCupertinoPlaylistAllowCaching**

---

## PROPERTY\_CUPERTINO\_PUT\_END\_TAG\_BEFORE\_CHUNKS

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

DVR Property "dvrCupertinoPutEndTagBeforeChunks": if set to true, puts the EXT-X-ENDLIST before the DVR chunks.

Default value is false

Add this to Application/DVR/Properties section of Application.xml  
Constant value: **dvrCupertinoPutEndTagBeforeChunks**

---

## PROPERTY\_SMOOTH\_MANIFEST\_LIVE\_CAN\_SEEK

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

---

(continued from last page)

DVR Property "dvrSmoothManifestLiveCanSeek": used to determine if 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 if 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 if CanSeek is enabled in smooth Manifest.

Default value is #DEFAULT\_PROPERTY\_SMOOTH\_MANIFEST\_RECORDED\_CAN\_SEEK, which is set to true.

Add this to Application/DVR/Properties section of Application.xml  
Constant value: **dvrSmoothManifestRecordedCanSeek**

---

## **DEFAULT\_PROPERTY\_SMOOTH\_MANIFEST\_RECORDED\_CAN\_SEEK**

```
public static final boolean DEFAULT_PROPERTY_SMOOTH_MANIFEST_RECORDED_CAN_SEEK
```

Constant value: **true**

---

## **PROPERTY\_SMOOTH\_MANIFEST\_RECORDED\_CAN\_PAUSE**

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

DVR Property "dvrSmoothManifestRecordedCanPause": used to determine if CanPause is enabled in smooth Manifest.

Default value is #DEFAULT\_PROPERTY\_SMOOTH\_MANIFEST\_RECORDED\_CAN\_PAUSE, which is set to true.

Add this to Application/DVR/Properties section of Application.xml  
Constant value: **dvrSmoothManifestRecordedCanPause**

---

(continued from last page)

## DEFAULT\_PROPERTY\_SMOOTH\_MANIFEST\_RECORDED\_CAN\_PAUSE

```
public static final boolean DEFAULT_PROPERTY_SMOOTH_MANIFEST_RECORDED_CAN_PAUSE
```

Constant value: **true**

## PROPERTY\_SMOOTH\_MANIFEST\_RECORDED\_SPECIFY\_DURATION

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

DVR Property "dvrSmoothManifestRecordedSpecifyDuration": used to determine if duration is specified. If not duration of zero is shown.

Default value is #DEFAULT\_PROPERTY\_SMOOTH\_MANIFEST\_RECORDED\_SPECIFY\_DURATION, which is set to true.

Add this to Application/DVR/Properties section of Application.xml  
Constant value: **dvrSmoothManifestRecordedSpecifyDuration**

## DEFAULT\_PROPERTY\_SMOOTH\_MANIFEST\_RECORDED\_SPECIFY\_DURATION

```
public static final boolean DEFAULT_PROPERTY_SMOOTH_MANIFEST_RECORDED_SPECIFY_DURATION
```

Constant value: **true**

## PROPERTY\_SMOOTH\_MANIFEST\_MAJOR\_VERSION

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

DVR Property "dvrSmoothManifestMajorVersion": used to determine Smooth manifest major version

Default value is #DEFAULT\_PROPERTY\_SMOOTH\_MANIFEST\_MAJOR\_VERSION.

Add this to Application/DVR/Properties section of Application.xml  
Constant value: **dvrSmoothManifestMajorVersion**

## DEFAULT\_PROPERTY\_SMOOTH\_MANIFEST\_MAJOR\_VERSION

```
public static final int DEFAULT_PROPERTY_SMOOTH_MANIFEST_MAJOR_VERSION
```

Constant value: **2**

## PROPERTY\_SMOOTH\_MANIFEST\_MINOR\_VERSION

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

DVR Property "dvrSmoothManifestMinorVersion": used to determine Smooth manifest major version

Default value is #DEFAULT\_PROPERTY\_SMOOTH\_MANIFEST\_MINOR\_VERSION.

Add this to Application/DVR/Properties section of Application.xml  
Constant value: **dvrSmoothManifestMinorVersion**

## DEFAULT\_PROPERTY\_SMOOTH\_MANIFEST\_MINOR\_VERSION

```
public static final int DEFAULT_PROPERTY_SMOOTH_MANIFEST_MINOR_VERSION
```

(continued from last page)

Constant value: **1**

---

## PROPERTY\_SMOOTH\_MANIFEST\_VERBOSE\_DURATION

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

DVR Property "dvrSmoothManifestVerboseDuration": used to force Smooth Manifest to include durations for each record

Default value is false

Add this to Application/DVR/Properties section of Application.xml  
Constant value: **dvrSmoothManifestVerboseDuration**

---

## PROPERTY\_SMOOTH\_MANIFEST\_H264\_CODEC

```
public static final java.lang.String PROPERTY_SMOOTH_MANIFEST_H264_CODEC
```

DVR Property "dvrSmoothManifestH264Codec": used to force Smooth Manifest to use this as its FourCC H264 codec info.

Default value is set by Smooth Streaming

Add this to Application/DVR/Properties section of Application.xml  
Constant value: **dvrSmoothManifestH264Codec**

---

## PROPERTY\_STREAM\_DATA

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

Constant value: **dvrStreamRecordedData**

**com.wowza.wms.dvr****Class IDvrConstants.DvrTimeScale**

```
java.lang.Object
  +-java.lang.Enum
    +-com.wowza.wms.dvr.IDvrConstants.DvrTimeScale
```

**All Implemented Interfaces:**

java.io.Serializable, Comparable

public static final class **IDvrConstants.DvrTimeScale**

extends Enum

**Field Summary**

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

**Methods inherited from class java.lang.Enum**

clone, compareTo, equals, finalize, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

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

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

**Methods inherited from interface java.lang.Comparable**

compareTo

**Fields**

(continued from last page)

## DVR\_TIME

```
public static final com.wowza.wms.dvr.IDvrConstants.DvrTimeScale DVR_TIME
```

---

## PACKET\_TIME

```
public static final com.wowza.wms.dvr.IDvrConstants.DvrTimeScale PACKET_TIME
```

---

## UTC\_TIME

```
public static final com.wowza.wms.dvr.IDvrConstants.DvrTimeScale UTC_TIME
```

## Methods

### values

```
public static IDvrConstants.DvrTimeScale\[\] values()
```

---

### valueOf

```
public static IDvrConstants.DvrTimeScale valueOf(String name)
```

## com.wowza.wms.dvr Interface IDvrEncryptionInfo

public interface **IDvrEncryptionInfo**  
extends Cloneable

Interface for persisting encryption information for DVR.

### Field Summary

public static final	<a href="#">ENCRYPTION_TYPE_CENC</a>
	Common Encryption encryption type. Value: <b>3</b>
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.

(continued from last page)

Constant value: **2**

## ENCRYPTION\_TYPE\_CENC

```
public static final int ENCRYPTION_TYPE_CENC
```

Common Encryption encryption type.  
Constant value: **3**

## Methods

### getSerializeSize

```
public int getSerializeSize()
```

Return number of bytes required to serialize this encryption info.

**Returns:**

number of bytes to serialize.

### getEncryptionType

```
public int getEncryptionType()
```

Serialize encryption info.

The encryption should consist of the following:

- version (1 byte)
- encryption type (1 byte)
- length of 'blob' (2 bytes)
- byte blob (n bytes)

**Returns:**

encryption type (constants defined in this class)

### serialize

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

Serialize the encryption info into buffer starting at position 'pos'.

**Parameters:**

buffer - buffer to serialize to  
pos - position to start serializing

**Returns:**

new position into buffer

### serialize

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

Serialize the encryption info, returning a byte buffer containing the serialized data.

**Returns:**

(continued from last page)

byte buffer of serialized data

---

**clone**

```
public Object clone()  
throws CloneNotSupportedException
```

## com.wowza.wms.dvr Interface IDvrManifest

public interface **IDvrManifest**  
extends

Interface for a DVR Store manifest. A DVR manifest collects type-specific manifest info, including:

- Audio manifest
- Video manifest
- data manifest
- time map manifest, to map DVR, packet and real time
- codec manifest to track codec information
- onMetadata manifest

The difference between retrieving 'recorded entries' and 'live entries' is an important one. Recorded Entries means every current manifest entry. Live Entries refers to a subset of these because a playlist will not include the very last entries, as some chunks must be available for buffering.

### Field Summary

public static final	<a href="#"><u>CODEC_TYPE</u></a>
	Constant for codec manifest type. Value: <b>999</b>
public static final	<a href="#"><u>MANIFEST_TAGNAME_CAN_PLAY</u></a>
	Value: <b>CanPlay</b>
public static final	<a href="#"><u>MANIFEST_TAGNAME_CAN_RECORD</u></a>
	Value: <b>CanRecord</b>
public static final	<a href="#"><u>MANIFEST_TAGNAME_CHUNK_GROUPING</u></a>
	Value: <b>ChunkGrouping</b>
public static final	<a href="#"><u>MANIFEST_TAGNAME_CURRENT_TIME</u></a>
	Value: <b>CurrentTime</b>
public static final	<a href="#"><u>MANIFEST_TAGNAME_HAS_ENCRYPTION</u></a>
	Value: <b>HasEncryption</b>
public static final	<a href="#"><u>MANIFEST_TAGNAME_PURGE_TIME</u></a>
	Value: <b>PurgeTime</b>
public static final	<a href="#"><u>ON_METADATA_TYPE</u></a>
	Constant for 'onMetadata' manifest type. Value: <b>0</b>
public static final	<a href="#"><u>TIME_MAP_TYPE</u></a>
	Constant for time map manifest type. Value: <b>998</b>

## Method Summary

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

int	<a href="#"><u>getNextTimeMapIndex()</u></a>
int	<a href="#"><u>getNumberLiveEntries(int type, long dvrStart)</u></a> Given manifest type and DVR time, get the number of live entries at the given time.
int	<a href="#"><u>getNumberLiveEntries(int type, long dvrStart, long dvrEnd)</u></a> Given manifest type and DVR time, get the number of live entries at the given time.
int	<a href="#"><u>getNumberRecordedEntries(int type, long dvrStart)</u></a> Given manifest type and DVR time, get the number of entries at the given time.
int	<a href="#"><u>getNumberRecordedEntries(int type, long dvrStart, long dvrEnd)</u></a> Given manifest type and DVR time, get the number of live entries at the given time.
long	<a href="#"><u>getRecordedDuration(int type)</u></a> Given manifest type, return DVR recorded duration in seconds
java.util.List	<a href="#"><u>getRecordedEntries(int type)</u></a> Get a copy of all manifest entries of a given type.
java.util.List	<a href="#"><u>getRecordedEntries(int type, long dvrStartTime)</u></a> Given manifest type and DVR time, get a list of recorded entries.
java.util.List	<a href="#"><u>getRecordedEntries(int type, long dvrStartTime, long dvrEndTime)</u></a> Given manifest type and DVR time, get a list of recorded entries.
java.util.List	<a href="#"><u>getRecordedEntriesInRange(DvrManifestEntryRange range)</u></a> Given a DvrManifestEntryRange, get a list of recorded entries in this range.
java.util.Map	<a href="#"><u>getRecordedEntriesMap(int type)</u></a> Get a copy of all manifest entries of a given type as a Map of indices.
java.util.List	<a href="#"><u>getRecordedEntriesWithLimit(int type, long dvrTime, int limit)</u></a> Given manifest type and DVR time, get a list of recorded entries limiting number of returned items.
DvrManifestEntry	<a href="#"><u>getRecordedEntryByIndex(int type, int index)</u></a> Get the manifest entry given a manifest type and an index.
DvrManifestEntry	<a href="#"><u>getRecordedEntryByTimeKey(int type, long dvrTime)</u></a> Get the manifest entry given a manifest type and a time (in DVR units)
IDvrTimeMap	<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="#">hasVideo()</a> Does manifest contain video.
void	<a href="#">importManifest(IDvrManifest manifest, boolean persist)</a> Import the specified manifest into this manifest
void	<a href="#">initialize()</a> Initialize the manifest.
java.util.List	<a href="#">purgeEntries(DvrManifestEntryRangeGroup ranges)</a> Given a group of ranges, purge the manifest entries.
void	<a href="#">refreshManifest()</a> Refresh the manifest.
byte[]	<a href="#">serialize(boolean ignoreEntries)</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 mainfest.

**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, DvrPacketHolder holder) Create audio chunk from set of packets.
DvrChunk	<a href="#">createDvrDataChunk</a> (long dvrTime, int index, int duration, DvrPacketHolder 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, DvrPacketHolder holder) Create video chunk from set of packets.

### Methods

#### createDvrAudioChunk

```
public DvrChunk createDvrAudioChunk(long dvrTime,  
        int index,  
        int duration,  
        DvrPacketHolder holder)
```

Create audio chunk from set of packets.

**Parameters:**

- dvrTime - DVR start time in milliseconds
- index - index of DVR chunk
- duration - Duration in milliseconds
- holder - packet holder

**Returns:**

the created chunk or null if chunk not created.

#### createDvrVideoChunk

```
public DvrChunk createDvrVideoChunk(long dvrTime,  
        int index,  
        int duration,  
        DvrPacketHolder holder)
```

Create video chunk from set of packets.

**Parameters:**

(continued from last page)

dvrTime - DVR start time in milliseconds  
index - index of DVR chunk  
duration - Duration in milliseconds  
holder - packet holder

**Returns:**

the created chunk or null if chunk not created.

---

## createDvrDataChunk

```
public DvrChunk createDvrDataChunk(long dvrTime,  
                                int index,  
                                int duration,  
                                DvrPacketHolder holder)
```

Create data chunk from set of packets.

**Parameters:**

dvrTime - DVR start time in milliseconds  
index - index of DVR chunk  
duration - Duration in milliseconds  
holder - packet holder

**Returns:**

the created chunk or null if chunk not created.

---

## createDvrOnMetadataChunk

```
public DvrChunk createDvrOnMetadataChunk(long dvrTime,  
                                         long pt,  
                                         AMFPacket metaPacket)
```

Create onMetadata chunk from set of packets.

**Parameters:**

dvrTime - DVR start time in milliseconds  
pt - Packet time  
metaPacket - metadata packet

**Returns:**

the created chunk or null if chunk not created.

## com.wowza.wms.dvr Interface IDvrPacketWriter

public interface **IDvrPacketWriter**  
extends

Interface for chunking and storing packets for DVR.

### Method Summary

void	<a href="#">addToChunk(DvrPacketHolder holder)</a>
boolean	<a href="#">canRecordAudio()</a>
boolean	<a href="#">canRecordData()</a>
boolean	<a href="#">canRecordVideo()</a>
void	<a href="#">endChunk(long videoEndTime, long audioEndTime)</a>
void	<a href="#">resetStream()</a>
void	<a href="#">sendOnMetadata(long pt, AMFPacket metaPacket)</a>
void	<a href="#">setCodecInfoAudio(com.wowza.wms.media.model.MediaCodecInfoAudio codecInfoAudio)</a>
void	<a href="#">setCodecInfoVideo(com.wowza.wms.media.model.MediaCodecInfoVideo codecInfoVideo)</a>
void	<a href="#">startChunk(int streamMode, int videoCodecId, int audioCodecId, long timecode, long utcTimecode)</a>

### 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="#">QUERY_PARAM_DVR</a>
	Value: <b>DVR</b>

### Method Summary

int	<a href="#">calcBitrate(IDvrStreamStore store, int type, long t)</a>
String	<a href="#">determineStreamVersion(IDvrStreamManager dvrMgr, String baseStreamName)</a>
DvrPlaylistRequest	<a href="#">getDvrPlaylistRequest(IHTTPStreamerApplicationContext appContext, IDvrStreamStore store, java.util.Map queryMap)</a>
DvrPlaylistRequest	<a href="#">getDvrPlaylistRequest(IHTTPStreamerApplicationContext appContext, java.util.List stores, java.util.Map queryMap)</a>
IDvrMbrPlaylistAlignment	<a href="#">getPlaylistAlignment(IHTTPStreamerSession httpStreamerSession)</a>
boolean	<a href="#">isPlaylistReady(IDvrStreamStore store, DvrPlaylistRequest playlistRequest)</a>
boolean	<a href="#">isPlaylistReady(java.util.List stores, DvrPlaylistRequest dvrPlaylistRequest)</a>

### Fields

#### QUERY\_PARAM\_DVR

public static final java.lang.String **QUERY\_PARAM\_DVR**

Constant value: **DVR**

### Methods

#### isPlaylistReady

public boolean **isPlaylistReady**(IDvrStreamStore store,  
DvrPlaylistRequest playlistRequest)

### isPlaylistReady

```
public boolean isPlaylistReady(java.util.List stores,  
    DvrPlaylistRequest dvrPlaylistRequest)
```

---

### determineStreamVersion

```
public String determineStreamVersion(IDvrStreamManager dvrMgr,  
    String baseStreamName)
```

---

### calcBitrate

```
public int calcBitrate(IDvrStreamStore store,  
    int type,  
    long t)
```

---

### getDvrPlaylistRequest

```
public DvrPlaylistRequest getDvrPlaylistRequest(IHTTPStreamerApplicationContext  
    appContext,  
    IDvrStreamStore store,  
    java.util.Map queryMap)
```

---

### getDvrPlaylistRequest

```
public DvrPlaylistRequest getDvrPlaylistRequest(IHTTPStreamerApplicationContext  
    appContext,  
    java.util.List stores,  
    java.util.Map queryMap)
```

---

### getPlaylistAlignment

```
public IDvrMbrPlaylistAlignment getPlaylistAlignment(IHTTPStreamerSession  
    httpStreamerSession)
```

---

## com.wowza.wms.dvr Interface IDvrPrivateConstants

All Subinterfaces:

[IDvrConstants](#)

public interface **IDvrPrivateConstants**  
extends

Internal and public constants. Public constants are contained in [IDvrConstants](#). Undocumented/Private in this file. To make the constant public (and documented in javadocs, simply move to IDvrConstants)

### Field Summary

public static final	<a href="#">CHUNK_FILENAME_FORMAT_AUDIO</a>  Value: <b>A-%04d_%02d_%02d_%03d.m4fa</b>
public static final	<a href="#">CHUNK_FILENAME_FORMAT_DATA</a>  Value: <b>D-%04d_%02d_%02d_%03d.m4fd</b>
public static final	<a href="#">CHUNK_FILENAME_FORMAT_METADATA</a>  Value: <b>M-%04d_%02d_%02d_%03d.m4fm</b>
public static final	<a href="#">CHUNK_FILENAME_FORMAT_VIDEO</a>  Value: <b>V-%04d_%02d_%02d_%03d.m4fv</b>
public static final	<a href="#">CHUNK_FOLDER_FORMAT</a>  Value: <b>%04d_%02d_%02d</b>
public static final	<a href="#">DEFAULT_PROPERTY_AUDIO_GROUP_COUNT</a>  Default value for DVR Property "dvrAudioGroupCount". Value: <b>3</b>
public static final	<a href="#">DEFAULT_PROPERTY_MANIFEST_PURGE_SIZE</a>  Default value for DVR Property "dvrManifestPurgeCacheSize". Value: <b>10</b>
public static final	<a href="#">DEFAULT_PROPERTY_MAX_ALLOWABLE_CHUNK_DURATION</a>  Default value for DVR Property "dvrMaxAllowableChunkDuration". Value: <b>120000</b>
public static final	<a href="#">DEFAULT_PROPERTY_MAX_SAVE HOLDER_SIZE</a>  Value: <b>500</b>
public static final	<a href="#">DEFAULT_PROPERTY_MINIMUM_AVAILABLE_CHUNKS</a>  Default value for DVR Property "dvrMinimumAvailableChunks". Value: <b>3</b>
public static final	<a href="#">DEFAULT_PROPERTY_PURGE_CONTROL_CLASS</a>  Default value for DVR Property "dvrPurgeControlClass". Value: <b>com.wowza.wms.dvr.impl.DvrPurgeController</b>

public static final	<a href="#"><u>DEFAULT_PROPERTY_RECALC_DURATION_MAX_AV_DIFFERENCE_TRIGGER</u></a> Default value for DVR Property "dvrRecalcDurationMaxAVStartDifference". Value: <b>500</b>
public static final	<a href="#"><u>DEFAULT_PROPERTY_RECALC_DURATION_TRIGGER_SIZE</u></a> Default value for DVR Property "dvrRecalcDurationMaxChunkSize". Value: <b>60000</b>
public static final	<a href="#"><u>DEFAULT_PROPERTY_STREAM_IDLE_TIMEOUT</u></a> Default value for DVR Property "streamTimeout". Value: <b>300000</b>
public static final	<a href="#"><u>DEFAULT_PROPERTY_STREAM_STARTUP_TIMEOUT</u></a> Default value for DVR Property "streamStartupTimeout". Value: <b>60000</b>
public static final	<a href="#"><u>DEFAULT_PROPERTY_UTC_ALIGN_TO_AUDIO</u></a> Default value for DVR Property "dvrUtcAlignToAudio". Value: <b>true</b>
public static final	<a href="#"><u>DEFAULT_PROPERTY_UTC_USE_WOWZA_PACKET_ARRIVAL</u></a> Default value for DVR Property "dvrUtcUseWowzaPacketArrivalTime". Value: <b>false</b>
public static final	<a href="#"><u>DEFAULT_PROPERTY_WAIT_FOR_CODEC_TIME</u></a> Default value for DVR Property "dvrWaitForCodecTime". Value: <b>8000</b>
public static final	<a href="#"><u>PROPERTY_AUDIO_GROUP_COUNT</u></a> DVR Property "dvrAudioGroupCount": for number of audio packets to group together. Value: <b>dvrAudioGroupCount</b>
public static final	<a href="#"><u>PROPERTY_DEBUG_MBR_RESOLVER</u></a> Value: <b>dvrDebugMbrResolver</b>
public static final	<a href="#"><u>PROPERTY_DEBUG_MBR_RESOLVER_GAP_DETAILS</u></a> Value: <b>dvrDebugMbrResolverGapDetails</b>
public static final	<a href="#"><u>PROPERTY_DEBUG_MBR_RESOLVER_INTERSECTION_DETAILS</u></a> Value: <b>dvrDebugMbrResolverIntersectionDetails</b>
public static final	<a href="#"><u>PROPERTY_INTERCEPTOR_CHUNK_DELEGATE</u></a> DVR Property "dvrChunkInterceptorDelegate": for specifying a delegate that determines if DVR chunk should be stored or not. Value: <b>dvrChunkInterceptorDelegate</b>
public static final	<a href="#"><u>PROPERTY_MANIFEST_PURGE_CACHE_SIZE</u></a> DVR Property "dvrManifestPurgeCacheSize": for specifying how many purged chunks are kept around just in case they are requested. Value: <b>dvrManifestPurgeCacheSize</b>
public static final	<a href="#"><u>PROPERTY_MAX_ALLOWABLE_CHUNK_DURATION</u></a> DVR Property "dvrMaxAllowableChunkDuration": for setting maximum chunk duration. Value: <b>dvrMaxAllowableChunkDuration</b>

public static final	<a href="#"><u>PROPERTY_MAX_SAVE HOLDER_SIZE</u></a> Value: <b>dvrMaxSaveHolderSize</b>
public static final	<a href="#"><u>PROPERTY_MBR_USE_SIMPLEALIGNMENT</u></a> HTTP Streamer Property "dvrMbrUseSimpleAligner": for defaulting to a simple mbr aligner Value: <b>dvrMbrUseSimpleAligner</b>
public static final	<a href="#"><u>PROPERTY_MINIMUM_AVAILABLE_CHUNKS</u></a> DVR Property "dvrMinimumAvailableChunks": for specifying how many chunks must be available in order to play back the DVR recording. Value: <b>dvrMinimumAvailableChunks</b>
public static final	<a href="#"><u>PROPERTY_PURGE_CONTROL_CLASS</u></a> DVR Property "dvrPurgeControlClass": for controlling the class responsible for determining which chunks to purge Value: <b>dvrPurgeControlClass</b>
public static final	<a href="#"><u>PROPERTY_RECALC_DURATION_MAX_AV_DIFFERENCE_TRIGGER</u></a> DVR Property "dvrRecalcDurationMaxAVStartDifference": for triggering recalculation of DVR chunk duration by examining each packet Value: <b>dvrRecalcDurationMaxAVStartDifference</b>
public static final	<a href="#"><u>PROPERTY_RECALC_DURATION_TRIGGER_SIZE</u></a> DVR Property "dvrRecalcDurationMaxChunkSize": for triggering recalculation of DVR chunk duration by examining each packet Value: <b>dvrRecalcDurationMaxChunkSize</b>
public static final	<a href="#"><u>PROPERTY_UTCTIME_ALIGN_TO_AUDIO</u></a> DVR Property "dvrUtcAlignToAudio": determines if calculated Utc time gets aligned to audio or video. Value: <b>dvrUtcAlignToAudio</b>
public static final	<a href="#"><u>PROPERTY_UTCTIME_DELEGATE_CLASS</u></a> DVR Property "dvrUtcTimeDelegate": for specifying a delegate that the utc time for a given chunk. Value: <b>dvrUtcTimeDelegate</b>
public static final	<a href="#"><u>PROPERTY_UTCTIME_USE_WOWZA_PACKET_ARRIVAL</u></a> DVR Property "dvrUtcUseWowzaPacketArrivalTime": determines if Utc times are based on the time the packet reaches Wowza or the time the chunks reach DVR (default). Value: <b>dvrUtcUseWowzaPacketArrivalTime</b>
public static final	<a href="#"><u>PROPERTY_VERIFY_CHUNK_EXISTENCE_ON_MANIFEST_LOAD</u></a> Value: <b>dvrVerifyChunkExistenceOnManifestLoad</b>
public static final	<a href="#"><u>PROPERTY_WAIT_FOR_CODEC_TIME</u></a> DVR Property "dvrWaitForCodecTime": for how long DVR will wait to get the codec information. Value: <b>dvrWaitForCodecTime</b>

## Fields

(continued from last page)

## **PROPERTY\_INTERCEPTOR\_CHUNK\_DELEGATE**

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

DVR Property "dvrChunkInterceptorDelegate": for specifying a delegate that determines if DVR chunk should be stored or not. It's only here to allow a backdoor method of testing mbr streams that have gaps.  
Constant value: **dvrChunkInterceptorDelegate**

## **PROPERTY\_UTC\_DELEGATE\_CLASS**

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

DVR Property "dvrUtcTimeDelegate": for specifying a delegate that the utc time for a given chunk. Currently in private constants until article is written.  
Constant value: **dvrUtcTimeDelegate**

## **PROPERTY\_UTC\_ALIGN\_TO\_AUDIO**

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

DVR Property "dvrUtcAlignToAudio": determines if calculated Utc time gets aligned to audio or video. Note: Currently in private constants until article is written.  
Constant value: **dvrUtcAlignToAudio**

## **DEFAULT\_PROPERTY\_UTC\_ALIGN\_TO\_AUDIO**

```
public static final boolean DEFAULT_PROPERTY_UTC_ALIGN_TO_AUDIO
```

Default value for DVR Property "dvrUtcAlignToAudio". Default value is true.  
Constant value: **true**

See Also:

[PROPERTY\\_UTC\\_ALIGN\\_TO\\_AUDIO](#)

## **PROPERTY\_UTC\_USE\_WOWZA\_PACKET\_ARRIVAL**

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

DVR Property "dvrUtcUseWowzaPacketArrivalTime": determines if Utc times are based on the time the packet reaches Wowza or the time the chunks reach DVR (default). Note: Currently in private constants until article is written.  
Constant value: **dvrUtcUseWowzaPacketArrivalTime**

## **DEFAULT\_PROPERTY\_UTC\_USE\_WOWZA\_PACKET\_ARRIVAL**

```
public static final boolean DEFAULT_PROPERTY_UTC_USE_WOWZA_PACKET_ARRIVAL
```

Default value for DVR Property "dvrUtcUseWowzaPacketArrivalTime". Default value is false.  
Constant value: **false**

See Also:

[PROPERTY\\_UTC\\_USE\\_WOWZA\\_PACKET\\_ARRIVAL](#)

## **PROPERTY\_MINIMUM\_AVAILABLE\_CHUNKS**

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

(continued from last page)

DVR Property "dvrMinimumAvailableChunks": for specifying how many chunks must be available in order to play back the DVR recording.

It is not recommended to have this set to less than three, as players typically require this many chunks.

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

See Also:

[DEFAULT\\_PROPERTY\\_MINIMUM\\_AVAILABLE\\_CHUNKS](#)

---

## DEFAULT\_PROPERTY\_MINIMUM\_AVAILABLE\_CHUNKS

public static final int **DEFAULT\_PROPERTY\_MINIMUM\_AVAILABLE\_CHUNKS**

Default value for DVR Property "dvrMinimumAvailableChunks".

Default value is 3.  
Constant value: **3**

See Also:

[PROPERTY\\_MINIMUM\\_AVAILABLE\\_CHUNKS](#)

---

## PROPERTY\_MANIFEST\_PURGE\_CACHE\_SIZE

public static final java.lang.String **PROPERTY\_MANIFEST\_PURGE\_CACHE\_SIZE**

DVR Property "dvrManifestPurgeCacheSize": for specifying how many purged chunks are kept around just in case they are requested.

When purging chunks occurs, the DVR keeps a few chunks around that normally would have been purged just in case they are requested.

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

See Also:

[DEFAULT\\_PROPERTY\\_MANIFEST\\_PURGE\\_SIZE](#)

---

## DEFAULT\_PROPERTY\_MANIFEST\_PURGE\_SIZE

public static final int **DEFAULT\_PROPERTY\_MANIFEST\_PURGE\_SIZE**

Default value for DVR Property "dvrManifestPurgeCacheSize".

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

See Also:

[PROPERTY\\_MANIFEST\\_PURGE\\_CACHE\\_SIZE](#)

---

## PROPERTY\_VERIFY\_CHUNK\_EXISTENCE\_ON\_MANIFEST\_LOAD

public static final java.lang.String **PROPERTY\_VERIFY\_CHUNK\_EXISTENCE\_ON\_MANIFEST\_LOAD**

Constant value: **dvrVerifyChunkExistenceOnManifestLoad**

---

## PROPERTY\_PURGE\_CONTROL\_CLASS

public static final java.lang.String **PROPERTY\_PURGE\_CONTROL\_CLASS**

---

(continued from last page)

DVR Property "dvrPurgeControlClass": for controlling the class responsible for determining which chunks to purge

The class must implement [IDvrPurgeController](#) and may sub-class DvrPurgeControllerBase

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

**See Also:**

[IDvrPurgeController](#)

[DEFAULT\\_PROPERTY\\_PURGE\\_CONTROL\\_CLASS](#)

---

## DEFAULT\_PROPERTY\_PURGE\_CONTROL\_CLASS

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

Default value for DVR Property "dvrPurgeControlClass".

Default value is "com.wowza.wms.dvr.impl.DvrPurgeController".

Constant value: **com.wowza.wms.dvr.impl.DvrPurgeController**

**See Also:**

[PROPERTY\\_PURGE\\_CONTROL\\_CLASS](#)

---

## PROPERTY\_MBR\_USE\_SIMPLEALIGNMENT

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

HTTP Streamer Property "dvrMbrUseSimpleAligner": for defaulting to a simple mbr aligner

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

Constant value: **dvrMbrUseSimpleAligner**

---

## PROPERTY\_MAX\_ALLOWABLE\_CHUNK\_DURATION

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

DVR Property "dvrMaxAllowableChunkDuration": for setting maximum chunk duration.

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

Constant value: **dvrMaxAllowableChunkDuration**

**See Also:**

[DEFAULT\\_PROPERTY\\_MAX\\_ALLOWABLE\\_CHUNK\\_DURATION](#)

---

## DEFAULT\_PROPERTY\_MAX\_ALLOWABLE\_CHUNK\_DURATION

```
public static final int DEFAULT_PROPERTY_MAX_ALLOWABLE_CHUNK_DURATION
```

Default value for DVR Property "dvrMaxAllowableChunkDuration".

Default value is 120000 ms (120 seconds).

Constant value: **120000**

**See Also:**

[PROPERTY\\_MAX\\_ALLOWABLE\\_CHUNK\\_DURATION](#)

---

## PROPERTY\_RECALC\_DURATION\_MAX\_AV\_DIFFERENCE\_TRIGGER

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

(continued from last page)

DVR Property "dvrRecalcDurationMaxAVStartDifference": for triggering recalculation of DVR chunk duration by examining each packet

This is a fallback setting for when the aligned audio and video packets differ their start times. When this happens, fallback logic is triggered which examines the actual packets and tries to adjust the packet times and duration accordingly.

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

**See Also:**

[DEFAULT\\_PROPERTY\\_RECALC\\_DURATION\\_MAX\\_AV\\_DIFFERENCE\\_TRIGGER](#)

---

## DEFAULT\_PROPERTY\_RECALC\_DURATION\_MAX\_AV\_DIFFERENCE\_TRIGGER

public static final int **DEFAULT\_PROPERTY\_RECALC\_DURATION\_MAX\_AV\_DIFFERENCE\_TRIGGER**

Default value for DVR Property "dvrRecalcDurationMaxAVStartDifference".

Default value is 500 ms.  
Constant value: **500**

**See Also:**

[PROPERTY\\_RECALC\\_DURATION\\_MAX\\_AV\\_DIFFERENCE\\_TRIGGER](#)

---

## PROPERTY\_RECALC\_DURATION\_TRIGGER\_SIZE

public static final java.lang.String **PROPERTY\_RECALC\_DURATION\_TRIGGER\_SIZE**

DVR Property "dvrRecalcDurationMaxChunkSize": for triggering recalculation of DVR chunk duration by examining each packet

This is a fallback setting for when the DVR packetizer determines the chunk is quite large, usually because the incoming packet times have done something odd like suddenly diverged. When this happens, some fallback logic is triggered which examines the actual packets and tries to adjust the duration accordingly.

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

**See Also:**

[DEFAULT\\_PROPERTY\\_RECALC\\_DURATION\\_TRIGGER\\_SIZE](#)

---

## DEFAULT\_PROPERTY\_RECALC\_DURATION\_TRIGGER\_SIZE

public static final int **DEFAULT\_PROPERTY\_RECALC\_DURATION\_TRIGGER\_SIZE**

Default value for DVR Property "dvrRecalcDurationMaxChunkSize".

Default value is 60000 ms (60 seconds).  
Constant value: **60000**

**See Also:**

[PROPERTY\\_RECALC\\_DURATION\\_TRIGGER\\_SIZE](#)

---

## PROPERTY\_MAX\_SAVE HOLDER\_SIZE

public static final java.lang.String **PROPERTY\_MAX\_SAVE HOLDER\_SIZE**

Constant value: **dvrMaxSaveHoldersize**

---

(continued from last page)

## DEFAULT\_PROPERTY\_MAX\_SAVE HOLDER\_SIZE

```
public static final int DEFAULT_PROPERTY_MAX_SAVE HOLDER_SIZE
```

Constant value: **500**

---

## PROPERTY\_WAIT\_FOR\_CODEC\_TIME

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

DVR Property "dvrWaitForCodecTime": for how long DVR will wait to get the codec information.

Amount of time Wowza nDVR will wait to receive audio and video codec information after streaming has begun. The unit is in ms. The default value is 8000 ms.

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

Constant value: **dvrWaitForCodecTime**

See Also:

[DEFAULT\\_PROPERTY\\_WAIT\\_FOR\\_CODEC\\_TIME](#)

---

## DEFAULT\_PROPERTY\_WAIT\_FOR\_CODEC\_TIME

```
public static final int DEFAULT_PROPERTY_WAIT_FOR_CODEC_TIME
```

Default value for DVR Property "dvrWaitForCodecTime".

Default value is 8000 ms.

Constant value: **8000**

See Also:

[PROPERTY\\_WAIT\\_FOR\\_CODEC\\_TIME](#)

---

## PROPERTY\_AUDIO\_GROUP\_COUNT

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

DVR Property "dvrAudioGroupCount": for number of audio packets to group together.

Default is 3. @see [DEFAULT\\_PROPERTY\\_AUDIO\\_GROUP\\_COUNT](#)

Constant value: **dvrAudioGroupCount**

---

## DEFAULT\_PROPERTY\_AUDIO\_GROUP\_COUNT

```
public static final int DEFAULT_PROPERTY_AUDIO_GROUP_COUNT
```

Default value for DVR Property "dvrAudioGroupCount".

Default value is 3.

Constant value: **3**

See Also:

[PROPERTY\\_AUDIO\\_GROUP\\_COUNT](#)

---

## DEFAULT\_PROPERTY\_STREAM\_IDLE\_TIMEOUT

```
public static final int DEFAULT_PROPERTY_STREAM_IDLE_TIMEOUT
```

---

(continued from last page)

Default value for DVR Property "streamTimeout".

Default value is 300000 ms (5 minutes).

Constant value: **300000**

---

## DEFAULT\_PROPERTY\_STREAM\_STARTUP\_TIMEOUT

```
public static final int DEFAULT_PROPERTY_STREAM_STARTUP_TIMEOUT
```

Default value for DVR Property "streamStartupTimeout".

Default value is 60000 ms (60 seconds).

Constant value: **60000**

---

## PROPERTY\_DEBUG\_MBR\_RESOLVER

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

Constant value: **dvrDebugMbrResolver**

---

## PROPERTY\_DEBUG\_MBR\_RESOLVER\_GAP\_DETAILS

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

Constant value: **dvrDebugMbrResolverGapDetails**

---

## PROPERTY\_DEBUG\_MBR\_RESOLVER\_INTERSECTION\_DETAILS

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

Constant value: **dvrDebugMbrResolverIntersectionDetails**

---

## CHUNK\_FILENAME\_FORMAT\_METADATA

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

Constant value: **M-%04d\_%02d\_%02d\_%03d.m4fm**

---

## CHUNK\_FILENAME\_FORMAT\_VIDEO

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

Constant value: **V-%04d\_%02d\_%02d\_%03d.m4fv**

---

## CHUNK\_FILENAME\_FORMAT\_AUDIO

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

Constant value: **A-%04d\_%02d\_%02d\_%03d.m4fa**

---

## CHUNK\_FILENAME\_FORMAT\_DATA

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

(continued from last page)

Constant value: **D-%04d\_%02d\_%02d\_%03d.m4fd**

---

## **CHUNK\_FOLDER\_FORMAT**

public static final java.lang.String **CHUNK\_FOLDER\_FORMAT**

Constant value: **%04d\_%02d\_%02d**

## com.wowza.wms.dvr Interface IDvrPurgeController

public interface **IDvrPurgeController**  
extends

Interface for controlling DVR chunk purging. The purge controller is instantiated using a factory class DvrPurgeControllerFactory.

See Also:

[com.wowza.wms.dvr.impl.DvrPurgeControllerFactory](#),  
[IDvrPrivateConstants.PROPERTY\\_PURGE\\_CONTROL\\_CLASS](#)

### Method Summary

long	<a href="#"><b>getCurrentTime()</b></a> Get current DVR time.
long	<a href="#"><b>getLastPurgeTime()</b></a> Get the DVR time when the last purge occurred.
void	<a href="#"><b>init(IDvrStreamStore store)</b></a> Initialize the controller.
boolean	<a href="#"><b>isPurgingEnabled()</b></a> Is Purging enabled for this controller.
void	<a href="#"><b>setCurrentDvrTime(long newDvrTime)</b></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

[retrieveRawChunk\(DvrManifestChunkEntry entry\)](#)

Given the specified manifest entry, return the raw DVR chunk.

### Methods

#### **retrieveRawChunk**

public DvrChunk [retrieveRawChunk\(DvrManifestChunkEntry entry\)](#)

Given the specified manifest entry, return the raw DVR chunk.

**Parameters:**

entry - The DVR manifest entry

**Returns:**

the DVR chunk or null if not able to return the chunk.

## com.wowza.wms.dvr Interface IDvrRecordingListener

public interface **IDvrRecordingListener**  
extends

Listener for DVR Recording events.

See Also:

[IDvrStreamManager.addDvrRecordingListener\(IDvrRecordingListener\)](#),  
[IDvrStreamManager.removeDvrRecordingListener\(IDvrRecordingListener\)](#)

### Method Summary

void	<a href="#"><b>recordingPaused</b>(<u>IDvrStreamStore</u> store)</a> Callback when DVR moves to paused state.
void	<a href="#"><b>recordingReset</b>(<u>IDvrStreamStore</u> store)</a> Callback when DVR gets reset.
void	<a href="#"><b>recordingResumed</b>(<u>IDvrStreamStore</u> store)</a> Callback when DVR moves out of paused state.
void	<a href="#"><b>recordingStarted</b>(<u>IDvrStreamStore</u> store)</a> Callback when DVR moves to recording state.
void	<a href="#"><b>recordingStopped</b>(<u>IDvrStreamStore</u> store)</a> Callback when DVR recording stops
void	<a href="#"><b>timeReset</b>(<u>IDvrStreamStore</u> store, long oldDvrTime, long oldPacketTime, TimeMapRecord newTime)</a> 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="#"><b>init</b>(<b>IDvrStreamManager</b> dvrMgr)</a> Initialize recordings loader
void	<a href="#"><b>loadArchivedRecordings</b>( )</a> Discover and load archived recording
boolean	<a href="#"><b>shouldLoadStream</b>(String streamName, java.util.SortedSet versions)</a> Should the given archived streams be loaded.
boolean	<a href="#"><b>shouldLoadStreamVersion</b>(String streamName, Integer version, java.util.SortedSet versions)</a> Should the given archived stream of specific version be loaded.

### 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 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)

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(IDvrStreamStore store)</a> Callback after a chunk is written to DVR store
void	<a href="#">postChunksPurged(IDvrStreamStore store, DvrManifestEntryRangeGroup entries, java.util.List deletedEntries)</a> Callback after DVR store performs a purge
void	<a href="#">preChunkAdded(IDvrStreamStore store)</a> Callback before a chunk is written to DVR store
void	<a href="#">preChunksPurged(IDvrStreamStore store, DvrManifestEntryRangeGroup entries)</a> 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="#"><b>dvrStreamStorageDeleted</b>(<u>IDvrStreamStore</u> store, <u>IDvrFileSystem</u> fileSystem, boolean success)</a> Callback after DVR store is deleted from disk.
void	<a href="#"><b>dvrStreamStorageLoaded</b>(<u>IDvrStreamManager</u> dvrMgr, <u>IDvrStreamStore</u> store)</a> Callback after DVR store is loaded from disk.
void	<a href="#"><b>dvrStreamStoreCreate</b>(<u>IDvrStreamStore</u> store)</a> Callback after DVR stream store is created.
void	<a href="#"><b>dvrStreamStoreDestroy</b>(<u>IDvrStreamStore</u> store)</a> Callback after DVR stream store is destroyed.
void	<a href="#"><b>dvrStreamStoreInit</b>(<u>IDvrStreamStore</u> store)</a> 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> ( <b>IDvrRecordingListener</b> listener) Add listener to DVR recording events.
void	<a href="#">addDvrStoreListener</a> ( <b>IDvrStoreListener</b> listener) Add listener to DVR store lifecycle events.
void	<a href="#">addManifestEntries</a> (String vStreamName, java.util.List entries, String serverGuid) 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="#">getDefaultValueStreamingStore</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="#"><u>IDvrStreamStore</u></a>	<a href="#"><u>getHighestVersionedStore()</u></a> Return stream store that is highest known version.
void	<a href="#"><u>getInitialRepeaterItems(java.util.List items)</u></a> Get initial repeater items to send to repeater receiver.
int	<a href="#"><u>getMinimumAvailableChunks()</u></a> Return number of chunks that must be available to stream.
String	<a href="#"><u>getPacketizerName()</u></a> Get live stream packetizer name.
<a href="#"><u>IDvrStreamStore</u></a>	<a href="#"><u>getRecordingStreamStore()</u></a> Get current recording store
void	<a href="#"><u>getRepeaterItemsDvr(java.util.List items, long lastSeq)</u></a>
String	<a href="#"><u>getStreamBaseName()</u></a> Get input streamName (no version info).
<a href="#"><u>IDvrStreamStore</u></a>	<a href="#"><u>getStreamStore(String vStreamName)</u></a> Given a stream name containing version info, return the associated stream store.
java.util.List	<a href="#"><u>getStreamStores()</u></a> Get list of all stream stores known to this stream manager.
<a href="#"><u>IDvrStreamVersionHandler</u></a>	<a href="#"><u>getStreamVersionHandler()</u></a> Get the stream version handler object.
void	<a href="#"><u>initialAttributes(String vStreamName, int dvrWindow)</u></a>
void	<a href="#"><u>initialManifest(String vStreamName, IDvrManifest manifest)</u></a>
void	<a href="#"><u>initialManifestEnd(String vStreamName)</u></a>
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.
<a href="#"><u>IDvrStreamStore</u></a>	<a href="#"><u>loadArchivedStore(String vStreamName, DvrManifestHolder manifestHolder)</u></a> Load archived stream store.
void	<a href="#"><u>notifyDvrStreamStorageDeleted(IDvrStreamStore store, IDvrFileSystem fileSystem, boolean success)</u></a>
void	<a href="#"><u>notifyDvrStreamStoreCreate(IDvrStreamStore store)</u></a>
void	<a href="#"><u>notifyDvrStreamStoreDestroy(IDvrStreamStore store)</u></a>
void	<a href="#"><u>notifyDvrStreamStoreInit(IDvrStreamStore store)</u></a>
void	<a href="#"><u>notifyDvrStreamStoreLoaded(IDvrStreamStore store)</u></a>

	void <a href="#">notifyTimeReset(IDvrStreamStore store, long oldDvrTime, long oldPacketTime, TimeMapRecord newTimeMapRecord)</a>
<a href="#">IDvrStreamStore</a>	<a href="#">pauseRecording()</a> Request that stream recording pause.
	void <a href="#">purgeManifestEntries(String vStreamName, DvrManifestEntryRangeGroup rangeGroup)</a> Purge entries from store
	void <a href="#">removeDvrRecordingListener(IDvrRecordingListener listener)</a> Remove listener to DVR recording events.
	void <a href="#">removeDvrStoreListener(IDvrStoreListener listener)</a> Remove listener of DVR store lifecycle events.
	void <a href="#">removeStreamStore(String vStreamName)</a>
	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(IDvrStreamStore store)</a> Set the store to be used for streaming.
	void <a href="#">setRecordingStreamStore(IDvrStreamStore store)</a> Set the store used for recording.
	void <a href="#">setStreamVersionHandler(IDvrStreamVersionHandler handler)</a> Set the stream version handler object.
<a href="#">IDvrStreamStore</a>	<a href="#">startRecording()</a> Request that stream recording start.
	void <a href="#">stateChange(String vStreamName, DvrStreamStoreState state)</a>
<a href="#">IDvrStreamStore</a>	<a href="#">stopRecording()</a> Request that stream recording stop.
DvrChunkStorageInfo	<a href="#">storeChunks(int vDuration, DvrPacketHolder vPackets, int aDuration, DvrPacketHolder aPackets, int dDuration, DvrPacketHolder dPackets)</a>
	boolean <a href="#">storeOnMetadata(long pt, long utc, AMFPacket metaPacket)</a>

**Methods inherited from interface [com.wowza.wms.stream.livepacketizer.ILiveStreamPacketizer](#)**

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

**Methods**

(continued from last page)

## getContextStr

```
public String getContextStr()
```

Get stream context string, useful for logging.

**Returns:**

stream context string

---

## getStreamBaseName

```
public String getStreamBaseName()
```

Get input streamName (no version info).

**Returns:**

unversioned incoming stream name

---

## getStreamStore

```
public IDvrStreamStore getStreamStore(String vStreamName)
```

Given a stream name containing version info, return the associated stream store.

**Parameters:**

vStreamName - versioned stream name

**Returns:**

store or null

---

## getPacketizerName

```
public String getPacketizerName()
```

Get live stream packetizer name.

**Returns:**

live stream packetizer name

---

## getStreamStores

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

Get list of all stream stores known to this stream manager.

**Returns:**

list of stream stores.

---

## getHighestVersionedStore

```
public IDvrStreamStore getHighestVersionedStore()
```

Return stream store that is highest known version.

**Returns:**

stream store

## getMinimumAvailableChunks

```
public int getMinimumAvailableChunks()
```

Return number of chunks that must be available to stream.

**Returns:**

minimum available chunks

---

## setRecordingStreamStore

```
public void setRecordingStreamStore(IDvrStreamStore store)
```

Set the store used for recording.

**Parameters:**

store - stream store, or null

---

## getRecordingStreamStore

```
public IDvrStreamStore getRecordingStreamStore()
```

Get current recording store

**Returns:**

current recording store (may be null)

---

## setDefaultStreamingStore

```
public void setDefaultStreamingStore(IDvrStreamStore store)
```

Set the store to be used for streaming.

**Parameters:**

store - store to stream.

---

## getDefaultStreamingStore

```
public IDvrStreamStore getDefaultStreamingStore()
```

Get the store to be used for streaming.

**Returns:**

store to stream.

---

## getDvrStorageWindowSeconds

```
public int getDvrStorageWindowSeconds()
```

Get DVR window size.

**Returns:**

window size in seconds (0 means infinite window size)

---

## getDvrStorageDir

```
public String getDvrStorageDir()
```

(continued from last page)

Get the storage directory.

**Returns:**

storage directory

---

**getDvrFileSystemClass**

```
public String getDvrFileSystemClass()
```

Get the class used for the DVR file system.

**Returns:**

fully qualified class name

---

**getArchiveStrategy**

```
public String getArchiveStrategy()
```

Get the DVR archive strategy.

**Returns:**

The archive strategy

---

**getEncryptionRepeaterSharedSecret**

```
public String getEncryptionRepeaterSharedSecret()
```

Get the DVR encryption shared secret.

**Returns:**

shared secret.

---

**getEncryptionDelegate**

```
public DvrBaseEncryptionInfoDelegate getEncryptionDelegate()
```

Get the encryption delegate for providing streaming side encryption info objects.

**Returns:**

delegate

---

**getStreamVersionHandler**

```
public IDvrStreamVersionHandler getStreamVersionHandler()
```

Get the stream version handler object.

**Returns:**

stream version handler object.

---

**setStreamVersionHandler**

```
public void setStreamVersionHandler(IDvrStreamVersionHandler handler)
```

Set the stream version handler object. Must be called after the stream manager is created and before it is initialized.

**Parameters:**

handler - stream version handler object.

## resetStream

```
public void resetStream()
```

Reset the stream.

---

## canRecord

```
public boolean canRecord()
```

Is this stream able to record.

**Returns:**

true if this stream is recordable.

---

## isRecording

```
public boolean isRecording()
```

Is this stream currently recording. If this stream is not recordable, the method returns false.

**Returns:**

true if has a recording stream and it is recording

---

## isRecordingPaused

```
public boolean isRecordingPaused()
```

Is this stream currently paused from recording. If this stream is not recording and not paused, the method returns false.

**Returns:**

true if has a recording is paused.

---

## startRecording

```
public IDvrStreamStore startRecording()
```

Request that stream recording start.

Used internally. Clients should call [IILiveStreamDvrRecorder.startRecording\(\)](#). Note that this call places the DVR stream store in the recording state. If the stream store is in the paused state, DVR recording will not occur.

If there is not currently a stream store for recording, one will attempt to be created. A successful start will result in registered [IDvrRecordingListeners](#) to have their [IDvrRecordingListener.recordingStarted\(IDvrStreamStore\)](#) method called.

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

**Returns:**

store if successful. null otherwise.

---

## stopRecording

```
public IDvrStreamStore stopRecording()
```

(continued from last page)

Request that stream recording stop.

Used internally. Clients should call [ILiveStreamDvrRecorder.stopRecording\(\)](#). Note that this call places the DVR stream in the *not* recording state.

A successful stop will result in registered [IDvrRecordingListeners](#) to have their [IDvrRecordingListener.recordingStopped\(IDvrStreamStore\)](#) method called.

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

**Returns:**

store if successful. null otherwise.

---

## pauseRecording

```
public IDvrStreamStore pauseRecording()
```

Request that stream recording pause.

Used internally. Clients should call [ILiveStreamDvrRecorder.pauseRecording\(\)](#). The stream does not have to be actively recording to be paused. For example, it could be paused before packets start flowing.

A successful pause will result in registered [IDvrRecordingListeners](#) to have their [IDvrRecordingListener.recordingPaused\(IDvrStreamStore\)](#) method called.

**Returns:**

store if successful. null otherwise.

---

## resumeRecording

```
public IDvrStreamStore resumeRecording()
```

Request that stream recording resume.

Used internally. Clients should call [ILiveStreamDvrRecorder.resumeRecording\(\)](#). The stream does not have to be actively recording to be resumed. For example, it could have been paused before the stream started, and this call would move it out of the paused state.

A successful resume will result in registered [IDvrRecordingListeners](#) to have their [IDvrRecordingListener.recordingResumed\(IDvrStreamStore\)](#) method called.

**Returns:**

store if successful. null otherwise.

---

## loadArchivedStore

```
public IDvrStreamStore loadArchivedStore(String vStreamName,  
                                         DvrManifestHolder manifestHolder)
```

Load archived stream store.

**Parameters:**

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

manifestHolder - manifest holder containing list of manifest properties

**Returns:**

stream store

---

## deleteArchivedStore

```
public void deleteArchivedStore(String vStreamName)
```

(continued from last page)

Delete archived stream store.

**Parameters:**

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

---

**addManifestEntries**

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

Add to stream stores manifest.

**Parameters:**

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

entries - list of manifest entries.

serverGuid

---

**purgeManifestEntries**

```
public void purgeManifestEntries(String vStreamName,  
    DvrManifestEntryRangeGroup rangeGroup)
```

Purge entries from store

**Parameters:**

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

rangeGroup - set of ranges for purging

---

**storeChunks**

```
public DvrChunkStorageInfo storeChunks(int vDuration,  
    DvrPacketHolder vPackets,  
    int aDuration,  
    DvrPacketHolder aPackets,  
    int dDuration,  
    DvrPacketHolder dPackets)
```

---

**storeOnMetadata**

```
public boolean storeOnMetadata(long pt,  
    long utc,  
    AMFPacket metaPacket)
```

---

**addDvrStoreListener**

```
public void addDvrStoreListener(IDvrStoreListener listener)
```

Add listener to DVR store lifecycle events.

**Parameters:**

listener - listener

---

**removeDvrStoreListener**

```
public void removeDvrStoreListener(IDvrStoreListener listener)
```

---

(continued from last page)

Remove listener of DVR store lifecycle events.

**Parameters:**

listener - listener

---

### **addDvrRecordingListener**

```
public void addDvrRecordingListener(IDvrRecordingListener listener)
```

Add listener to DVR recording events.

**Parameters:**

listener - listener

---

### **removeDvrRecordingListener**

```
public void removeDvrRecordingListener(IDvrRecordingListener listener)
```

Remove listener to DVR recording events.

**Parameters:**

listener - listener

---

### **notifyDvrStreamStoreCreate**

```
public void notifyDvrStreamStoreCreate(IDvrStreamStore store)
```

---

### **notifyDvrStreamStoreInit**

```
public void notifyDvrStreamStoreInit(IDvrStreamStore store)
```

---

### **notifyDvrStreamStoreDestroy**

```
public void notifyDvrStreamStoreDestroy(IDvrStreamStore store)
```

---

### **notifyDvrStreamStoreLoaded**

```
public void notifyDvrStreamStoreLoaded(IDvrStreamStore store)
```

---

### **notifyDvrStreamStorageDeleted**

```
public void notifyDvrStreamStorageDeleted(IDvrStreamStore store,
                                         IDvrFileSystem fileSystem,
                                         boolean success)
```

---

(continued from last page)

## notifyTimeReset

```
public void notifyTimeReset(IDvrStreamStore store,  
    long oldDvrTime,  
    long oldPacketTime,  
    TimeMapRecord newTimeMapRecord)
```

---

## getInitialRepeaterItems

```
public void getInitialRepeaterItems(java.util.List items)
```

Get initial repeater items to send to repeater receiver.

When an edge initially requests the items, instead of sending the entire manifest and all repeater items up until that point, the player sends a DvrRepeaterHolder.REPEATER\_INITIAL\_MANIFEST event followed by several DvrRepeaterHolder.REPEATER\_ADD\_MANIFEST\_ENTRIES events.

This allows us to not bother sending any purged manifest entries.

**Parameters:**

items

---

## getRepeaterItemsDvr

```
public void getRepeaterItemsDvr(java.util.List items,  
    long lastSeq)
```

---

## initialManifest

```
public void initialManifest(String vStreamName,  
    IDvrManifest manifest)
```

---

## initialManifestEnd

```
public void initialManifestEnd(String vStreamName)
```

---

## initialAttributes

```
public void initialAttributes(String vStreamName,  
    int dvrWindow)
```

---

## stateChange

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

(continued from last page)

## **removeStreamStore**

```
public void removeStreamStore(String vStreamName)
```

---

## **addRepeaterHeartBeatItem**

```
public void addRepeaterHeartBeatItem()
```

## com.wowza.wms.dvr Interface IDvrStreamStore

public interface **IDvrStreamStore**  
extends

### Method Summary

void	<a href="#"><code>addDvrChunkListener</code></a> ( <a href="#">IDvrStoreChunkListener</a> listener) Add (very fine) listener for chunk events.
void	<a href="#"><code>addManifestEntries</code></a> (java.util.List entries)
boolean	<a href="#"><code>canPlay</code></a> () Is this store capable of playing.
boolean	<a href="#"><code>canRecord</code></a> () Is this store capable of recording.
<a href="#">IApplicationInstance</a>	<a href="#"><code>getAppInstance</code></a> () Get associated application instance.
<a href="#">IDvrChunker</a>	<a href="#"><code>getChunker</code></a> ()
long	<a href="#"><code>getClosestStartTime</code></a> (int type, long t)
String	<a href="#"><code>getContextStr</code></a> () Get stream context string, useful for logging.
DvrChunk	<a href="#"><code>getDvrChunkAtTime</code></a> (int fragmentType, long t)
DvrChunk	<a href="#"><code>getDvrChunkByIndex</code></a> (int fragmentType, int index)
DvrChunk	<a href="#"><code>getDvrChunkNearTime</code></a> (int fragmentType, long t, long delta)
<a href="#">IDvrStreamManager</a>	<a href="#"><code>getDvrManager</code></a> () Get DVR Stream Manager
int	<a href="#"><code>getDvrStorageWindowSeconds</code></a> () Get DVR window size.
<a href="#">IDvrFileSystem</a>	<a href="#"><code>getFileSystem</code></a> ()
<a href="#">IDvrManifest</a>	<a href="#"><code>getManifest</code></a> () Get manifest
<a href="#">WMSProperties</a>	<a href="#"><code>getProperties</code></a> () Get stream store properties.
<a href="#">IDvrPurgeController</a>	<a href="#"><code>getPurgeController</code></a> () Get purge controller for stream store.

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

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

## 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 DvrChunkStorageInfo storeChunks(int vDuration,  
                                      DvrPacketHolder vPackets,  
                                      int aDuration,  
                                      DvrPacketHolder aPackets,  
                                      int dDuration,  
                                      DvrPacketHolder dPackets)
```

---

## storeOnMetadata

```
public boolean storeOnMetadata(long pt,  
                               long utc,  
                               AMFPacket metaPacket)
```

---

## purgeEntries

```
public void purgeEntries(DvrManifestEntryRangeGroup rangeGroup)
```

Purge entries from store

**Parameters:**

rangeGroup - s set of ranges for purging

---

(continued from last page)

## addManifestEntries

```
public void addManifestEntries(java.util.List entries)
```

---

## addDvrChunkListener

```
public void addDvrChunkListener(IDvrStoreChunkListener listener)
```

Add (very fine) listener for chunk events.

**Parameters:**

listener - listener

---

## removeDvrChunkListener

```
public void removeDvrChunkListener(IDvrStoreChunkListener listener)
```

Remove (very fine) listener for chunk events.

**Parameters:**

listener - listener

---

## getRecordedEntriesWithLimit

```
public java.util.List getRecordedEntriesWithLimit(int fragmentType,  
    long t,  
    int limit)
```

---

## getRecordedEntryByIndex

```
public DvrManifestEntry getRecordedEntryByIndex(int fragmentType,  
    int index)
```

---

## getRecordedEntryByTime

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

---

## getDvrChunkByIndex

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

---

## getDvrChunkAtTime

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

---

**getDvrChunkNearTime**

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

---

**getClosestStartTime**

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

## com.wowza.wms.dvr

### Interface IDvrStreamVersionHandler

All Known Implementing Classes:

[DefaultDvrStreamVersionHandler](#)

public interface **IDvrStreamVersionHandler**

extends

Callback to manage (archival) versions of DVR stream stores

### Method Summary

<a href="#">IDvrStreamStore</a>	<b>determineExistingStoreForPlaying</b> ( <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>	<b>determineExistingStoreForRecording</b> ( <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	<b>getArchiveStrategy</b> ( <a href="#">IDvrStreamManager</a> dvrManager, String baseStreamName)
	Determine the archive strategy for a given set of streams.
boolean	<b>handleArchivedStream</b> ( <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	<b>shouldDeleteArchivedStream</b> ( <a href="#">IDvrStreamManager</a> dvrManager, <a href="#">IDvrStreamStore</a> store)
	Determine if a given stream store version should be deleted.
boolean	<b>shouldLoadArchivedStream</b> ( <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 gto 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 init's, 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	<code>init(<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.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	<a href="#"><code>init(IApplicationInstance appInstance, IMediaStream stream, IDvrFileSystem fileSystem, String artifactName)</code></a>
------	--

#### Methods inherited from interface com.wowza.io.ITextWriter

<code>close, exists, getBasePath, getMediaName, getPath, init, isAppend, isOpen, lastModified, length, open, setAppend, write, write, write</code>
--

### 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](#), [getRecordedEntriesInRange](#), [getRecordedEntriesWithLimit](#), [getRecordedEntryByIndex](#), [getRecordedEntryByTimeKey](#), [getRecordedEntryStartingBeforeTime](#), [getType](#), [isEmpty](#)

### Methods

#### **getTimeMap**

public java.util.List **getTimeMap**()

(continued from last page)

**getTimeMapEntries**

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

---

**dvrToUtc**

```
public long dvrToUtc(long dt)
```

---

**dvrToPt**

```
public long dvrToPt(long dt)
```

---

**ptToDvr**

```
public long ptToDvr(long pt)
```

---

**utcToDvr**

```
public long utcToDvr(long utc)
```

---

**toDvr**

```
public long toDvr(long t,  
    IDvrConstants.DvrTimeScale timeScale)
```

---

**containsTime**

```
public boolean containsTime(long startTime,  
    IDvrConstants.DvrTimeScale timeScale)
```

---

---

Package

**com.wowza.wms.http**

## com.wowza.wms.http Class HTTPProvider2Base

```
java.lang.Object
+-com.wowza.wms.http.HTTPProvider2Base
```

All Implemented Interfaces:

[IHTTPProvider2](#)

```
public abstract class HTTPProvider2Base
extends Object
implements IHTTPProvider2
```

HTTPProvider2Base: base class for implementing HTTP Providers.

### Simple HTTPProvider class

```
public class HTTPHelloWowza extends HTTPProvider2Base
{
    public void onHTTPRequest(IVHost vhost, IHTTPRequest req, IHTTPResponse resp)
    {
        if (!doHTTPAuthentication(vhost, req, resp))
            return;

        String retStr = "<head><title>Hello Wowza</title></head><body>Hello
Wowza</body>" ;
        try
        {
            OutputStream out = resp.getOutputStream();
            byte[] outBytes = retStr.getBytes();
            out.write(outBytes);
        }
        catch (Exception e)
        {

WMSLoggerFactory.getLogger(HTTPHelloWowza.class).error("HTTPHelloWowza ", e);
        }

    }
}
```

### Field Summary

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

## Methods inherited from class java.lang.Object

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

**Methods inherited from interface com.wowza.wms.http.IHTTPProvider2**

[canHandle](#), [getAuthenticationMethod](#), [getRequestFilters](#), [init](#), [setAuthenticationMethod](#), [setRequestFilters](#)

**Methods inherited from interface com.wowza.wms.http.IHTTPProvider**

[onBind](#), [onHTTPRequest](#), [onUnbind](#), [setProperties](#)

## Fields

### properties

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

---

### filters

```
protected java.util.List filters
```

---

### requestFilters

```
protected java.lang.String requestFilters
```

---

### authenticateHandler

```
protected com.wowza.wms.authentication.IAuthenticate authenticateHandler
```

---

### authenticateHTTPProviderHandler

```
protected com.wowza.wms.authentication.IAuthenticateHTTPProvider  
authenticateHTTPProviderHandler
```

---

### authenticationMethod

```
protected java.lang.String authenticationMethod
```

## Constructors

### HTTPProvider2Base

```
public HTTPProvider2Base()
```

(continued from last page)

## Methods

### init

```
public void init()
```

Initialize the HTTProvider

### canHandle

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

Return true if can handle the request

**Parameters:**

path - request path

**Returns:**

true, if can handle the request

### getPath

```
public String getPath(IHTTPRequest req,  
           boolean removeFilter)
```

Get the request path

**Parameters:**

req - request

removeFilter - set to true to remove filter part of URL

**Returns:**

path

### setRequestFilters

```
public void setRequestFilters(String requestFilters)
```

Set the request filter

**Parameters:**

requestFilters - request filter

### getRequestFilters

```
public String getRequestFilters()
```

Get the request filter

**Returns:**

request filter

### setProperties

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

Set properties

(continued from last page)

**Parameters:**

properties - properties

---

**onBind**

```
public void onBind(IVHost vhost,  
                  HostPort hostPort)
```

Called when bind is called on port

**Parameters:**

vhost - vhost  
hostPort - host port

---

**onUnbind**

```
public void onUnbind(IVHost vhost,  
                     HostPort hostPort)
```

Called when unbind is called on port

**Parameters:**

vhost - vhost  
hostPort - host port

---

**getAuthenticationMethod**

```
public String getAuthenticationMethod()
```

Get the authentication method: digest, basic, none...

**Returns:**

authentication method

---

**setAuthenticationMethod**

```
public void setAuthenticationMethod(String authenticationMethod)
```

Set authentication method: digest, basic, none...

**Parameters:**

authenticationMethod - authentication method

---

**doHTTPAuthentication**

```
public boolean doHTTPAuthentication(IVHost vhost,  
                                    IHTTPRequest req,  
                                    IHTTPResponse resp)
```

Handle authentication request

**Parameters:**

vhost - vhost  
req - request  
resp - response

**Returns:**

true, if request should be handled by the HTTPProvider subclass

## com.wowza.wms.http Interface IHTTPProvider

All Subinterfaces:

[IHTTPProvider2](#)

public interface **IHTTPProvider**

extends

IHTTPProvider: HTTP provider class for a given HostPort definition. Receive all HTTP requests that are not RTMPT requests. See IHTTPProvider2 for the extended version of this interface.

### Method Summary

void	<a href="#">onBind(IVHost vhost, HostPort hostPort)</a> Triggered after hostPort binds to socket
void	<a href="#">onHTTPRequest(IVHost vhost, IHttpRequest req, IHttpResponse resp)</a> Triggered for each HTTP request to the given hostPort that is not an RTMPT request.
void	<a href="#">onUnbind(IVHost vhost, HostPort hostPort)</a> Triggered after hostPort unbinds
void	<a href="#">setProperties(WMSProperties properties)</a> Called to provide properties set in configuration files

### Methods

#### onBind

```
public void onBind\(IVHost vhost,
HostPort hostPort\)
```

Triggered after hostPort binds to socket

**Parameters:**

vhost - parent vhost  
hostPort - host port definition

#### onHTTPRequest

```
public void onHTTPRequest\(IVHost vhost,
IHttpRequest req,
IHttpResponse resp\)
```

Triggered for each HTTP request to the given hostPort that is not an RTMPT request.

**Parameters:**

vhost - parent vhost  
req - HTML request object  
resp - HTML response object

## onUnbind

```
public void onUnbind(IVHost vhost,  
                 HostPort hostPort)
```

Triggered after hostPort unbinds

**Parameters:**

vhost - parent vhost  
hostPort - host port definition

---

## setProperties

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

Called to provide properties set in configuration files

## com.wowza.wms.http Interface IHTTPProvider2

All Superinterfaces:

[IHTTPProvider](#)

All Known Implementing Classes:

[HTTPProvider2Base](#)

---

public interface **IHTTPProvider2**

extends [IHTTPProvider](#)

IHTTPProvider2: Extension of IHTTPProvider that adds support for multiple HTTPProviders attached to a single HostPort along with authentication. Multiple HTTPProviders can be added to a HostPort definition. They are configured as follows:

```
<HTTPProvider>
  <BaseClass>com.wowza.wms.http.HTTPServerVersion</BaseClass>
  <RequestFilters>*serverversion</RequestFilters>
  <AuthenticationMethod>none</AuthenticationMethod>
</HTTPProvider>
```

Below is an sample implementation of the HTTPServerVersion provider:

```

import java.io.*;

import com.wowza.wms.server.*;
import com.wowza.wms.stream.*;
import com.wowza.wms.vhost.*;
import com.wowza.wms.logging.*;

public class HTTPServerVersion extends HTTPProvider2Base
{
    public void onHTTPRequest(IVHost vhost, IHTTPRequest req, IHTTPResponse resp)
    {
        if (!doHTTPAuthentication(vhost, req, resp))
            return;

        String version = MediaStreamBase.p+" "+ReleaseInfo.getVersion()+""
build"+ReleaseInfo.getBuildNumber();
        String retStr =
"<html><head><title>" +version+"</title></head><body>" +version+"</body></html> ";
        try
        {
            OutputStream out = resp.getOutputStream();
            byte[] outBytes = retStr.getBytes();
            out.write(outBytes);
        }
        catch (Exception e)
        {

WMSLoggerFactory.getLogger(HTTPServerVersion.class).error("HTMLServerVersion ", e);
        }

    }
}

```

## Method Summary

boolean	<a href="#">canHandle(String path)</a>
String	<a href="#">getAuthenticationMethod()</a>
String	<a href="#">getRequestFilters()</a>
void	<a href="#">init()</a>
void	<a href="#">setAuthenticationMethod(String authenticationMethod)</a>
void	<a href="#">setRequestFilters(String requestFilters)</a>

---

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="#">getContentType()</a>
	Get the content type of the body of the message
String	<a href="#">getHeader(String name)</a>
	Get a HTTP header value such as 'Content-Length'
byte[]	<a href="#">getHeaderBytes()</a>
	Returns the header as bytes
java.util.Map	<a href="#">getHeaderMap()</a>
	Get a copy of the HTTP request header map
java.util.Set	<a href="#">getHeaderNames()</a>
	Get a Set of the header names
java.io.InputStream	<a href="#">getInputStream()</a>
	Get the body of the message as an input stream
int	<a href="#">getIntHeader(String name)</a>
	Get a HTTP header value such as 'Content-Length' and return as int
java.util.Locale	<a href="#">getLocale()</a>
	Get locale of request (Example: en-us)
String	<a href="#">getMethod()</a>
	Get the method invocation method: GET, POST, HEAD
byte[]	<a href="#">getMsgBytes()</a>
	Return the message bytes
String	<a href="#">getParameter(String name)</a>
	Get a parameter value
java.util.Map	<a href="#">getParameterMap()</a>
	Get the entire parameter Map
java.util.Set	<a href="#">getParameterNames()</a>
	Get a Set of parameter names
String[]	<a href="#">getParameterValues(String name)</a>
	Get a multi-value parameter as an array of String
String	<a href="#">getPath()</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 Streaming Engine")
int	<a href="#"><u>getServerPort()</u></a> Get the port this request was received on
boolean	<a href="#"><u>isHeadRequest()</u></a> Returns true if the request is a HEAD request
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'

(continued from last page)

**Parameters:**

name - header name

**Returns:**

header value

---

**getIntHeader**

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

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

**Parameters:**

name - header name

**Returns:**

header value

---

**getHeaderNames**

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

Get a Set of the header names

**Returns:**

Set of header names

---

**getMethod**

```
public String getMethod()
```

Get the method invocation method: GET, POST, HEAD

**Returns:**

method

---

**getQueryString**

```
public String getQueryString()
```

Get the query string part of the url (everything after the ?)

**Returns:**

query string

---

**getContentLength**

```
public int getContentLength()
```

Get the content length of the body of the message

**Returns:**

content length of the body of the message

---

**getRequestURI**

```
public String getRequestURI()
```

(continued from last page)

Get the full request URI

**Returns:**

full request URI

---

**getRequestURL**

```
public String getRequestURL()
```

Get the request url (same as URI minus the query string)

**Returns:**

request url

---

**getContentType**

```
public String getContentType()
```

Get the request content type

**Returns:**

request content type

---

**parseBodyForParams**

```
public void parseBodyForParams(boolean doDecode)
```

If the body of the message contains parameter data (data in name value pairs separated by & character) call this routine to decode those parameters and add them to the parameter map.

**Parameters:**

doDecode - true to decode the params as though they are url params

---

**parseBodyForParams**

```
public void parseBodyForParams()
```

If the body of the message contains parameter data (data in name value pairs separated by & character) call this routine to decode those parameters and add them to the parameter map.

---

**getParameter**

```
public String getParameter(String name)
```

Get a parameter value

**Parameters:**

name - parameter name

**Returns:**

parameter value

---

**getParameterNames**

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

Get a Set of parameter names

(continued from last page)

**Returns:**

Set of parameter names

---

**getParameterValues**

```
public String[] getParameterValues(String name)
```

Get a multi-value parameter as an array of String

**Parameters:**

name - parameter name

**Returns:**

multi-value parameter as an array of String

---

**getParameterMap**

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

Get the entire parameter Map

**Returns:**

parameter Map

---

**getInputStream**

```
public java.io.InputStream getInputStream()
```

Get the body of the message as an input stream

**Returns:**

body of the message as an input stream

---

**getProtocol**

```
public String getProtocol()
```

Get the request protocol (example: HTTP/1.1)

**Returns:**

request protocol

---

**getScheme**

```
public String getScheme()
```

Get the request scheme (Example "http")

**Returns:**

request scheme

---

**getServerName**

```
public String getServerName()
```

Get the name of the server (Example: "Wowza Streaming Engine")

**Returns:**

(continued from last page)

---

name of the server

---

**getServerPort**

```
public int getServerPort()
```

Get the port this request was received on

**Returns:**

the port this request was received on

---

**getRemoteAddr**

```
public String getRemoteAddr()
```

Get the remote ip address of the request

**Returns:**

remote ip address of the request

---

**getRemoteHost**

```
public String getRemoteHost()
```

Get the remote host name (if known) if not return ip address

**Returns:**

remote host name

---

**getLocale**

```
public java.util.Locale getLocale()
```

Get locale of request (Example: en-us)

**Returns:**

locale of request

---

**isSecure**

```
public boolean isSecure()
```

Returns true is the request is protected by SSL

**Returns:**

true is the request is protected by SSL

---

**getPath**

```
public String getPath()
```

Returns the HTTP path element of the request

**Returns:**

HTTP path element of the request

---

(continued from last page)

## getHeaderBytes

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

Returns the header as bytes

**Returns:**

header as bytes

---

## getMsgBytes

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

Return the message bytes

**Returns:**

message bytes

---

## isHeadRequest

```
public boolean isHeadRequest()
```

Returns true if the request is a HEAD request

**Returns:**

true if the request is a HEAD request

## com.wowza.wms.http Interface IHTTPResponse

public interface **IHTTPResponse**  
extends

### Method Summary

boolean	<a href="#"><u>containsHeader</u></a> (String name) Returns true if response header contains parameter name
String	<a href="#"><u>getHeader</u></a> (String name) Get header value
java.util.Map	<a href="#"><u>getHeaders</u></a> () Get the current response headers as a map
int	<a href="#"><u>getHeaderSize</u></a> () Get the size in bytes of the HTTP header
int	<a href="#"><u>getIntHeader</u></a> (String name) Get header value as int
java.io.OutputStream	<a href="#"><u>getOutputStream</u></a> () Get the output stream for the response.
boolean	<a href="#"><u>isHeadResponse</u></a> () returns true if the response is due to a HEAD request
void	<a href="#"><u>removeHeader</u></a> (String name) Remove header value
void	<a href="#"><u>setHeader</u></a> (String name, String value) Set header value
void	<a href="#"><u>setHeadResponse</u></a> (boolean isHeadResponse) indicates if the response is a response to a HEAD request
void	<a href="#"><u>setIntHeader</u></a> (String name, int value) Set header value as int
void	<a href="#"><u>setResponseCode</u></a> (int responseCode) Set the HTTP response code

### Methods

#### **getOutputStream**

public java.io.OutputStream [getOutputStream](#)()

Get the output stream for the response. You can then write directly into the output stream.

(continued from last page)

**Returns:**

output stream for the response

**containsHeader**

```
public boolean containsHeader(String name)
```

Returns true if response header contains parameter name

**Parameters:**

name - header parameter name

**Returns:**

true if header contains value

**setHeader**

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

Set header value

**Parameters:**

name - header parameter name

value - parameter value

**removeHeader**

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

Remove header value

**Parameters:**

name - name

**setIntHeader**

```
public void setIntHeader(String name,
                         int value)
```

Set header value as int

**Parameters:**

name - header parameter name

value - parameter value

**getHeaders**

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

Get the current response headers as a map

**Returns:**

current response headers as a map

**getHeader**

```
public String getHeader(String name)
```

(continued from last page)

Get header value

**Parameters:**

name - header parameter name

**Returns:**

parameter value

---

**getIntHeader**

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

Get header value as int

**Parameters:**

name - header parameter name

**Returns:**

parameter value

---

**setResponseCode**

```
public void setResponseCode(int responseCode)
```

Set the HTTP response code

**Parameters:**

responseCode - HTTP response code

---

**getHeaderSize**

```
public int getHeaderSize()
```

Get the size in bytes of the HTTP header

**Returns:**

header size in bytes

---

**setHeadResponse**

```
public void setHeadResponse(boolean isHeadResponse)
```

indicates if the response is a response to a HEAD request

**Parameters:**

isHeadResponse - indicating if this is a response to a HEAD request

---

**isHeadResponse**

```
public boolean isHeadResponse()
```

returns true if the response is due to a HEAD request

**Returns:**

true if the response is due to a HEAD request

---

**Package**

**com.wowza.wms.httpstreamer.cupertinostreaming.httpstreamer**

## com.wowza.wms.httpstreamer.cupertinostreaming.httpstreamer Class HTTPStreamerSessionCupertino

```
java.lang.Object
  +-- com.wowza.wms.httpstreamer.model.HTTPStreamerSessionBase
    +-
com.wowza.wms.httpstreamer.cupertinostreaming.httpstreamer.HTTPStreamerSessionCupertino
```

### All Implemented Interfaces:

[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, notifyCreate, playDuration, playSeek,
playStart, properties, queryStr, redirectSession, redirectSessionBody, redirectSessionCode,
redirectSessionContentType, redirectSessionURL, referrer, serverIp, serverPort, sessionId,
sessionProtocol, sessionTimeout, sessionType, stream, streamDomainStrSet, streamExt,
streamName, streamNamePartMap, streamPosition, timeoutSession, totalIOPerformance2Last,
totalIOPerformanceLast, transcoderVODIndex, transcoderVODIndexDestinationsMap,
transcoderVODIndexDestinationsOrder, transcoderVODSession, uri, userAgent, userHTTPHeaders,
userQueryStr, vhost, vodTranscodeNGRP
```

### Fields inherited from interface com.wowza.wms.httpstreamer.model.IHTTPStreamerSession

```
SESSIONPROTOCOL_COUNT, SESSIONPROTOCOL_CUPERTINOSTREAMING,
SESSIONPROTOCOL_DIRECTSTREAMING, SESSIONPROTOCOL_DVRCHUNKSTREAMING,
SESSIONPROTOCOL_MPEGDASHSTREAMING, SESSIONPROTOCOL_SANJOSESTREAMING,
SESSIONPROTOCOL_SMOOTHSTREAMING, SESSIONPROTOCOL_UNKNOWN, SESSIONPROTOCOL_WEBMSTREAMING,
SESSIONTYPE_LIVE, SESSIONTYPE_LIVEDVR, SESSIONTYPE_UNKNOWN, SESSIONTYPE_VOD
```

## Constructor Summary

public	<a href="#">HTTPStreamerSessionCupertino()</a>
--------	--

## Method Summary

void	<a href="#">clearLoggingValues()</a>
boolean	<a href="#">containsIndex(String streamName)</a>
IHTTPStreamerCupertinoIndex	<a href="#">getIndex(IHTTPStreamerApplicationContext appContext, IHTTPStreamerSession httpStreamerSession, String rawStreamName, String streamExt, String streamName, long playStart, long playDuration, long playSeek, TimedTextRequest captionRequest, String vodTranscodeNGRP)</a>

void	<a href="#">indexTranscoderVOD(IHTTPStreamerApplicationContext appContext, String vodTranscodeNGRP)</a>
void	<a href="#">logLiveChunk(LiveStreamPacketizerCupertinoChunk chunk)</a>
void	<a href="#">logVODChunk(LiveStreamPacketizerCupertinoChunk chunk)</a>
void	<a href="#">notifyHTTPSessionCreate(IAplicationInstance appInstance, IHTTPStreamerSession httpStreamerSession)</a>
void	<a href="#">notifyHTTPSessionDestroy(IAplicationInstance 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, doSessionRedirect, extractHTTPRequestInfo, getAndClearNotifyCreate, getAppInstance, getConnectionHolder, getCookieStr, getDvrSessionInfo, getElapsedTme, getFileInfo, getHTTPDate, getHTTPHeader, getHTTPHeaderMap, getHTTPHeaderNames, getHTTPIntHeader, getHTTPStreamerAdapter, getIOPerformanceCounter, getIpAddress, getLastRequest, getLiveStreamingPacketizer, getLock, getPlayDuration, getPlaySeek, getPlayStart, getProperties, getQueryStr, getRedirectSessionBody, getRedirectSessionCode, getRedirectSessionContentType, getRedirectSessionURL, getReferrer, getServerIp, getServerPort, getSessionId, getSessionProtocol, getSessionTimeout, getSessionType, getStream, getStreamDomainStr, getStreamDomainStrList, getStreamExt, getStreamName, getStreamNameParts, getStreamPosition, getTimeRunning, getTimeRunningSeconds, getTranscoderVODIndex, getTranscoderVODIndex, getTranscoderVODIndexDestinationsMap, getTranscoderVODIndexDestinationsOrder, getTranscoderVODSession, getUri, getUserAgent, getUserHTTPHeaders, getUserQueryStr, getVHost, getVODTranscodeNGRP, indexTranscoderVOD, isAcceptSession, isActive, isFileInfo, isHTTPOrigin, isPlayLogged, isRedirectSession, isTimeout, isTimeoutSession, isvalidated, isValidStreamDomainStr, lockRepeaterStreams, putFileInfo, putStreamNameParts, redirectSession, redirectSession, rejectSession, removeStreamDomainStr, setAcceptSession, setActive, setAppInstance, setCookieStr, setDvrSessionInfo, setHTTPOrigin, setHTTPStreamerAdapter, setIpAddress, setLiveStreamingPacketizer, setPlayDuration, setPlayLogged, setPlaySeek, setPlayStart, setQueryStr, setRedirectSession, setRedirectSessionBody, setRedirectSessionCode, setRedirectSessionContentType, setRedirectSessionURL, setReferrer, setServerIp, setServerPort, sessionId, setSessionProtocol, setSessionTimeout, setSessionType, setStream, setStreamExt, setStreamName, setStreamPosition, setThreadContext, setTimeOutSession, setTranscoderVODIndex, setTranscoderVODSession, setUri, setUserAgent, setUserHTTPHeader, setUserQueryStr, setValidated, setVHost, setVODTranscodeNGRP, shutdown, shutdownLocked, touch, updateLoggingValues, validStreamDomainToString

**Methods inherited from class java.lang.Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait

**Methods inherited from interface com.wowza.wms.httpstreamer.model.IHTTPStreamerSession**

```

acceptSession, addIOPerformance, addIOPerformance2, addStreamDomainStr,
addStreamDomainStrs, addUserHTTPHeaders, checkAndSetPlayLogged, clearLoggingValues,
containsStreamDomainStr, containsStreamNameParts, doSessionRedirect,
doSessionRedirect, extractHTTPRequestInfo, getAndClearNotifyCreate, getAppInstance,
getConnectionHolder, getCookieStr, getDvrSessionInfo, getElapsedTime, getFileInfo,
getHTTPHeader, getHTTPHeaderMap, getHTTPHeaderNames, getHTTPIntHeader,
getHTTPStreamerAdapter, getIOPerformanceCounter, getIpAddress, getLastRequest,
getLiveStreamingPacketizer, getLock, getPlayDuration, getPlaySeek, getPlayStart,
getProperties, getQueryStr, getRedirectSessionBody, getRedirectSessionCode,
getRedirectSessionContentType, getRedirectSessionURL, getReferrer, getServerIp,
getServerPort, getSessionId, getSessionProtocol, getSessionTimeout, getSessionType,
getStream, getStreamExt, getStreamName, getStreamNameParts, getStreamPosition,
getTimeRunning, getTimeRunningSeconds, getTranscoderVODIndex, getTranscoderVODIndex,
getTranscoderVODIndexDestinationsMap, getTranscoderVODIndexDestinationsOrder,
getTranscoderVODSession, getUri, getUserAgent, getUserHTTPHeaders, getUserQueryStr,
getVHost, getVODTranscodeNGRP, indexTranscoderVOD, isAcceptSession, isActive,
isFileInfo, isHTTPOrigin, isPlayLogged, isRedirectSession, isTimeout,
isTimeoutSession, isValidate, isValidStreamDomainStr, lockRepeaterStreams,
putFileInfo, putStreamNameParts, redirectSession, redirectSession, rejectSession,
removeStreamDomainStr, setAcceptSession, setActive, setAppInstance, setCookieStr,
setDvrSessionInfo, setHTTPOrigin, setHTTPStreamerAdapter, setIpAddress,
setLiveStreamingPacketizer, setPlayDuration, setPlayLogged, setPlaySeek,
setPlayStart, setQueryStr, setRedirectSession, setRedirectSessionBody,
setRedirectSessionCode, setRedirectSessionContentType, setRedirectSessionURL,
setReferrer, setServerIp, setServerPort, sessionId, sessionProtocol,
setSessionTimeout, setSessionType, setStream, setStreamExt, setStreamName,
setStreamPosition, setTimeoutSession, setTranscoderVODIndex, setTranscoderVODSession,
setUri, setUserAgent, setUserHTTPHeader, setUserQueryStr, setVHost,
setVODTranscodeNGRP, shutdown, touch, updateLoggingValues, validStreamDomainToString

```

**Methods inherited from interface com.wowza.wms.httpstreamer.model.IHTTPSessionNotify**

```
notifyHTTPSessionCreate, notifyHTTPSessionDestroy
```

## Constructors

### **HTTPStreamerSessionCupertino**

```
public HTTPStreamerSessionCupertino()
```

## Methods

### **shutdown**

```
public void shutdown()
```

### **containsIndex**

```
public boolean containsIndex(String streamName)
```

(continued from last page)

---

## getIndex

```
public IHTTPStreamerCupertinoIndex getIndex(IHTTPStreamerApplicationContext  
appContext,  
    IHTTPStreamerSession httpStreamerSession,  
    String rawStreamName,  
    String streamExt,  
    String streamName,  
    long playStart,  
    long playDuration,  
    long playSeek,  
    TimedTextRequest captionRequest,  
    String vodTranscodeNGRP)
```

---

## updateLoggingValues

```
public void updateLoggingValues()
```

---

## clearLoggingValues

```
public void clearLoggingValues()
```

---

## logLiveChunk

```
public void logLiveChunk(LiveStreamPacketizerCupertinoChunk chunk)
```

---

## logVODChunk

```
public void logVODChunk(LiveStreamPacketizerCupertinoChunk chunk)
```

---

## notifyHTTPSessionCreate

```
public void notifyHTTPSessionCreate(IApplicationInstance appInstance,  
    IHTTPStreamerSession httpStreamerSession)
```

---

## notifyHTTPSessionDestroy

```
public void notifyHTTPSessionDestroy(IApplicationInstance appInstance,  
    IHTTPStreamerSession httpStreamerSession)
```

---

(continued from last page)

## indexTranscoderVOD

```
public void indexTranscoderVOD(IHTTPStreamerApplicationContext appContext,  
String vodTranscodeNGRP)
```

## com.wowza.wms.httpstreamer.cupertinostreaming.httpstreamer Interface IHTTPStreamerCupertinoVODActionNotify

All Subinterfaces:

[IHTTPStreamerCupertinoVODActionNotify2](#), [IHTTPStreamerCupertinoVODActionNotify3](#)

---

public interface **IHTTPStreamerCupertinoVODActionNotify**  
extends

IHTTPStreamerCupertinoVODActionNotify: listener interface for video on demand iOS streaming. See  
HTTPStreamerApplicationContextCupertinoStreamer.addVODActionListener(IHTTPStreamerCupertinoVODActionNotify listener)

### Method Summary

void	<code>onCreate(IHTTPStreamerCupertinoIndex fileIndex, IHTTPStreamerApplicationContext appContext, IHTTPStreamerSession httpStreamerSession, String rawStreamName, String streamExt, String streamName)</code> Called when file index created
void	<code>onDestroy(IHTTPStreamerCupertinoIndex fileIndex)</code> Called after file index is destroyed
void	<code>onFillChunkEnd(IHTTPStreamerCupertinoIndex fileIndex, IHTTPStreamerCupertinoIndexItem item, LiveStreamPacketizerCupertinoChunk chunk, boolean audioOnly)</code> Called after each chunk is filled.
void	<code>onFillChunkStart(IHTTPStreamerCupertinoIndex fileIndex, IHTTPStreamerCupertinoIndexItem item, LiveStreamPacketizerCupertinoChunk chunk, boolean audioOnly)</code> Called each time a chunk is filled.
void	<code>onIndex(IHTTPStreamerCupertinoIndex fileIndex, IHTTPStreamerApplicationContext appContext, IHTTPStreamerSession httpStreamerSession, String rawStreamName, String streamExt, String streamName)</code> Called after file is indexed
void	<code>onInit(IHTTPStreamerCupertinoIndex fileIndex, IHTTPStreamerApplicationContext appContext, IHTTPStreamerSession httpStreamerSession, String rawStreamName, String streamExt, String streamName)</code> Called after initialized
void	<code>onOpen(IHTTPStreamerCupertinoIndex fileIndex, IHTTPStreamerApplicationContext appContext, IHTTPStreamerSession httpStreamerSession, String rawStreamName, String streamExt, String streamName)</code> Called after open

### Methods

---

(continued from last page)

## onCreate

```
public void onCreate(IHTTPStreamerCupertinoIndex fileIndex,  
    IHTTPStreamerApplicationContext appContext,  
    IHTTPStreamerSession httpStreamerSession,  
    String rawStreamName,  
    String streamExt,  
    String streamName)
```

Called when file index created

### Parameters:

- fileIndex - file index
- appContext - HTTP application context
- httpStreamerSession - HTTP streaming session
- rawStreamName - stream name
- streamExt - stream extension
- streamName - adjusted stream name

---

## onInit

```
public void onInit(IHTTPStreamerCupertinoIndex fileIndex,  
    IHTTPStreamerApplicationContext appContext,  
    IHTTPStreamerSession httpStreamerSession,  
    String rawStreamName,  
    String streamExt,  
    String streamName)
```

Called after initialized

### Parameters:

- fileIndex - file index
- appContext - HTTP application context
- httpStreamerSession - HTTP streaming session
- rawStreamName - stream name
- streamExt - stream extension
- streamName - adjusted stream name

---

## onOpen

```
public void onOpen(IHTTPStreamerCupertinoIndex fileIndex,  
    IHTTPStreamerApplicationContext appContext,  
    IHTTPStreamerSession httpStreamerSession,  
    String rawStreamName,  
    String streamExt,  
    String streamName)
```

Called after open

### Parameters:

- fileIndex - file index
- appContext - HTTP application context
- httpStreamerSession - HTTP streaming session
- rawStreamName - stream name
- streamExt - stream extension
- streamName - adjusted stream name

---

(continued from last page)

## onIndex

```
public void onIndex(IHTTPStreamerCupertinoIndex fileIndex,
    IHTTPStreamerApplicationContext appContext,
    IHTTPStreamerSession httpStreamerSession,
    String rawStreamName,
    String streamExt,
    String streamName)
```

Called after file is indexed

**Parameters:**

- fileIndex - file index
- appContext - HTTP application context
- httpStreamerSession - HTTP streaming session
- rawStreamName - stream name
- streamExt - stream extension
- streamName - adjusted stream name

## onFillChunkStart

```
public void onFillChunkStart(IHTTPStreamerCupertinoIndex fileIndex,
    IHTTPStreamerCupertinoIndexItem item,
    LiveStreamPacketizerCupertinoChunk chunk,
    boolean audioOnly)
```

Called each time a chunk is filled. Can be used to add ID3 data to the header of a chunk.

**Parameters:**

- fileIndex - file index
- item - index item
- chunk - chunk being filled
- audioOnly - is audio-only chunk

## onFillChunkEnd

```
public void onFillChunkEnd(IHTTPStreamerCupertinoIndex fileIndex,
    IHTTPStreamerCupertinoIndexItem item,
    LiveStreamPacketizerCupertinoChunk chunk,
    boolean audioOnly)
```

Called after each chunk is filled. Can be used to add ID3 data to the end of a chunk.

**Parameters:**

- fileIndex - file index
- item - index item
- chunk - chunk being filled
- audioOnly - is audio-only chunk

## onDestroy

```
public void onDestroy(IHTTPStreamerCupertinoIndex fileIndex)
```

Called after file index is destroyed

**Parameters:**

- fileIndex - file index

## com.wowza.wms.httpstreamer.cupertinostreaming.httpstreamer Interface IHTTPStreamerCupertinoVODActionNotify2

All Superinterfaces:

[IHTTPStreamerCupertinoVODActionNotify](#)

All Subinterfaces:

[IHTTPStreamerCupertinoVODActionNotify3](#)

---

public interface **IHTTPStreamerCupertinoVODActionNotify2**

extends [IHTTPStreamerCupertinoVODActionNotify](#)

### Method Summary

void	<a href="#"><b>onFillChunkDataPacket</b></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

## com.wowza.wms.httpstreamer.cupertinostreaming.httpstreamer Interface IHTTPStreamerCupertinoVODActionNotify3

All Superinterfaces:

[IHTTPStreamerCupertinoVODActionNotify2](#), [IHTTPStreamerCupertinoVODActionNotify](#)

public interface **IHTTPStreamerCupertinoVODActionNotify3**

extends [IHTTPStreamerCupertinoVODActionNotify2](#)

### Method Summary

<pre>void <a href="#">onFillChunkDataPacket</a>(IHTTPStreamerCupertinoIndex fileIndex,     IHTTPStreamerCupertinoIndexItem item,     LiveStreamPacketizerCupertinoChunk chunk, CupertinoStreamingRendition     rendition, <a href="#">AMFPacket</a> packet, <a href="#">ID3Frames</a> id3Frames)</pre>	<p>Called when data packet is encountered.</p>
<pre>void <a href="#">onFillChunkEnd</a>(IHTTPStreamerCupertinoIndex fileIndex,     IHTTPStreamerCupertinoIndexItem item,     LiveStreamPacketizerCupertinoChunk chunk, CupertinoStreamingRendition     rendition)</pre>	<p>Called after each chunk is filled.</p>
<pre>void <a href="#">onFillChunkStart</a>(IHTTPStreamerCupertinoIndex fileIndex,     IHTTPStreamerCupertinoIndexItem item,     LiveStreamPacketizerCupertinoChunk chunk, CupertinoStreamingRendition     rendition)</pre>	<p>Called each time a chunk is filled.</p>

Methods inherited from interface

[com.wowza.wms.httpstreamer.cupertinostreaming.httpstreamer.IHTTPStreamerCupertinoVODActionNotify2](#)

[onFillChunkDataPacket](#)

Methods inherited from interface

[com.wowza.wms.httpstreamer.cupertinostreaming.httpstreamer.IHTTPStreamerCupertinoVODActionNotify](#)

[onCreate](#), [onDestroy](#), [onFillChunkEnd](#), [onFillChunkStart](#), [onIndex](#), [onInit](#), [onOpen](#)

### Methods

#### **onFillChunkStart**

```
public void onFillChunkStart(IHTTPStreamerCupertinoIndex fileIndex,
    IHTTPStreamerCupertinoIndexItem item,
    LiveStreamPacketizerCupertinoChunk chunk,
    CupertinoStreamingRendition rendition)
```

Called each time a chunk is filled. Can be used to add ID3 data to the header of a chunk.

**Parameters:**

fileIndex - file index  
item - index item

---

(continued from last page)

chunk - chunk being filled

rendition - rendition (audioOnly, videoOnly...)

---

## onFillChunkEnd

```
public void onFillChunkEnd(IHTTPStreamerCupertinoIndex fileIndex,  
    IHTTPStreamerCupertinoIndexItem item,  
    LiveStreamPacketizerCupertinoChunk chunk,  
    CupertinoStreamingRendition rendition)
```

Called after each chunk is filled. Can be used to add ID3 data to the end of a chunk.

**Parameters:**

fileIndex - file index

item - index item

chunk - chunk being filled

rendition - rendition (audioOnly, videoOnly...)

---

## onFillChunkDataPacket

```
public void onFillChunkDataPacket(IHTTPStreamerCupertinoIndex fileIndex,  
    IHTTPStreamerCupertinoIndexItem item,  
    LiveStreamPacketizerCupertinoChunk chunk,  
    CupertinoStreamingRendition rendition,  
    AMF Packet packet,  
    ID3Frames id3Frames)
```

Called when data packet is encountered. Used to convert AMF events into ID3 tags for iOS streaming

**Parameters:**

fileIndex - file index

item - index item

chunk - chunk being filled

rendition - rendition (audioOnly, videoOnly...)

packet - amf packet

id3Frames - ID3 frames

---

Package

**com.wowza.wms.httpstreamer.directstreaming.httpstreamer**

## com.wowza.wms.httpstreamer.directstreaming.httpstreamer Class HTTPStreamerSessionDirect

```
java.lang.Object
  +-- com.wowza.wms.httpstreamer.model.HTTPStreamerSessionBase
    +-
com.wowza.wms.httpstreamer.directstreaming.httpstreamer.HTTPStreamerSessionDirect
```

### All Implemented Interfaces:

[IHTTPStreamerSession](#)

---

```
public class HTTPStreamerSessionDirect
extends HTTPStreamerSessionBase
```

#### Fields inherited from class com.wowza.wms.httpstreamer.model.HTTPStreamerSessionBase

```
acceptSession, appInstance, connectionHolder, cookieStr, DATEFORMAT, elapsedTime,
fastDateFormat, fileInfoMap, firstCheck, httpHeaders, httpStreamerAdapter,
ioPerformanceCounter, ipAddress, isActive, isHTTPOrigin, isPlayLogged, lastRequest,
liveStreamingPacketizer, lock, mediaCasterStreamLock, notifyCreate, playDuration, playSeek,
playStart, properties, queryStr, redirectSession, redirectSessionBody, redirectSessionCode,
redirectSessionContentType, redirectSessionURL, referrer, serverIp, serverPort, sessionId,
sessionProtocol, sessionTimeout, sessionType, stream, streamDomainStrSet, streamExt,
streamName, streamNamePartMap, streamPosition, timeoutSession, totalIOPerformance2Last,
totalIOPerformanceLast, transcoderVODIndex, transcoderVODIndexDestinationsMap,
transcoderVODIndexDestinationsOrder, transcoderVODSession, uri, userAgent, userHTTPHeaders,
userQueryStr, vhost, vodTranscodeNGRP
```

#### Fields inherited from interface com.wowza.wms.httpstreamer.model.IHTTPStreamerSession

```
SESSIONPROTOCOL_COUNT, SESSIONPROTOCOL_CUPERTINOSTREAMING,
SESSIONPROTOCOL_DIRECTSTREAMING, SESSIONPROTOCOL_DVRCHUNKSTREAMING,
SESSIONPROTOCOL_MPEGDASHSTREAMING, SESSIONPROTOCOL_SANJOSESTREAMING,
SESSIONPROTOCOL_SMOOTHSTREAMING, SESSIONPROTOCOL_UNKNOWN, SESSIONPROTOCOL_WEBMSTREAMING,
SESSIONTYPE_LIVE, SESSIONTYPE_LIVEDVR, SESSIONTYPE_UNKNOWN, SESSIONTYPE_VOD
```

## Constructor Summary

public	<a href="#">HTTPStreamerSessionDirect()</a>
--------	---

#### Methods inherited from class com.wowza.wms.httpstreamer.model.HTTPStreamerSessionBase

```

acceptSession, addIOPerformance, addIOPerformance2, addStreamDomainStr,
addStreamDomainStrs, addUserHTTPHeaders, checkAndSetPlayLogged, clearLoggingValues,
containsStreamDomainStr, containsStreamNameParts, doSessionRedirect,
doSessionRedirect, extractHTTPRequestInfo, getAndClearNotifyCreate, getAppInstance,
getConnectionHolder, getCookieStr, getDvrSessionInfo, getElapsedTime, getFileInfo,
getHTTPDate, getHTTPHeader, getHTTPHeaderMap, getHTTPHeaderNames, getHTTPIntHeader,
getHTTPStreamerAdapter, getIOPerformanceCounter, getIpAddress, getLastRequest,
getLiveStreamingPacketizer, getLock, getPlayDuration, getPlaySeek, getPlayStart,
getProperties, getQueryStr, getRedirectSessionBody, getRedirectSessionCode,
getRedirectSessionContentType, getRedirectSessionURL, getReferrer, getServerIp,
getServerPort, getSessionId, getSessionProtocol, getSessionTimeout, getSessionType,
getStream, getStreamDomainStr, getStreamDomainStrList, getStreamExt, getStreamName,
getStreamNameParts, getStreamPosition, getTimeRunning, getTimeRunningSeconds,
getTranscoderVODIndex, getTranscoderVODIndex, getTranscoderVODIndexDestinationsMap,
getTranscoderVODIndexDestinationsOrder, getTranscoderVODSession, getUri,
getUserAgent, getUserHTTPHeaders, getUserQueryStr, getVHost, getVODTranscodeNGRP,
indexTranscoderVOD, isAcceptSession, isActive, isFileInfo, isHTTPOrigin,
isPlayLogged, isRedirectSession, isTimeout, isTimeoutSession, isvalidated,
isValidStreamDomainStr, lockRepeaterStreams, putFileInfo, putStreamNameParts,
redirectSession, redirectSession, rejectSession, removeStreamDomainStr,
setAcceptSession, setActive, setAppInstance, setCookieStr, setDvrSessionInfo,
setHTTPOrigin, setHTTPStreamerAdapter, setIpAddress, setLiveStreamingPacketizer,
setPlayDuration, setPlayLogged, setPlaySeek, setPlayStart, setQueryStr,
setRedirectSession, setRedirectSessionBody, setRedirectSessionCode,
setRedirectSessionContentType, setRedirectSessionURL, setReferrer, setServerIp,
setServerPort, sessionId, sessionProtocol, sessionTimeout, sessionType,
setStream, setStreamExt, setStreamName, setStreamPosition, setThreadContext,
setTimeoutSession, setTranscoderVODIndex, setTranscoderVODSession, setUri,
setUserAgent, setUserHTTPHeader, setUserQueryStr, setValidated, setVHost,
setVODTranscodeNGRP, shutdown, shutdownLocked, touch, updateLoggingValues,
validStreamDomainToString

```

**Methods inherited from class java.lang.Object**

```

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait,
wait

```

**Methods inherited from interface [com.wowza.wms.httpstreamer.model.IHTTPStreamerSession](#)**

```

acceptSession, addIOPerformance, addIOPerformance2, addStreamDomainStr,
addStreamDomainStrs, addUserHTTPHeaders, checkAndSetPlayLogged, clearLoggingValues,
containsStreamDomainStr, containsStreamNameParts, doSessionRedirect,
doSessionRedirect, extractHTTPRequestInfo, getAndClearNotifyCreate, getAppInstance,
getConnectionHolder, getCookieStr, getDvrSessionInfo, getElapsedTime, getFileInfo,
getHTTPHeader, getHTTPHeaderMap, getHTTPHeaderNames, getHTTPIntHeader,
getHTTPStreamerAdapter, getIOPerformanceCounter, getIpAddress, getLastRequest,
getLiveStreamingPacketizer, getLock, getPlayDuration, getPlaySeek, getPlayStart,
getProperties, getQueryStr, getRedirectSessionBody, getRedirectSessionCode,
getRedirectSessionContentType, getRedirectSessionURL, getReferrer, getServerIp,
getServerPort, getSessionId, getSessionProtocol, getSessionTimeout, getSessionType,
getStream, getStreamExt, getStreamName, getStreamNameParts, getStreamPosition,
getTimeRunning, getTimeRunningSeconds, getTranscoderVODIndex, getTranscoderVODIndex,
getTranscoderVODIndexDestinationsMap, getTranscoderVODIndexDestinationsOrder,
getTranscoderVODSession, getUri, getUserAgent, getUserHTTPHeaders, getUserQueryStr,
getVHost, getVODTranscodeNGRP, indexTranscoderVOD, isAcceptSession, isActive,
isFileInfo, isHTTPOrigin, isPlayLogged, isRedirectSession, isTimeout,
isTimeoutSession, isvalidated, isValidStreamDomainStr, lockRepeaterStreams,
putFileInfo, putStreamNameParts, redirectSession, redirectSession, rejectSession,
removeStreamDomainStr, setAcceptSession, setActive, setAppInstance, setCookieStr,
setDvrSessionInfo, setHTTPOrigin, setHTTPStreamerAdapter, setIpAddress,
setLiveStreamingPacketizer, setPlayDuration, setPlayLogged, setPlaySeek,
setPlayStart, setQueryStr, setRedirectSession, setRedirectSessionBody,
setRedirectSessionCode, setRedirectSessionContentType, setRedirectSessionURL,
setReferrer, setServerIp, setServerPort, sessionId, sessionProtocol,
getSessionTimeout, getSessionType, setStream, setStreamExt, setStreamName,
setStreamPosition, setTimeoutSession, setTranscoderVODIndex, setTranscoderVODSession,
setUri, setUserAgent, setUserHTTPHeader, setUserQueryStr, setVHost,
setVODTranscodeNGRP, shutdown, touch, updateLoggingValues, validStreamDomainToString

```

## Constructors

### HTTPStreamerSessionDirect

```
public HTTPStreamerSessionDirect()
```

---

**Package**

**com.wowza.wms.httpstreamer.model**

## com.wowza.wms.httpstreamer.model Interface IHTTPCryptoHolder

public interface IHTTPCryptoHolder  
extends

### Method Summary

byte[]	<a href="#">getBuffer()</a>
int	<a href="#">getOffset()</a>
int	<a href="#">getPacketType()</a>
long	<a href="#">getSample()</a>
int	<a href="#">getSize()</a>
void	<a href="#">setBuffer(byte[] buffer)</a>
void	<a href="#">setOffset(int offset)</a>
void	<a href="#">setPacketType(int packetType)</a>
void	<a href="#">setSample(long sample)</a>
void	<a href="#">setSize(int size)</a>

### Methods

#### **getBuffer**

public byte[] **getBuffer()**

#### **setBuffer**

public void **setBuffer(byte[] buffer)**

#### **getOffset**

public int **getOffset()**

---

### **setOffset**

```
public void setOffset(int offset)
```

---

### **getSize**

```
public int getSize()
```

---

### **setSize**

```
public void setSize(int size)
```

---

### **getPacketType**

```
public int getPacketType()
```

---

### **setPacketType**

```
public void setPacketType(int packetType)
```

---

### **getSample**

```
public long getSample()
```

---

### **setSample**

```
public void setSample(long sample)
```

---

## com.wowza.wms.httpstreamer.model Interface IHTTPSessionFactory

---

public interface IHTTPSessionFactory  
extends

### Method Summary

<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](#), [HTTPStreamerSessionMPEGDash](#),  
[HTTPStreamerSessionCupertino](#)

---

public interface **IHTTPSessionNotify**

extends

### Method Summary

void	<a href="#">notifyHTTPSessionCreate(<i>IAplicationInstance</i> appInstance, <i>IHTTPStreamerSession</i> httpStreamerSession)</a>
void	<a href="#">notifyHTTPSessionDestroy(<i>IAplicationInstance</i> appInstance, <i>IHTTPStreamerSession</i> httpStreamerSession)</a>

### Methods

#### **notifyHTTPSessionCreate**

public void **notifyHTTPSessionCreate**([IAplicationInstance](#) appInstance,  
[IHTTPStreamerSession](#) httpStreamerSession)

---

#### **notifyHTTPSessionDestroy**

public void **notifyHTTPSessionDestroy**([IAplicationInstance](#) appInstance,  
[IHTTPStreamerSession](#) httpStreamerSession)

## com.wowza.wms.httpstreamer.model Interface IHTTPStreamerAdapter

public interface **IHTTPStreamerAdapter**  
extends

IHTTPStreamerAdapter: HTTP streaming adapter interface

### Method Summary

boolean	<a href="#">canHandle(String path)</a> Return true if can handle request
String	<a href="#">getAdapterName()</a> Get the name of this adapter
HTTPStreamerItem	<a href="#">getHTTPStreamerItem()</a> Get the HTTP streamer item associated with this adapter
String	<a href="#">getID()</a> Get the id of this adapter
int	<a href="#">getIdleFrequency()</a> Get the idle frequency (milliseconds) for HTTP requests.
<a href="#">WMSProperties</a>	<a href="#">getProperties()</a> Get properties
int	<a href="#">getSessionProtocol()</a> Set the adapter session protocol.
<a href="#">IVHost</a>	<a href="#">getVHost()</a> Get the vhost associated with this adapter
void	<a href="#">init()</a> Initialize the HTTP streaming adapter
void	<a href="#">service(org.apache.mina.common.IoSession session, RtmpRequestMessage req, RtmpResponseMessage resp)</a> Called to service each request
void	<a href="#">setHTTPStreamerItem(HTTPEmbedderItem httpStreamerItem)</a> Set the HTTP streamer item associated with this adapter
void	<a href="#">setID(String id)</a> Set the id of this adapter
void	<a href="#">setProperties(<a href="#">WMSProperties</a> properties)</a> Set properties
void	<a href="#">setSessionProtocol(int sessionProtocol)</a> Get the adapter session protocol.
void	<a href="#">setVHost(<a href="#">IVHost</a> vhost)</a> Set the vhost associated with this adapter

void	<u><a href="#">shutdownSession</a></u> ( <u><a href="#">IApplicationInstance</a></u> appInstance, <u><a href="#">IHTTPStreamerSession</a></u> 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.IoSession session,
    RtmpRequestMessage req,
    RtmpResponseMessage resp)
```

Called to service each request

**Parameters:**

session - io session  
req - request  
resp - response

### getProperties

```
public WMSProperties getProperties()
```

Get properties

**Returns:**

properties

### setProperties

```
public void setProperties(WMSProperties properties)
```

Set properties

**Parameters:**

properties - properties

### getHTTPStreamerItem

```
public HTTPStreamerItem getHTTPStreamerItem()
```

Get the HTTP streamer item associated with this adapter

**Returns:**

(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="#">HTTPORIGINMODE_AUTO</a>
	Value: <b>1</b>
public static final	<a href="#">HTTPORIGINMODE_OFF</a>
	Value: <b>0</b>
public static final	<a href="#">HTTPORIGINMODE_ON</a>
	Value: <b>2</b>

### Method Summary

void	<a href="#">addQueryStrRemoveList(String name)</a> Add a name to the list of query string parameters that will not be copied to the media URLs for HTTP streaming
java.util.Map	<a href="#">filterQueryStr(java.util.Map map)</a> Internal
<a href="#">IApplicationInstance</a>	<a href="#">getAppInstance()</a> Get application instance
String	<a href="#">getCacheControlStatus4xx()</a> Get the cache-control header for 4xx HTTP responses
int	<a href="#">getHTTPOriginMode()</a> Get current HTTP origin mode.
<a href="#">WMSProperties</a>	<a href="#">getProperties()</a> Get properties
<a href="#">MediaCasterItem</a>	<a href="#">getRepeaterMediaCasterDef()</a> Get the live repeater media caster definition
int	<a href="#">getResponseOnIncompleteChunk()</a> Get the response code used when an incomplete chunk (read error) is encountered.
String	<a href="#">getStreamTypeStr()</a> Get stream type
java.util.Map	<a href="#">getUserHTTPHeaders()</a> Get user HTTP headers.

<a href="#">IVHost</a>	<a href="#"><code>getVHost()</code></a> Get vhost
void	<a href="#"><code>init(IApplicationInstance appInstance, HTTPStreamerItem httpStreamerItem)</code></a> Initialize context
boolean	<a href="#"><code>isHTTPOriginOn()</code></a> Returns true if HTTP origin mode is set to ON.
boolean	<a href="#"><code>isStreamDomainProtectionActive()</code></a> Return true if stream domain protection is on
void	<a href="#"><code>setCacheControlStatus4xx(String cacheControlStatus4xx)</code></a> Set the cache-control header for 4xx HTTP responses
void	<a href="#"><code>setHTTPOriginMode(int httpOriginMode)</code></a> Set current HTTP origin mode.
void	<a href="#"><code>setRepeaterMediaCasterDef(MediaCasterItem repeaterMediaCasterDef)</code></a> Set the live repeater media caster definition
void	<a href="#"><code>setResponseOnIncompleteChunk(int responseOnIncompleteChunk)</code></a> Set the response code used when an incomplete chunk (read error) is encountered.
void	<a href="#"><code>setStreamDomainProtectionActive(boolean streamDomainProtectionActive)</code></a> Turn on/off stream domain protection
void	<a href="#"><code>setStreamTypeStr(String streamTypeStr)</code></a> Set stream type
void	<a href="#"><code>setUserHTTPHeader(String name, String value)</code></a> Add a user HTTP header to the header map.

## Fields

### HTTPORIGINMODE\_OFF

```
public static final int HTTPORIGINMODE_OFF
```

Constant value: 0

### HTTPORIGINMODE\_AUTO

```
public static final int HTTPORIGINMODE_AUTO
```

Constant value: 1

### HTTPORIGINMODE\_ON

```
public static final int HTTPORIGINMODE_ON
```

Constant value: 2

(continued from last page)

## Methods

### init

```
public void init(IApplicationInstance appInstance,  
    HTTPStreamerItem httpStreamerItem)
```

Initialize context

**Parameters:**

appInstance - application instance  
httpStreamerItem - HTTP streamer

---

### getProperties

```
public WMSProperties getProperties()
```

Get properties

**Returns:**

properties

---

### getAppInstance

```
public IApplicationInstance getAppInstance()
```

Get application instance

**Returns:**

application instance

---

### getVHost

```
public IVHost getVHost()
```

Get vhost

**Returns:**

vhost

---

### getRepeaterMediaCasterDef

```
public MediaCasterItem getRepeaterMediaCasterDef()
```

Get the live repeater media caster definition

**Returns:**

live repeater media caster definition

---

### setRepeaterMediaCasterDef

```
public void setRepeaterMediaCasterDef(MediaCasterItem repeaterMediaCasterDef)
```

Set the live repeater media caster definition

**Parameters:**

repeaterMediaCasterDef - live repeater media caster definition

---

## getStreamTypeStr

```
public String getStreamTypeStr()
```

Get stream type

**Returns:**

stream type

---

## setStreamTypeStr

```
public void setStreamTypeStr(String streamTypeStr)
```

Set stream type

**Parameters:**

streamTypeStr - stream type

---

## isStreamDomainProtectionActive

```
public boolean isStreamDomainProtectionActive()
```

Return true if stream domain protection is on

**Returns:**

true if stream domain protection is on

---

## setStreamDomainProtectionActive

```
public void setStreamDomainProtectionActive(boolean streamDomainProtectionActive)
```

Turn on/off stream domain protection

**Parameters:**

streamDomainProtectionActive - true if stream domain protection is on

---

## getHTTPOriginMode

```
public int getHTTPOriginMode()
```

Get current HTTP origin mode. See HTTPORIGINMODE\_ (auto is currently not supported)

**Returns:**

HTTP origin mode

---

## setHTTPOriginMode

```
public void setHTTPOriginMode(int httpOriginMode)
```

Set current HTTP origin mode. See HTTPORIGINMODE\_ (auto is currently not supported)

**Parameters:**

httpOriginMode - HTTP origin mode

---

## isHTTPOriginOn

```
public boolean isHTTPOriginOn()
```

(continued from last page)

Returns true if HTTP origin mode is set to ON.

**Returns:**

true if HTTP origin mode is set to ON.

---

**setUserHTTPHeader**

```
public void setUserHTTPHeader(String name,  
                           String value)
```

Add a user HTTP header to the header map. These headers will be added to every HTTP response.

**Parameters:**

name - name

value - value (null if no value)

---

**getUserHTTPHeaders**

```
public java.util.Map getUserHTTPHeaders()
```

Get user HTTP headers.

**Returns:**

header map

---

**addQueryStrRemoveList**

```
public void addQueryStrRemoveList(String name)
```

Add a name to the list of query string parameters that will not be copied to the media URLs for HTTP streaming

**Parameters:**

name - query name

---

**filterQueryStr**

```
public java.util.Map filterQueryStr(java.util.Map map)
```

Internal

**Parameters:**

map - query map

**Returns:**

query map

---

**getCacheControlStatus4xx**

```
public String getCacheControlStatus4xx()
```

Get the cache-control header for 4xx HTTP responses

**Returns:**

cache-control header for 4xx HTTP responses

---

**setCacheControlStatus4xx**

```
public void setCacheControlStatus4xx(String cacheControlStatus4xx)
```

---

(continued from last page)

Set the cache-control header for 4xx HTTP responses

**Parameters:**

cacheControlStatus4xx - cache-control header for 4xx HTTP responses

---

## getResponseOnIncompleteChunk

```
public int getResponseOnIncompleteChunk()
```

Get the response code used when an incomplete chunk (read error) is encountered. If zero, chunk is sent.

**Returns:**

response code

---

## setResponseOnIncompleteChunk

```
public void setResponseOnIncompleteChunk(int responseOnIncompleteChunk)
```

Set the response code used when an incomplete chunk (read error) is encountered. If zero, chunk is sent.

**Parameters:**

responseOnIncompleteChunk - response code

## 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="#"><u>SESSIONPROTOCOL_COUNT</u></a>
	Value: <b>7</b>
public static final	<a href="#"><u>SESSIONPROTOCOL_CUPERTINOSTREAMING</u></a>
	Value: <b>1</b>
public static final	<a href="#"><u>SESSIONPROTOCOL_DIRECTSTREAMING</u></a>
	Value: <b>6</b>
public static final	<a href="#"><u>SESSIONPROTOCOL_DVRCHUNKSTREAMING</u></a>
	Value: <b>5</b>
public static final	<a href="#"><u>SESSIONPROTOCOL_MPEGDASHSTREAMING</u></a>
	Value: <b>4</b>
public static final	<a href="#"><u>SESSIONPROTOCOL_SANJOSESTREAMING</u></a>
	Value: <b>2</b>
public static final	<a href="#"><u>SESSIONPROTOCOL_SMOOTHSTREAMING</u></a>
	Value: <b>0</b>
public static final	<a href="#"><u>SESSIONPROTOCOL_UNKNOWN</u></a>
	Value: <b>-1</b>
public static final	<a href="#"><u>SESSIONPROTOCOL_WEBMSTREAMING</u></a>
	Value: <b>3</b>
public static final	<a href="#"><u>SESSIONTYPE_LIVE</u></a>
	Value: <b>1</b>
public static final	<a href="#"><u>SESSIONTYPE_LIVEDVR</u></a>
	Value: <b>3</b>
public static final	<a href="#"><u>SESSIONTYPE_UNKNOWN</u></a>
	Value: <b>0</b>

<code>public static final</code>	<code>SESSIONTYPE_VOD</code>
----------------------------------	------------------------------

	Value: <b>2</b>
--	-----------------

## Method Summary

<code>void</code>	<a href="#"><code>acceptSession()</code></a> Accept the HTTP session.
<code>void</code>	<a href="#"><code>addIOPerformance(IOPerformanceCounter totalIOPerformanceResult)</code></a> Internal user, keep track of IO performance
<code>void</code>	<a href="#"><code>addIOPerformance2(IOPerformanceCounter totalIOPerformanceResult)</code></a> Internal user, keep track of IO performance
<code>void</code>	<a href="#"><code>addStreamDomainStr(String streamDomainStr)</code></a> Internal user, add stream name
<code>void</code>	<a href="#"><code>addStreamDomainStrs(java.util.List streamNames)</code></a> Internal user, add stream names
<code>void</code>	<a href="#"><code>addUserHTTPHeaders(IHTTPResponse resp)</code></a> Internal
<code>boolean</code>	<a href="#"><code>checkAndSetPlayLogged()</code></a> If play has not been logged return false, else returns true, sets play has been logged
<code>void</code>	<a href="#"><code>clearLoggingValues()</code></a> Internal user, clear logging values
<code>boolean</code>	<a href="#"><code>containsStreamDomainStr(String streamDomainStr)</code></a> Internal user, test stream name
<code>boolean</code>	<a href="#"><code>containsStreamNameParts(String streamName)</code></a> Return true if stream name in stream name parts
<code>void</code>	<a href="#"><code>doSessionRedirect(IHTTPResponse resp)</code></a> Internal
<code>void</code>	<a href="#"><code>doSessionRedirect(IHTTPResponse resp, String redirectSessionURL, String redirectSessionContentType, int redirectSessionCode)</code></a> Internal
<code>void</code>	<a href="#"><code>extractHTTPRequestInfo(IHTTPRequest req)</code></a> Extract information from HTTP request
<code>boolean</code>	<a href="#"><code>getAndClearNotifyCreate()</code></a>
<code>IApplicationInstance</code>	<a href="#"><code>getAppInstance()</code></a> Get the application instance associated with this HTTP session
<code>ConnectionHolder</code>	<a href="#"><code>getConnectionHolder()</code></a> Connection holder for this session
<code>String</code>	<a href="#"><code>getCookieStr()</code></a> Get cookie string
<code>DvrSessionInfo</code>	<a href="#"><code>getDvrSessionInfo()</code></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
<a href="#"><u>HTTPStreamerFileInfo</u></a>	<a href="#"><u>getFileInfo(String streamName)</u></a> Get the file information if video on demand streaming
<a href="#"><u>String</u></a>	<a href="#"><u>getHTTPHeader(String name)</u></a> Get a HTTP header value such as 'Content-Length'
<a href="#"><u>java.util.Map</u></a>	<a href="#"><u>getHTTPHeaderMap()</u></a> Get a copy of the HTTP request header map
<a href="#"><u>java.util.Set</u></a>	<a href="#"><u>getHTTPHeaderNames()</u></a> Get a Set of the header names
<a href="#"><u>int</u></a>	<a href="#"><u>getHTTPIntHeader(String name)</u></a> Get a HTTP header value such as 'Content-Length' and return as int
<a href="#"><u>IHTTPStreamerAdapter</u></a>	<a href="#"><u>getHTTPStreamerAdapter()</u></a> Get the HTTP streaming adapter associated with this HTTP session
<a href="#"><u>IOPerformanceCounter</u></a>	<a href="#"><u>getIOPerformanceCounter()</u></a> Get IO performance counter
<a href="#"><u>String</u></a>	<a href="#"><u>getIpAddress()</u></a> Get the IP address
<a href="#"><u>long</u></a>	<a href="#"><u>getLastRequest()</u></a> Get the last timestamp of the last Io request
<a href="#"><u>String</u></a>	<a href="#"><u>getLiveStreamingPacketizer()</u></a> Get the live stream packetizer name
<a href="#"><u>Object</u></a>	<a href="#"><u>getLock()</u></a> Get the synchronization lock for this HTTP session
<a href="#"><u>long</u></a>	<a href="#"><u>getPlayDuration()</u></a> Get the play duration (milliseconds) for video on demand playback.
<a href="#"><u>long</u></a>	<a href="#"><u>getPlaySeek()</u></a> Get the seek location (milliseconds) where playback should start
<a href="#"><u>long</u></a>	<a href="#"><u>getPlayStart()</u></a> Get the play start time offset (milliseconds) for video on demand playback.
<a href="#"><u>WMSProperties</u></a>	<a href="#"><u>getProperties()</u></a> Get the properties associated with this session
<a href="#"><u>String</u></a>	<a href="#"><u>getQueryStr()</u></a> Get query string
<a href="#"><u>byte[]</u></a>	<a href="#"><u>getRedirectSessionBody()</u></a> Get redirect session body
<a href="#"><u>int</u></a>	<a href="#"><u>getRedirectSessionCode()</u></a> Get session redirect HTTP response code (default 302)
<a href="#"><u>String</u></a>	<a href="#"><u>getRedirectSessionContentType()</u></a> Get redirect session HTTP Content-Type

String	<a href="#"><code>getRedirectSessionURL()</code></a> Get redirect session URL
String	<a href="#"><code>getReferrer()</code></a> Get referrer
String	<a href="#"><code>getServerIp()</code></a> Get server IP address
int	<a href="#"><code>getServerPort()</code></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="#"><u>IMediaStream</u></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)
<a href="#"><u>IHTTPStreamerTranscoderVODIndex</u></a>	<a href="#"><code>getTranscoderVODIndex()</code></a> Get the VOD Transcoder index information
<a href="#"><u>IHTTPStreamerTranscoderVODDestinationIndex</u></a>	<a href="#"><code>getTranscoderVODIndex(String bitrateStr)</code></a> Get VOD transcoder index for given bitrate
java.util.Map	<a href="#"><code>getTranscoderVODIndexDestinationsMap()</code></a> Get the VOD Transcoder destination map
java.util.List	<a href="#"><code>getTranscoderVODIndexDestinationsOrder()</code></a> Get the list of VOD Transcoder destinations in playlist order
<a href="#"><u>com.wowza.wms.transcoder.vod.TranscoderVODSession</u></a>	<a href="#"><code>getTranscoderVODSession()</code></a> Get the vod transcoder session associated with this HTTP session

String	<a href="#"><u>getUri()</u></a> Get the URI associated with initial request
String	<a href="#"><u>getUserAgent()</u></a> Get user agent
java.util.Map	<a href="#"><u>getUserHTTPHeaders()</u></a> Get user HTTP header.
String	<a href="#"><u>getUserQueryStr()</u></a> This query string will be added to URLs used in HTTP streaming
<a href="#"><u>IVHost</u></a>	<a href="#"><u>getVHost()</u></a> Get vhost
String	<a href="#"><u>getVODTranscodeNGRP()</u></a> Get the VOD Transcoder NGRP name
void	<a href="#"><u>indexTranscodervOD(IHTTPStreamerApplicationContext appContext, String vodTranscodeNGRP)</u></a> Index the HTTP session for VOD transcoder streaming
boolean	<a href="#"><u>isAcceptSession()</u></a> Return true if this session has not been rejected
boolean	<a href="#"><u>isActive()</u></a> Is this session active, false after shutdown
boolean	<a href="#"><u>isFileInfo(String streamName)</u></a> Return true if the is file information for a given stream name
boolean	<a href="#"><u>isHTTPPOrigin()</u></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(long timecode)</u></a> 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(String streamDomainStr)</u></a> Internal user, is stream name valid for HTTP session
void	<a href="#"><u>lockRepeaterStreams(java.util.List streamNames, String liveStreamPacketizer, String liveStreamRepeater, String streamTypeStr)</u></a> Internal user, lock in reapter streams
void	<a href="#"><u>putFileInfo(String streamName, HTTPStreamerFileInfo fileInfo)</u></a> 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>setPlaySeek</u></a> (long playSeek) Set the seek location (milliseconds) where playback should start
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> (IMediaStream stream) Set the IMediaStream associated with this HTTP session
void	<a href="#"><u>setStreamExt</u></a> (String streamExt) Set stream extension
void	<a href="#"><u>setStreamName</u></a> (String streamName) Set stream name
void	<a href="#"><u>setStreamPosition</u></a> (long streamPosition) Set stream position (will not cause seek)
void	<a href="#"><u>setTimeoutSession</u></a> (boolean timeoutSession) Set is session timeout.
void	<a href="#"><u>setTranscoderVODIndex</u></a> (IHTTPStreamerTranscoderVODIndex transcoderVODIndex) Set the VOD Transcoder index information
void	<a href="#"><u>setTranscoderVODSession</u></a> (com.wowza.wms.transcoder.vod.TranscoderVODSession transcoderVODSession) Set the vod transcoder session associated with this HTTP session
void	<a href="#"><u>setUri</u></a> (String uri) Set the URI associated with initial request

void	<a href="#"><u>setUserAgent</u></a> (String userAgent) Set user agent
void	<a href="#"><u>setUserHTTPHeader</u></a> (String name, String value) Set user HTTP header.
void	<a href="#"><u>setUserQueryStr</u></a> (String userQueryStr) This query string will be added to URLs used in HTTP streaming
void	<a href="#"><u>setVHost</u></a> (IVHost vhost) Set vhost
void	<a href="#"><u>setVODTranscodeNGRP</u></a> (String vodTranscodeNGRP) Set the VOD Transcoder NGRP name
void	<a href="#"><u>shutdown</u></a> () Called then the HTTP session is shutting down
void	<a href="#"><u>touch</u></a> (long timecode) Touch this session to keep it active.
void	<a href="#"><u>updateLoggingValues</u></a> () Internal user, update logging values
String	<a href="#"><u>validStreamDomainToString</u></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**

### 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
```

Constant value: **5**

---

**SESSIONPROTOCOL\_DIRECTSTREAMING**

```
public static final int SESSIONPROTOCOL_DIRECTSTREAMING
```

Constant value: **6**

---

(continued from last page)

## SESSIONPROTOCOL\_COUNT

```
public static final int SESSIONPROTOCOL_COUNT
```

Constant value: 7

### Methods

#### getSessionId

```
public String getSessionId()
```

Get session id

**Returns:**

session id

#### setSessionId

```
public void setSessionId(String sessionId)
```

Set session id

**Parameters:**

sessionId - session id

#### getVHost

```
public IVHost getVHost()
```

Get vhost

**Returns:**

vhost

#### setVHost

```
public void setVHost(IVHost vhost)
```

Set vhost

**Parameters:**

vhost - vhost

#### touch

```
public void touch(long timecode)
```

Touch this session to keep it active.

**Parameters:**

timecode - timecode of touch - System.currentTimeMillis();

#### isTimeout

```
public boolean isTimeout(long timecode)
```

---

(continued from last page)

Return true if this session is timed out.

**Parameters:**

timecode - last touch - System.currentTimeMillis();

**Returns:**

true, if timed out

---

## getLock

```
public Object getLock()
```

Get the synchronization lock for this HTTP session

**Returns:**

synchronization lock

---

## shutdown

```
public void shutdown()
```

Called then the HTTP session is shutting down

---

## isActive

```
public boolean isActive()
```

Is this session active, false after shutdown

**Returns:**

true if session is active

---

## setActive

```
public void setActive(boolean isActive)
```

Set session active

**Parameters:**

isActive - session active

---

## getStream

```
public IMediaStream getStream()
```

Get the IMediaStream associated with this HTTP session

**Returns:**

IMediaStream associated with this HTTP session

---

## setStream

```
public void setStream(IMediaStream stream)
```

Set the IMediaStream associated with this HTTP session

**Parameters:**

stream - IMediaStream associated with this HTTP session

---

## isTimeoutSession

```
public boolean isTimeoutSession()
```

Get is session timeout.

**Returns:**

true if this session is timed out

---

## setTimeoutSession

```
public void setTimeoutSession(boolean timeoutSession)
```

Set is session timeout.

**Parameters:**

timeoutSession - true if this session is timed out

---

## getSessionTimeout

```
public int getSessionTimeout()
```

Get the session timeout for this session (milliseconds)

**Returns:**

session timeout for this session (milliseconds)

---

## setSessionTimeout

```
public void setSessionTimeout(int sessionTimeout)
```

Set the session timeout for this session (milliseconds)

**Parameters:**

sessionTimeout - session timeout for this session (milliseconds)

---

## isValidated

```
public boolean isValidated()
```

Has this session been validated

**Returns:**

true, if validated

---

## checkAndSetPlayLogged

```
public boolean checkAndSetPlayLogged()
```

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()
```

(continued from last page)

true, if play has been logged

**Returns:**

true, if play has been logged

---

**setPlayLogged**

```
public void setPlayLogged(boolean isPlayLogged)
```

true, if play has been logged

**Parameters:**

isPlayLogged - true, if play has been logged

---

**addIOPerformance**

```
public void addIOPerformance(IOPerformanceCounter totalIOPerformanceResult)
```

Internal user, keep track of IO performance

**Parameters:**

totalIOPerformanceResult - IO performance

---

**addIOPerformance2**

```
public void addIOPerformance2(IOPerformanceCounter totalIOPerformanceResult)
```

Internal user, keep track of IO performance

**Parameters:**

totalIOPerformanceResult - IO performance

---

**getConnectionHolder**

```
public ConnectionHolder getConnectionHolder()
```

Connection holder for this session

**Returns:**

connection holder

---

**getHTTPStreamerAdapter**

```
public IHTTPStreamerAdapter getHTTPStreamerAdapter()
```

Get the HTTP streaming adapter associated with this HTTP session

**Returns:**

HTTP streaming adapter

---

**setHTTPStreamerAdapter**

```
public void setHTTPStreamerAdapter(IHTTPStreamerAdapter httpStreamerAdapter)
```

Set the HTTP streaming adapter associated with this HTTP session

**Parameters:**

httpStreamerAdapter - HTTP streaming adapter

## getAppInstance

```
public IAppliationInstance getAppInstance()
```

Get the application instance associated with this HTTP session

**Returns:**

application instance associated with this HTTP session

## setAppInstance

```
public void setAppInstance(IAppliationInstance 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()
```

---

(continued from last page)

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

(continued from last page)

**Parameters:**

serverIp - server IP address

**getServerPort****public int getServerPort()**

Get server port

**Returns:**

server port

**setServerPort****public void setServerPort(int serverPort)**

Set server port

**Parameters:**

serverPort - server port

**getUserAgent****public String getUserAgent()**

Get user agent

**Returns:**

user agent

**setUserAgent****public void setUserAgent(String userAgent)**

Set user agent

**Parameters:**

userAgent - user agent

**getUri****public String getUri()**

Get the URI associated with initial request

**Returns:**

URI associated with initial request

**setUri****public void setUri(String uri)**

Set the URI associated with initial request

**Parameters:**

uri - URI associated with initial request

## getReferrer

```
public String getreferrer()
```

Get referrer

**Returns:**

referrer

---

## setReferrer

```
public void setreferrer(String referrer)
```

Set referrer

**Parameters:**

referrer - referrer

---

## getQueryStr

```
public String getQueryStr()
```

Get query string

**Returns:**

query string

---

## setQueryStr

```
public void setQueryStr(String queryStr)
```

Set query string

**Parameters:**

queryStr - query string

---

## lockRepeaterStreams

```
public void lockRepeaterStreams(java.util.List streamNames,  
                                String liveStreamPacketizer,  
                                String liveStreamRepeater,  
                                String streamTypeStr)
```

Internal user, lock in repeater streams

**Parameters:**

streamNames - list of stream names  
liveStreamPacketizer - live stream packetizer name  
liveStreamRepeater - live repeater name  
streamTypeStr - stream type

---

## rejectSession

```
public void rejectSession()
```

Reject this HTTP session. No further processing should occur

---

(continued from last page)

## acceptSession

```
public void acceptSession()
```

Accept the HTTP session.

## isAcceptSession

```
public boolean isAcceptSession()
```

Return true if this session has not been rejected

**Returns:**

true if this session has not been rejected

## setAcceptSession

```
public void setAcceptSession(boolean acceptSession)
```

Set to false to reject session

**Parameters:**

acceptSession - false to reject session

## getCookieStr

```
public String getCookieStr()
```

Get cookie string

**Returns:**

cookie string

## setCookieStr

```
public void setCookieStr(String cookieStr)
```

Set cookie string

**Parameters:**

cookieStr - cookie string

## getStreamName

```
public String getStreamName()
```

Get stream name

**Returns:**

stream name

## setStreamName

```
public void setStreamName(String streamName)
```

Set stream name

**Parameters:**

(continued from last page)

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:**

streamName - stream name

streamNameParts - stream name parts

---

## getStreamPosition

```
public long getStreamPosition()
```

---

(continued from last page)

Get stream position

**Returns:**

stream position

---

## setStreamPosition

```
public void setStreamPosition(long streamPosition)
```

Set stream position (will not cause seek)

**Parameters:**

streamPosition - stream position

---

## getIOPerformanceCounter

```
public IOPerformanceCounter getIOPerformanceCounter()
```

Get IO performance counter

**Returns:**

IO performance counter

---

## getFileInfo

```
public HTTPStreamerFileInfo getFileInfo(String streamName)
```

Get the file information if video on demand streaming

**Parameters:**

streamName - stream name

**Returns:**

file information

---

## putFileInfo

```
public void putFileInfo(String streamName,  
                      HTTPStreamerFileInfo fileInfo)
```

Set the file information

**Parameters:**

streamName - stream name

fileInfo - file information

---

## isFileInfo

```
public boolean isFileInfo(String streamName)
```

Return true if the is file information for a given stream name

**Parameters:**

streamName - stream name

**Returns:**

true if the is file information for a given stream name

---

## isValidStreamDomainStr

```
public boolean isValidStreamDomainStr(String streamDomainStr)
```

Internal user, is stream name valid for HTTP session

**Parameters:**

streamDomainStr - stream name

**Returns:**

true if valid

---

## validStreamDomainToString

```
public String validStreamDomainToString()
```

Return the valid domain strings as a string

**Returns:**

valid domain strings as a string

---

## containsStreamDomainStr

```
public boolean containsStreamDomainStr(String streamDomainStr)
```

Internal user, test stream name

**Parameters:**

streamDomainStr - stream name

**Returns:**

true if valid

---

## removeStreamDomainStr

```
public void removeStreamDomainStr(String streamDomainStr)
```

Internal user, remove stream name

**Parameters:**

streamDomainStr - stream name

---

## addStreamDomainStr

```
public void addStreamDomainStr(String streamDomainStr)
```

Internal user, add stream name

**Parameters:**

streamDomainStr - stream name

---

## addStreamDomainStrs

```
public void addStreamDomainStrs(java.util.List streamNames)
```

Internal user, add stream names

(continued from last page)

**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)**

This query string will be added to URLs used in HTTP streaming

**Parameters:**

userQueryStr - user query string

(continued from last page)

## setDvrSessionInfo

```
public void setDvrSessionInfo(DvrSessionInfo dvr)
```

This information is used to manage the connection to the DVR store.

**Parameters:**

dvr - The DVR session info.

---

## getDvrSessionInfo

```
public DvrSessionInfo getDvrSessionInfo()
```

This information is used to manage the connection to the DVR store.

**Returns:**

DVR session info.

---

## extractHTTPRequestInfo

```
public void extractHTTPRequestInfo(IHTTPRequest req)
```

Extract information from HTTP request

**Parameters:**

req - HTTP request

---

## getHTTPHeaderMap

```
public java.util.Map getHTTPHeaderMap()
```

Get a copy of the HTTP request header map

**Returns:**

copy of the HTTP request header map

---

## getHTTPHeader

```
public String getHTTPHeader(String name)
```

Get a HTTP header value such as 'Content-Length'

**Parameters:**

name - header name

**Returns:**

header value

---

## getHTTPIntHeader

```
public int getHTTPIntHeader(String name)
```

Get a HTTP header value such as 'Content-Length' and return as int

**Parameters:**

name - header name

**Returns:**

(continued from last page)

header value

---

## getHTTPHeaderNames

```
public java.util.Set getHTTPHeaderNames()
```

Get a Set of the header names

**Returns:**

Set of header names

---

## getPlayStart

```
public long getPlayStart()
```

Get the play start time offset (milliseconds) for video on demand playback.

**Returns:**

play start time offset (milliseconds)

---

## setPlayStart

```
public void setPlayStart(long playStart)
```

Set the play start time offset (milliseconds) for video on demand playback.

**Parameters:**

playStart - play start time offset (milliseconds)

---

## getPlayDuration

```
public long getPlayDuration()
```

Get the play duration (milliseconds) for video on demand playback. A values of -1 means play to end.

**Returns:**

play duration (milliseconds)

---

## setPlayDuration

```
public void setPlayDuration(long playDuration)
```

Set the play duration (milliseconds) for video on demand playback. A values of -1 means play to end.

**Parameters:**

playDuration - play duration (milliseconds)

---

## getPlaySeek

```
public long getPlaySeek()
```

Get the seek location (milliseconds) where playback should start

**Returns:**

seek location (milliseconds)

---

(continued from last page)

## setPlaySeek

```
public void setPlaySeek(long playSeek)
```

Set the seek location (milliseconds) where playback should start

**Parameters:**

playSeek - seek location (milliseconds)

---

## setUserHTTPHeader

```
public void setUserHTTPHeader(String name,  
                           String value)
```

Set user HTTP header. This header value will be added to all HTTP responses

**Parameters:**

name - name

value - value

---

## getUserHTTPHeaders

```
public java.util.Map getUserHTTPHeaders()
```

Get user HTTP header. This header value will be added to all HTTP responses

**Returns:**

header map

---

## addUserHTTPHeaders

```
public void addUserHTTPHeaders(IHTTPResponse resp)
```

Internal

**Parameters:**

resp - response

---

## doSessionRedirect

```
public void doSessionRedirect(IHTTPResponse resp)
```

Internal

**Parameters:**

resp - response

---

## doSessionRedirect

```
public void doSessionRedirect(IHTTPResponse resp,  
                           String redirectSessionURL,  
                           String redirectSessionContentType,  
                           int redirectSessionCode)
```

Internal

**Parameters:**

resp - response

redirectSessionURL - redirect session URL

(continued from last page)

`redirectSessionContentType`  
`redirectSessionCode` - redirect session response code (default 302)

---

## isRedirectSession

`public boolean isRedirectSession()`

Is session redirect

**Returns:**

session redirect

---

## setRedirectSession

`public void setRedirectSession(boolean redirectSession)`

Set session redirect

**Parameters:**

`redirectSession` - session redirect

---

## getRedirectSessionCode

`public int getRedirectSessionCode()`

Get session redirect HTTP response code (default 302)

**Returns:**

session redirect HTTP response code

---

## setRedirectSessionCode

`public void setRedirectSessionCode(int redirectSessionCode)`

Set session redirect HTTP response code (default 302)

**Parameters:**

`redirectSessionCode` - session redirect HTTP response code

---

## getRedirectSessionURL

`public String getRedirectSessionURL()`

Get redirect session URL

**Returns:**

redirect session URL

---

## setRedirectSessionURL

`public void setRedirectSessionURL(String redirectSessionURL)`

Set redirect session URL

**Parameters:**

`redirectSessionURL` - redirect session URL

(continued from last page)

## redirectSession

```
public void redirectSession(String redirectSessionURL)
```

Redirect session

**Parameters:**

redirectSessionURL - redirect session URL

## redirectSession

```
public void redirectSession(String redirectSessionURL,  
                           int redirectSessionCode)
```

Redirect session

**Parameters:**

redirectSessionURL - redirect session URL

redirectSessionCode - redirect session response code (default 302)

## getRedirectSessionBody

```
public byte[] getRedirectSessionBody()
```

Get redirect session body

**Returns:**

redirect session body

## setRedirectSessionBody

```
public void setRedirectSessionBody(byte[] redirectSessionBody)
```

Set redirect session body

**Parameters:**

redirectSessionBody - redirect session body

## getRedirectSessionContentType

```
public String getRedirectSessionContentType()
```

Get redirect session HTTP Content-Type

**Returns:**

redirect session HTTP Content-Type

## setRedirectSessionContentType

```
public void setRedirectSessionContentType(String redirectSessionContentType)
```

Set redirect session HTTP Content-Type

**Parameters:**

redirectSessionContentType

(continued from last page)

## getLastRequest

```
public long getLastRequest()
```

Get the last timestamp of the last Io request

**Returns:**

last timestamp of the last Io request (milliseconds)

---

## isHTTPOrigin

```
public boolean isHTTPOrigin()
```

Return true if session in an HTTP origin session

**Returns:**

true if session in an HTTP origin session

---

## setHTTPOrigin

```
public void setHTTPOrigin(boolean isHTTPOrigin)
```

Set to true if HTTP origin session

**Parameters:**

isHTTPOrigin - true if HTTP origin session

---

## getAndClearNotifyCreate

```
public boolean getAndClearNotifyCreate()
```

---

## getTranscoderVODSession

```
public com.wowza.wms.transcoder.vod.TranscoderVODSession getTranscoderVODSession()
```

Get the vod transcoder session associated with this HTTP session

**Returns:**

vod transcoder session

---

## setTranscoderVODSession

```
public void setTranscoderVODSession(com.wowza.wms.transcoder.vod.TranscoderVODSession transcoderVODSession)
```

Set the vod transcoder session associated with this HTTP session

**Parameters:**

transcoderVODSession - vod transcoder session

---

## getVODTranscodeNGRP

```
public String getVODTranscodeNGRP()
```

Get the VOD Transcoder NGRP name

**Returns:**

(continued from last page)

VOD Transcoder NGRP name

---

## setVODTranscodeNGRP

```
public void setVODTranscodeNGRP(String vodTranscodeNGRP)
```

Set the VOD Transcoder NGRP name

**Parameters:**

vodTranscodeNGRP - VOD Transcoder NGRP name

---

## getTranscoderVODIndex

```
public IHTTPStreamerTranscoderVODIndex getTranscoderVODIndex()
```

Get the VOD Transcoder index information

**Returns:**

VOD Transcoder index

---

## setTranscoderVODIndex

```
public void setTranscoderVODIndex(IHTTPStreamerTranscoderVODIndex transcoderVODIndex)
```

Set the VOD Transcoder index information

**Parameters:**

transcoderVODIndex - VOD Transcoder index

---

## getTranscoderVODIndexDestinationsOrder

```
public java.util.List getTranscoderVODIndexDestinationsOrder()
```

Get the list of VOD Transcoder destinations in playlist order

**Returns:**

list of VOD Transcoder destinations

---

## getTranscoderVODIndexDestinationsMap

```
public java.util.Map getTranscoderVODIndexDestinationsMap()
```

Get the VOD Transcoder destination map

**Returns:**

VOD Transcoder destination map

---

## indexTranscoderVOD

```
public void indexTranscoderVOD(IHTTPStreamerApplicationContext appContext,  
String vodTranscodeNGRP)
```

Index the HTTP session for VOD transcoder streaming

**Parameters:**

appContext - application context

vodTranscodeNGRP - transcoder name group

---

## getTranscoderVODIndex

```
public IHTTPStreamerTranscoderVODDestinationIndex getTranscoderVODIndex(String  
bitrateStr)
```

Get VOD transcoder index for given bitrate

**Parameters:**

bitrateStr - bitrate string

**Returns:**

VOD transcoder index

## com.wowza.wms.httpstreamer.model Interface IHTTPStreamerSessionNotify

---

public interface IHTTPStreamerSessionNotify  
extends

IHTTPStreamerSessionNotify: HTTP session create/destroy interface

---

### Method Summary

void	<a href="#">onHTTPStreamerSessionCreate(IHTTPStreamerSession httpStreamerSession)</a> Called when an HTTP streaming session is created
void	<a href="#">onHTTPStreamerSessionDestroy(IHTTPStreamerSession httpStreamerSession)</a> Called when an HTTP streaming session is destroyed

---

### Methods

#### onHTTPStreamerSessionCreate

public void [onHTTPStreamerSessionCreate\(IHTTPStreamerSession httpStreamerSession\)](#)

Called when an HTTP streaming session is created

**Parameters:**

httpStreamerSession - HTTP streaming session

---

#### onHTTPStreamerSessionDestroy

public void [onHTTPStreamerSessionDestroy\(IHTTPStreamerSession httpStreamerSession\)](#)

Called when an HTTP streaming session is destroyed

**Parameters:**

httpStreamerSession - HTTP streaming session

## com.wowza.wms.httpstreamer.model Interface IHTTPStreamerTranscoderVODDestinationIndex

public interface IHTTPStreamerTranscoderVODDestinationIndex  
extends

### Method Summary

int	<a href="#">getBitrate()</a>
com.wowza.wms.transco der.vod.TranscoderVOD Destination	<a href="#">getTranscoderVODDestination()</a>
void	<a href="#">setBitrate(int bitrate)</a>
void	<a href="#">setTranscoderVODDestination(com.wowza.wms.transcoder.vod.TranscoderVODD estination transcoderVODDestination)</a>

### Methods

#### **getTranscoderVODDestination**

```
public com.wowza.wms.transcoder.vod.TranscoderVODDestination  
getTranscoderVODDestination()
```

#### **setTranscoderVODDestination**

```
public void  
setTranscoderVODDestination(com.wowza.wms.transcoder.vod.TranscoderVODDestination  
transcoderVODDestination)
```

#### **getBitrate**

```
public int getBitrate()
```

#### **setBitrate**

```
public void setBitrate(int bitrate)
```

**com.wowza.wms.httpstreamer.model  
Interface IHTTPStreamerTranscoderVODIndex**

---

public interface IHTTPStreamerTranscoderVODIndex  
extends

---

---

## com.wowza.wms.httpstreamer.model Interface ILiveStreamPacketizerChunkWriter

public interface ILiveStreamPacketizerChunkWriter  
extends

### Method Summary

void	<a href="#">addToChunk(LiveStreamPacketizerPacketHolder holder)</a>
void	<a href="#">endChunk(long timecode)</a>
boolean	<a href="#">isPacketizeAudio()</a>
boolean	<a href="#">isPacketizeData()</a>
boolean	<a href="#">isPacketizeVideo()</a>
boolean	<a href="#">isValidAudioCodec(int codec)</a>
boolean	<a href="#">isValidVideoCodec(int codec)</a>
void	<a href="#">resetStream()</a>
void	<a href="#">setCodecInfoAudio(com.wowza.wms.media.model.MediaCodecInfoAudio codecInfoAudio)</a>
void	<a href="#">setCodecInfoVideo(com.wowza.wms.media.model.MediaCodecInfoVideo codecInfoVideo)</a>
void	<a href="#">startChunk(int streamMode, int videoCodecId, int audioCodecId, long timecode, String originRepeaterContext, com.wowza.wms.drm.cenc.CencInfo cencInfo)</a>

### Methods

#### startChunk

```
public void startChunk(int streamMode,
                      int videoCodecId,
                      int audioCodecId,
                      long timecode,
                      String originRepeaterContext,
                      com.wowza.wms.drm.cenc.CencInfo cencInfo)
```

(continued from last page)

**endChunk**

```
public void endChunk(long timecode)
```

---

**addToChunk**

```
public void addToChunk(LiveStreamPacketizerPacketHolder holder)
```

---

**setCodecInfoAudio**

```
public void setCodecInfoAudio(com.wowza.wms.media.model.MediaCodecInfoAudio  
codecInfoAudio)
```

---

**setCodecInfoVideo**

```
public void setCodecInfoVideo(com.wowza.wms.media.model.MediaCodecInfoVideo  
codecInfoVideo)
```

---

**resetStream**

```
public void resetStream()
```

---

**isPacketizeAudio**

```
public boolean isPacketizeAudio()
```

---

**isPacketizeVideo**

```
public boolean isPacketizeVideo()
```

---

**isPacketizeData**

```
public boolean isPacketizeData()
```

---

**isValidAudioCodec**

```
public boolean isValidAudioCodec(int codec)
```

---

(continued from last page)

## isValidVideoCodec

```
public boolean isValidVideoCodec(int codec)
```

---

Package

**com.wowza.wms.httpstreamer.mpegdashstreaming.httpstreamer**

## com.wowza.wms.httpstreamer.mpegdashstreaming.httpstreamer Class HTTPStreamerSessionMPEGDash

```
java.lang.Object
+-com.wowza.wms.httpstreamer.model.HTTPStreamerSessionBase
+-+
com.wowza.wms.httpstreamer.mpegdashstreaming.httpstreamer.HTTPStreamerSessionMPEGDash
```

### All Implemented Interfaces:

[IHTTPSessionNotify](#), [IHTTPStreamerSession](#)

```
public class HTTPStreamerSessionMPEGDash
extends HTTPStreamerSessionBase
implements IHTTPStreamerSession, IHTTPSessionNotify
```

### Fields inherited from class com.wowza.wms.httpstreamer.model.HTTPStreamerSessionBase

```
acceptSession, appInstance, connectionHolder, cookieStr, DATEFORMAT, elapsedTime,
fastDateFormat, fileInfoMap, firstCheck, httpHeaders, httpStreamerAdapter,
ioPerformanceCounter, ipAddress, isActive, isHTTPOrigin, isPlayLogged, lastRequest,
liveStreamingPacketizer, lock, mediaCasterStreamLock, notifyCreate, playDuration, playSeek,
playStart, properties, queryStr, redirectSession, redirectSessionBody, redirectSessionCode,
redirectSessionContentType, redirectSessionURL, referrer, serverIp, serverPort, sessionId,
sessionProtocol, sessionTimeout, sessionType, stream, streamDomainStrSet, streamExt,
streamName, streamNamePartMap, streamPosition, timeoutSession, totalIOPerformance2Last,
totalIOPerformanceLast, transcoderVODIndex, transcoderVODIndexDestinationsMap,
transcoderVODIndexDestinationsOrder, transcoderVODSession, uri, userAgent, userHTTPHeaders,
userQueryStr, vhost, vodTranscodeNGRP
```

### Fields inherited from interface com.wowza.wms.httpstreamer.model.IHTTPStreamerSession

```
SESSIONPROTOCOL_COUNT, SESSIONPROTOCOL_CUPERTINOSTREAMING,
SESSIONPROTOCOL_DIRECTSTREAMING, SESSIONPROTOCOL_DVRCHUNKSTREAMING,
SESSIONPROTOCOL_MPEGDASHSTREAMING, SESSIONPROTOCOL_SANJOSESTREAMING,
SESSIONPROTOCOL_SMOOTHSTREAMING, SESSIONPROTOCOL_UNKNOWN, SESSIONPROTOCOL_WEBMSTREAMING,
SESSIONTYPE_LIVE, SESSIONTYPE_LIVEDVR, SESSIONTYPE_UNKNOWN, SESSIONTYPE_VOD
```

## Constructor Summary

public	<a href="#">HTTPStreamerSessionMPEGDash()</a>
--------	---

## Method Summary

void	<a href="#">clearLoggingValues()</a>
boolean	<a href="#">containsIndex(String streamName)</a>
static IHTTPStreamerMPEGDash Index	<a href="#">createIndexLive(IHTTPStreamerApplicationContext appContext, IHTTPStreamerSession httpStreamerSession, String rawStreamName, String streamExt, String streamName)</a>

IHTTPStreamerMPEGDash Index	<code>getIndex(IHTTPStreamerApplicationContext appContext, IHTTPStreamerSession httpStreamerSession, String rawStreamName, String streamExt, String streamName, long playStart, long playDuration, long playSeek, TimedTextRequest captionRequest, String vodTranscodeNGRP)</code>
IHTTPStreamerMPEGDash Index	<code>getIndexLive(IHTTPStreamerApplicationContext appContext, IHTTPStreamerSession httpStreamerSession, String rawStreamName, String streamExt, String streamName)</code>
void	<code>indexTranscoderVOD(IHTTPStreamerApplicationContext appContext, String vodTranscodeNGRP)</code>
void	<code>logLiveChunk(MPEGDashDashChunk chunk)</code>
void	<code>logVODChunk(MPEGDashDashChunk chunk)</code>
void	<code>notifyHTTPSessionCreate(IApplicationInstance appInstance, IHTTPStreamerSession httpStreamerSession)</code>
void	<code>notifyHTTPSessionDestroy(IApplicationInstance appInstance, IHTTPStreamerSession httpStreamerSession)</code>
void	<code>shutdown()</code>
void	<code>updateLoggingValues()</code>

Methods inherited from class com.wowza.wms.httpstreamer.model.HTTPStreamerSessionBase

```

acceptSession, addIOPerformance, addIOPerformance2, addStreamDomainStr,
addStreamDomainStrs, addUserHTTPHeaders, checkAndSetPlayLogged, clearLoggingValues,
containsStreamDomainStr, containsStreamNameParts, doSessionRedirect,
doSessionRedirect, extractHTTPRequestInfo, getAndClearNotifyCreate, getAppInstance,
getConnectionHolder, getCookieStr, getDvrSessionInfo, getElapsedTime, getFileInfo,
getHTTPDate, getHTTPHeader, getHTTPHeaderMap, getHTTPHeaderNames, getHTTPIntHeader,
getHTTPStreamerAdapter, getIOPerformanceCounter, getIpAddress, getLastRequest,
getLiveStreamingPacketizer, getLock, getPlayDuration, getPlaySeek, getPlayStart,
getProperties, getQueryStr, getRedirectSessionBody, getRedirectSessionCode,
getRedirectSessionContentType, getRedirectSessionURL, getReferrer, getServerIp,
getServerPort, getSessionId, getSessionProtocol, getSessionTimeout, getSessionType,
getStream, getStreamDomainStr, getStreamDomainStrList, getStreamExt, getStreamName,
getStreamNameParts, getStreamPosition, getTimeRunning, getTimeRunningSeconds,
getTranscoderVODIndex, getTranscoderVODIndex, getTranscoderVODIndexDestinationsMap,
getTranscoderVODIndexDestinationsOrder, getTranscoderVODSession, getUri,
getUserAgent, getUserHTTPHeaders, getUserQueryStr, getVHost, getVODTranscodeNGRP,
indexTranscoderVOD, isAcceptSession, isActive, isFileInfo, isHTTPOrigin,
isPlayLogged, isRedirectSession, isTimeout, isTimeoutSession, isvalidated,
isValidStreamDomainStr, lockRepeaterStreams, putFileInfo, putStreamNameParts,
redirectSession, redirectSession, rejectSession, removeStreamDomainStr,
setAcceptSession, setActive, setAppInstance, setCookieStr, setDvrSessionInfo,
setHTTPOrigin, setHTTPStreamerAdapter, setIpAddress, setLiveStreamingPacketizer,
setPlayDuration, setPlayLogged, setPlaySeek, setPlayStart, setQueryStr,
setRedirectSession, setRedirectSessionBody, setRedirectSessionCode,
setRedirectSessionContentType, setRedirectSessionURL, setReferrer, setServerIp,
setServerPort, sessionId, sessionProtocol, sessionTimeout, sessionType,
setStream, setStreamExt, setStreamName, setStreamPosition, setThreadContext,
setTimeoutSession, setTranscoderVODIndex, setTranscoderVODSession, setUri,
setUserAgent, setUserHTTPHeader, setUserQueryStr, setValidated, setVHost,
setVODTranscodeNGRP, shutdown, shutdownLocked, touch, updateLoggingValues,
validStreamDomainToString

```

**Methods inherited from class java.lang.Object**

```

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait,
wait

```

**Methods inherited from interface [com.wowza.wms.httpstreamer.model.IHTTPStreamerSession](#)**

```

acceptSession, addIOPerformance, addIOPerformance2, addStreamDomainStr,
addStreamDomainStrs, addUserHTTPHeaders, checkAndSetPlayLogged, clearLoggingValues,
containsStreamDomainStr, containsStreamNameParts, doSessionRedirect,
doSessionRedirect, extractHTTPRequestInfo, getAndClearNotifyCreate, getAppInstance,
getConnectionHolder, getCookieStr, getDvrSessionInfo, getElapsedTime, getFileInfo,
getHTTPHeader, getHTTPHeaderMap, getHTTPHeaderNames, getHTTPIntHeader,
getHTTPStreamerAdapter, getIOPerformanceCounter, getIpAddress, getLastRequest,
getLiveStreamingPacketizer, getLock, getPlayDuration, getPlaySeek, getPlayStart,
getProperties, getQueryStr, getRedirectSessionBody, getRedirectSessionCode,
getRedirectSessionContentType, getRedirectSessionURL, getReferrer, getServerIp,
getServerPort, getSessionId, getSessionProtocol, getSessionTimeout, getSessionType,
getStream, getStreamExt, getStreamName, getStreamNameParts, getStreamPosition,
getTimeRunning, getTimeRunningSeconds, getTranscoderVODIndex, getTranscoderVODIndex,
getTranscoderVODIndexDestinationsMap, getTranscoderVODIndexDestinationsOrder,
getTranscoderVODSession, getUri, getUserAgent, getUserHTTPHeaders, getUserQueryStr,
getVHost, getVODTranscodeNGRP, indexTranscoderVOD, isAcceptSession, isActive,
isFileInfo, isHTTPOrigin, isPlayLogged, isRedirectSession, isTimeout,
isTimeoutSession, isvalidated, isValidStreamDomainStr, lockRepeaterStreams,
putFileInfo, putStreamNameParts, redirectSession, redirectSession, rejectSession,
removeStreamDomainStr, setAcceptSession, setActive, setAppInstance, setCookieStr,
setDvrSessionInfo, setHTTPOrigin, setHTTPStreamerAdapter, setIpAddress,
setLiveStreamingPacketizer, setPlayDuration, setPlayLogged, setPlaySeek,
setPlayStart, setQueryStr, setRedirectSession, setRedirectSessionBody,
setRedirectSessionCode, setRedirectSessionContentType, setRedirectSessionURL,
setReferrer, setServerIp, setServerPort, sessionId, sessionProtocol,
getSessionTimeout, getSessionType, setStream, setStreamExt, setStreamName,
setStreamPosition, setTimeoutSession, setTranscoderVODIndex, setTranscoderVODSession,
setUri, setUserAgent, setUserHTTPHeader, setUserQueryStr, setVHost,
setVODTranscodeNGRP, shutdown, touch, updateLoggingValues, validStreamDomainToString

```

Methods inherited from interface [com.wowza.wms.httpstreamer.model.IHTTPSessionNotify](#)

[notifyHTTPSessionCreate](#), [notifyHTTPSessionDestroy](#)

## Constructors

### HTTPStreamerSessionMPEGDash

```
public HTTPStreamerSessionMPEGDash()
```

## Methods

### shutdown

```
public void shutdown()
```

### containsIndex

```
public boolean containsIndex(String streamName)
```

**createIndexLive**

```
public static IHTTPStreamerMPEGDashIndex  
createIndexLive(IHTTPStreamerApplicationContext appContext,  
    IHTTPStreamerSession httpStreamerSession,  
    String rawStreamName,  
    String streamExt,  
    String streamName)
```

---

**getIndexLive**

```
public IHTTPStreamerMPEGDashIndex getIndexLive(IHTTPStreamerApplicationContext  
appContext,  
    IHTTPStreamerSession httpStreamerSession,  
    String rawStreamName,  
    String streamExt,  
    String streamName)
```

---

**getIndex**

```
public IHTTPStreamerMPEGDashIndex getIndex(IHTTPStreamerApplicationContext appContext,  
    IHTTPStreamerSession httpStreamerSession,  
    String rawStreamName,  
    String streamExt,  
    String streamName,  
    long playStart,  
    long playDuration,  
    long playSeek,  
    TimedTextRequest captionRequest,  
    String vodTranscodeNGRP)
```

---

**updateLoggingValues**

```
public void updateLoggingValues()
```

---

**clearLoggingValues**

```
public void clearLoggingValues()
```

---

**logLiveChunk**

```
public void logLiveChunk(MPEGDashDashChunk chunk)
```

---

**logVODChunk**

```
public void logVODChunk(MPEGDashDashChunk chunk)
```

---

### notifyHTTPSessionCreate

```
public void notifyHTTPSessionCreate(IApplicationInstance appInstance,  
IHTTPStreamerSession httpStreamerSession)
```

---

### notifyHTTPSessionDestroy

```
public void notifyHTTPSessionDestroy(IApplicationInstance appInstance,  
IHTTPStreamerSession httpStreamerSession)
```

---

### indexTranscoderVOD

```
public void indexTranscoderVOD(IHTTPStreamerApplicationContext appContext,  
String vodTranscodeNGRP)
```

---

**Package**

**com.wowza.wms.httpstreamer.sanjosestreaming.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, notifyCreate, playDuration, playSeek,
playStart, properties, queryStr, redirectSession, redirectSessionBody, redirectSessionCode,
redirectSessionContentType, redirectSessionURL, referrer, serverIp, serverPort, sessionId,
sessionProtocol, sessionTimeout, sessionType, stream, streamDomainStrSet, streamExt,
streamName, streamNamePartMap, streamPosition, timeoutSession, totalIOPerformance2Last,
totalIOPerformanceLast, transcoderVODIndex, transcoderVODIndexDestinationsMap,
transcoderVODIndexDestinationsOrder, transcoderVODSession, uri, userAgent, userHTTPHeaders,
userQueryStr, vhost, vodTranscodeNGRP
```

### Fields inherited from interface com.wowza.wms.httpstreamer.model.IHTTPStreamerSession

```
SESSIONPROTOCOL_COUNT, SESSIONPROTOCOL_CUPERTINOSTREAMING,
SESSIONPROTOCOL_DIRECTSTREAMING, SESSIONPROTOCOL_DVRCHUNKSTREAMING,
SESSIONPROTOCOL_MPEGDASHSTREAMING, SESSIONPROTOCOL_SANJOSESTREAMING,
SESSIONPROTOCOL_SMOOTHSTREAMING, SESSIONPROTOCOL_UNKNOWN, SESSIONPROTOCOL_WEBMSTREAMING,
SESSIONTYPE_LIVE, SESSIONTYPE_LIVEDVR, SESSIONTYPE_UNKNOWN, SESSIONTYPE_VOD
```

## Constructor Summary

public	<a href="#">HTTPStreamerSessionSanJose()</a>
--------	--

## Method Summary

void	<a href="#">addFirstABSTRequest(String streamName)</a>
------	--

void	<a href="#">clearLoggingValues()</a>
------	--------------------------------------

boolean	<a href="#">containsIndex(String streamName)</a>
---------	--

<code>static IHTTPStreamerSanJoseIndex</code>	<code>createIndexLive(<a href="#">IHTTPStreamerApplicationContext</a> appContext, <a href="#">IHTTPStreamerSession</a> httpStreamerSession, String rawStreamName, String streamExt, String streamName)</code>
<code>IHTTPStreamerSanJoseIndex</code>	<code>getIndex(<a href="#">IHTTPStreamerApplicationContext</a> appContext, <a href="#">IHTTPStreamerSession</a> httpStreamerSession, String rawStreamName, String streamExt, String streamName, long playStart, long playDuration, long playSeek, TimedTextRequest captionRequest, String vodTranscodeNGRP)</code>
<code>IHTTPStreamerSanJoseIndex</code>	<code>getIndexLive(<a href="#">IHTTPStreamerApplicationContext</a> appContext, <a href="#">IHTTPStreamerSession</a> httpStreamerSession, String rawStreamName, String streamExt, String streamName)</code>
<code>void</code>	<code><u>indexTranscoderVOD</u>(<a href="#">IHTTPStreamerApplicationContext</a> appContext, String vodTranscodeNGRP)</code>
<code>boolean</code>	<code><u>isFirstABSTRequest</u>(String streamName)</code>
<code>void</code>	<code><u>logLiveChunk</u>(LiveStreamPacketizerSanJoseChunk chunk)</code>
<code>void</code>	<code><u>logVODChunk</u>(LiveStreamPacketizerSanJoseChunk chunk)</code>
<code>void</code>	<code><u>notifyHTTPSessionCreate</u>(<a href="#">IAplicationInstance</a> appInstance, <a href="#">IHTTPStreamerSession</a> httpStreamerSession)</code>
<code>void</code>	<code><u>notifyHTTPSessionDestroy</u>(<a href="#">IAplicationInstance</a> appInstance, <a href="#">IHTTPStreamerSession</a> httpStreamerSession)</code>
<code>void</code>	<code><u>shutdown</u>()</code>
<code>void</code>	<code><u>updateLoggingValues</u>()</code>

Methods inherited from class com.wowza.wms.httpstreamer.model.HTTPStreamerSessionBase

```

acceptSession, addIOPerformance, addIOPerformance2, addStreamDomainStr,
addStreamDomainStrs, addUserHTTPHeaders, checkAndSetPlayLogged, clearLoggingValues,
containsStreamDomainStr, containsStreamNameParts, doSessionRedirect,
doSessionRedirect, extractHTTPRequestInfo, getAndClearNotifyCreate, getAppInstance,
getConnectionHolder, getCookieStr, getDvrSessionInfo, getElapsedTime, getFileInfo,
getHTTPDate, getHTTPHeader, getHTTPHeaderMap, getHTTPHeaderNames, getHTTPIntHeader,
getHTTPStreamerAdapter, getIOPerformanceCounter, getIpAddress, getLastRequest,
getLiveStreamingPacketizer, getLock, getPlayDuration, getPlaySeek, getPlayStart,
getProperties, getQueryStr, getRedirectSessionBody, getRedirectSessionCode,
getRedirectSessionContentType, getRedirectSessionURL, getReferrer, getServerIp,
getServerPort, getSessionId, getSessionProtocol, getSessionTimeout, getSessionType,
getStream, getStreamDomainStr, getStreamDomainStrList, getStreamExt, getStreamName,
getStreamNameParts, getStreamPosition, getTimeRunning, getTimeRunningSeconds,
getTranscoderVODIndex, getTranscoderVODIndex, getTranscoderVODIndexDestinationsMap,
getTranscoderVODIndexDestinationsOrder, getTranscoderVODSession, getUri,
getUserAgent, getUserHTTPHeaders, getUserQueryStr, getVHost, getVODTranscodeNGRP,
indexTranscoderVOD, isAcceptSession, isActive, isFileInfo, isHTTPOrigin,
isPlayLogged, isRedirectSession, isTimeout, isTimeoutSession, isValidated,
isValidStreamDomainStr, lockRepeaterStreams, putFileInfo, putStreamNameParts,
redirectSession, redirectSession, rejectSession, removeStreamDomainStr,
setAcceptSession, setActive, setAppInstance, setCookieStr, setDvrSessionInfo,
setHTTPOrigin, setHTTPStreamerAdapter, setIpAddress, setLiveStreamingPacketizer,
setPlayDuration, setPlayLogged, setPlaySeek, setPlayStart, setQueryStr,
setRedirectSession, setRedirectSessionBody, setRedirectSessionCode,
setRedirectSessionContentType, setRedirectSessionURL, setReferrer, setServerIp,
setServerPort, setSessionId, setSessionProtocol, setSessionTimeout, setSessionType,
setStream, setStreamExt, setStreamName, setStreamPosition, setThreadContext,
setTimeoutSession, setTranscoderVODIndex, setTranscoderVODSession, setUri,
setUserAgent, setUserHTTPHeader, setUserQueryStr, setValidated, setVHost,
setVODTranscodeNGRP, shutdown, shutdownLocked, touch, updateLoggingValues,
validStreamDomainToString

```

**Methods inherited from class java.lang.Object**

```

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait,
wait

```

**Methods inherited from interface [com.wowza.wms.httpstreamer.model.IHTTPStreamerSession](#)**

```

acceptSession, addIOPerformance, addIOPerformance2, addStreamDomainStr,
addStreamDomainStrs, addUserHTTPHeaders, checkAndSetPlayLogged, clearLoggingValues,
containsStreamDomainStr, containsStreamNameParts, doSessionRedirect,
doSessionRedirect, extractHTTPRequestInfo, getAndClearNotifyCreate, getAppInstance,
getConnectionHolder, getCookieStr, getDvrSessionInfo, getElapsedTime, getFileInfo,
getHTTPHeader, getHTTPHeaderMap, getHTTPHeaderNames, getHTTPIntHeader,
getHTTPStreamerAdapter, getIOPerformanceCounter, getIpAddress, getLastRequest,
getLiveStreamingPacketizer, getLock, getPlayDuration, getPlaySeek, getPlayStart,
getProperties, getQueryStr, getRedirectSessionBody, getRedirectSessionCode,
getRedirectSessionContentType, getRedirectSessionURL, getReferrer, getServerIp,
getServerPort, getSessionId, getSessionProtocol, getSessionTimeout, getSessionType,
getStream, getStreamExt, getStreamName, getStreamNameParts, getStreamPosition,
getTimeRunning, getTimeRunningSeconds, getTranscoderVODIndex, getTranscoderVODIndex,
getTranscoderVODIndexDestinationsMap, getTranscoderVODIndexDestinationsOrder,
getTranscoderVODSession, getUri, getUserAgent, getUserHTTPHeaders, getUserQueryStr,
getVHost, getVODTranscodeNGRP, indexTranscoderVOD, isAcceptSession, isActive,
isFileInfo, isHTTPOrigin, isPlayLogged, isRedirectSession, isTimeout,
isTimeoutSession, isvalidated, isValidStreamDomainStr, lockRepeaterStreams,
putFileInfo, putStreamNameParts, redirectSession, redirectSession, rejectSession,
removeStreamDomainStr, setAcceptSession, setActive, setAppInstance, setCookieStr,
setDvrSessionInfo, setHTTPOrigin, setHTTPStreamerAdapter, setIpAddress,
setLiveStreamingPacketizer, setPlayDuration, setPlayLogged, setPlaySeek,
setPlayStart, setQueryStr, setRedirectSession, setRedirectSessionBody,
setRedirectSessionCode, setRedirectSessionContentType, setRedirectSessionURL,
setReferrer, setServerIp, setServerPort, sessionId, sessionProtocol,
getSessionTimeout, getSessionType, setStream, setStreamExt, setStreamName,
setStreamPosition, setTimeoutSession, setTranscoderVODIndex, setTranscoderVODSession,
setUri, setUserAgent, setUserHTTPHeader, setUserQueryStr, setVHost,
setVODTranscodeNGRP, shutdown, touch, updateLoggingValues, validStreamDomainToString

```

Methods inherited from interface [com.wowza.wms.httpstreamer.model.IHTTPSessionNotify](#)

[notifyHTTPSessionCreate](#), [notifyHTTPSessionDestroy](#)

## Constructors

### HTTPStreamerSessionSanJose

```
public HTTPStreamerSessionSanJose()
```

## Methods

### shutdown

```
public void shutdown()
```

### containsIndex

```
public boolean containsIndex(String streamName)
```

---

**createIndexLive**

```
public static IHTTPStreamerSanJoseIndex  
createIndexLive(IHTTPStreamerApplicationContext appContext,  
    IHTTPStreamerSession httpStreamerSession,  
    String rawStreamName,  
    String streamExt,  
    String streamName)
```

---

**getIndexLive**

```
public IHTTPStreamerSanJoseIndex getIndexLive(IHTTPStreamerApplicationContext  
appContext,  
    IHTTPStreamerSession httpStreamerSession,  
    String rawStreamName,  
    String streamExt,  
    String streamName)
```

---

**getIndex**

```
public IHTTPStreamerSanJoseIndex getIndex(IHTTPStreamerApplicationContext appContext,  
    IHTTPStreamerSession httpStreamerSession,  
    String rawStreamName,  
    String streamExt,  
    String streamName,  
    long playStart,  
    long playDuration,  
    long playSeek,  
    TimedTextRequest captionRequest,  
    String vodTranscodeNGRP)
```

---

**updateLoggingValues**

```
public void updateLoggingValues()
```

---

**clearLoggingValues**

```
public void clearLoggingValues()
```

---

**logLiveChunk**

```
public void logLiveChunk(LiveStreamPacketizerSanJoseChunk chunk)
```

---

**logVODChunk**

```
public void logVODChunk(LiveStreamPacketizerSanJoseChunk chunk)
```

---

---

### isFirstABSTRequest

```
public boolean isFirstABSTRequest(String streamName)
```

---

### addFirstABSTRequest

```
public void addFirstABSTRequest(String streamName)
```

---

### notifyHTTPSessionCreate

```
public void notifyHTTPSessionCreate(IApplicationInstance appInstance,  
IHTTPStreamerSession httpStreamerSession)
```

---

### notifyHTTPSessionDestroy

```
public void notifyHTTPSessionDestroy(IApplicationInstance appInstance,  
IHTTPStreamerSession httpStreamerSession)
```

---

### indexTranscoderVOD

```
public void indexTranscoderVOD(IHTTPStreamerApplicationContext appContext,  
String vodTranscodeNGRP)
```

---

**Package**

**com.wowza.wms.httpstreamer.smoothstreaming.httpstreamer**

## com.wowza.wms.httpstreamer.smoothstreaming.httpstreamer Class HTTPStreamerSessionSmoothStreamer

```
java.lang.Object
  +- com.wowza.wms.httpstreamer.model.HTTPStreamerSessionBase
    +-+
com.wowza.wms.httpstreamer.smoothstreaming.httpstreamer.HTTPStreamerSessionSmoothStreamer
```

### All Implemented Interfaces:

[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, notifyCreate, playDuration, playSeek,
playStart, properties, queryStr, redirectSession, redirectSessionBody, redirectSessionCode,
redirectSessionContentType, redirectSessionURL, referrer, serverIp, serverPort, sessionId,
sessionProtocol, sessionTimeout, sessionType, stream, streamDomainStrSet, streamExt,
streamName, streamNamePartMap, streamPosition, timeoutSession, totalIOPerformance2Last,
totalIOPerformanceLast, transcoderVODIndex, transcoderVODIndexDestinationsMap,
transcoderVODIndexDestinationsOrder, transcoderVODSession, uri, userAgent, userHTTPHeaders,
userQueryStr, vhost, vodTranscodeNGRP
```

### Fields inherited from interface com.wowza.wms.httpstreamer.model.IHTTPStreamerSession

```
SESSIONPROTOCOL_COUNT, SESSIONPROTOCOL_CUPERTINOSTREAMING,
SESSIONPROTOCOL_DIRECTSTREAMING, SESSIONPROTOCOL_DVRCHUNKSTREAMING,
SESSIONPROTOCOL_MPEGDASHSTREAMING, SESSIONPROTOCOL_SANJOSESTREAMING,
SESSIONPROTOCOL_SMOOTHSTREAMING, SESSIONPROTOCOL_UNKNOWN, SESSIONPROTOCOL_WEBMSTREAMING,
SESSIONTYPE_LIVE, SESSIONTYPE_LIVEDVR, SESSIONTYPE_UNKNOWN, SESSIONTYPE_VOD
```

## Constructor Summary

public	<a href="#">HTTPStreamerSessionSmoothStreamer()</a>
--------	---

## Method Summary

void	<a href="#">clearLoggingValues()</a>
------	--------------------------------------

boolean	<a href="#">containsIndex(String streamName)</a>
---------	--

IHTTPStreamerSmoothStreamerIndex	<code>getIndex(IHTTPStreamerSession httpStreamerSession, IHTTPStreamerApplicationContext appContext, String streamExt, String streamName, long playStart, long playDuration, long playSeek, String vodTranscodeNGRP)</code>
SmoothStreamingLivePlaylist	<code>getLivePlaylist()</code>
void	<code>indexTranscoderVOD(IHTTPStreamerApplicationContext appContext, String vodTranscodeNGRP)</code>
void	<code>logLiveFragment(SmoothStreamerFragmentId fragmentId, PacketFragmentList fragmentData)</code>
void	<code>logVODFragment(SmoothStreamerFragmentId fragmentId, PacketFragmentList fragmentData)</code>
void	<code>notifyHTTPSessionCreate(IAplicationInstance appInstance, IHTTPStreamerSession httpStreamerSession)</code>
void	<code>notifyHTTPSessionDestroy(IAplicationInstance appInstance, IHTTPStreamerSession httpStreamerSession)</code>
void	<code>setLivePlaylist(SmoothStreamingLivePlaylist livePlaylist)</code>
void	<code>shutdown()</code>
void	<code>updateLoggingValues()</code>

Methods inherited from class com.wowza.wms.httpstreamer.model.HTTPStreamerSessionBase

```

acceptSession, addIOPerformance, addIOPerformance2, addStreamDomainStr,
addStreamDomainStrs, addUserHTTPHeaders, checkAndSetPlayLogged, clearLoggingValues,
containsStreamDomainStr, containsStreamNameParts, doSessionRedirect,
doSessionRedirect, extractHTTPRequestInfo, getAndClearNotifyCreate, getAppInstance,
getConnectionHolder, getCookieStr, getDvrSessionInfo, getElapsedTime, getFileInfo,
getHTTPDate, getHTTPHeader, getHTTPHeaderMap, getHTTPHeaderNames, getHTTPIntHeader,
getHTTPStreamerAdapter, getIOPerformanceCounter, getIpAddress, getLastRequest,
getLiveStreamingPacketizer, getLock, getPlayDuration, getPlaySeek, getPlayStart,
getProperties, getQueryStr, getRedirectSessionBody, getRedirectSessionCode,
getRedirectSessionContentType, getRedirectSessionURL, getReferrer, getServerIp,
getServerPort, getSessionId, getSessionProtocol, getSessionTimeout, getSessionType,
getStream, getStreamDomainStr, getStreamDomainStrList, getStreamExt, getStreamName,
getStreamNameParts, getStreamPosition, getTimeRunning, getTimeRunningSeconds,
getTranscoderVODIndex, getTranscoderVODIndex, getTranscoderVODIndexDestinationsMap,
getTranscoderVODIndexDestinationsOrder, getTranscoderVODSession, getUri,
getUserAgent, getUserHTTPHeaders, getUserQueryStr, getVHost, getVODTranscodeNGRP,
indexTranscoderVOD, isAcceptSession, isActive, isFileInfo, isHTTPOrigin,
isPlayLogged, isRedirectSession, isTimeout, isTimeoutSession, isvalidated,
isValidStreamDomainStr, lockRepeaterStreams, putFileInfo, putStreamNameParts,
redirectSession, redirectSession, rejectSession, removeStreamDomainStr,
setAcceptSession, setActive, setAppInstance, setCookieStr, setDvrSessionInfo,
setHTTPOrigin, setHTTPStreamerAdapter, setIpAddress, setLiveStreamingPacketizer,
setPlayDuration, setPlayLogged, setPlaySeek, setPlayStart, setQueryStr,
setRedirectSession, setRedirectSessionBody, setRedirectSessionCode,
setRedirectSessionContentType, setRedirectSessionURL, setReferrer, setServerIp,
setServerPort, sessionId, sessionProtocol, sessionTimeout, sessionType,
setStream, setStreamExt, setStreamName, setStreamPosition, setThreadContext,
setTimeoutSession, setTranscoderVODIndex, setTranscoderVODSession, setUri,
setUserAgent, setUserHTTPHeader, setUserQueryStr, setValidated, setVHost,
setVODTranscodeNGRP, shutdown, shutdownLocked, touch, updateLoggingValues,
validStreamDomainToString

```

**Methods inherited from class java.lang.Object**

```

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait,
wait

```

**Methods inherited from interface [com.wowza.wms.httpstreamer.model.IHTTPStreamerSession](#)**

```

acceptSession, addIOPerformance, addIOPerformance2, addStreamDomainStr,
addStreamDomainStrs, addUserHTTPHeaders, checkAndSetPlayLogged, clearLoggingValues,
containsStreamDomainStr, containsStreamNameParts, doSessionRedirect,
doSessionRedirect, extractHTTPRequestInfo, getAndClearNotifyCreate, getAppInstance,
getConnectionHolder, getCookieStr, getDvrSessionInfo, getElapsedTime, getFileInfo,
getHTTPHeader, getHTTPHeaderMap, getHTTPHeaderNames, getHTTPIntHeader,
getHTTPStreamerAdapter, getIOPerformanceCounter, getIpAddress, getLastRequest,
getLiveStreamingPacketizer, getLock, getPlayDuration, getPlaySeek, getPlayStart,
getProperties, getQueryStr, getRedirectSessionBody, getRedirectSessionCode,
getRedirectSessionContentType, getRedirectSessionURL, getReferrer, getServerIp,
getServerPort, getSessionId, getSessionProtocol, getSessionTimeout, getSessionType,
getStream, getStreamExt, getStreamName, getStreamNameParts, getStreamPosition,
getTimeRunning, getTimeRunningSeconds, getTranscoderVODIndex, getTranscoderVODIndex,
getTranscoderVODIndexDestinationsMap, getTranscoderVODIndexDestinationsOrder,
getTranscoderVODSession, getUri, getUserAgent, getUserHTTPHeaders, getUserQueryStr,
getVHost, getVODTranscodeNGRP, indexTranscoderVOD, isAcceptSession, isActive,
isFileInfo, isHTTPOrigin, isPlayLogged, isRedirectSession, isTimeout,
isTimeoutSession, isvalidated, isValidStreamDomainStr, lockRepeaterStreams,
putFileInfo, putStreamNameParts, redirectSession, redirectSession, rejectSession,
removeStreamDomainStr, setAcceptSession, setActive, setAppInstance, setCookieStr,
setDvrSessionInfo, setHTTPOrigin, setHTTPStreamerAdapter, setIpAddress,
setLiveStreamingPacketizer, setPlayDuration, setPlayLogged, setPlaySeek,
setPlayStart, setQueryStr, setRedirectSession, setRedirectSessionBody,
setRedirectSessionCode, setRedirectSessionContentType, setRedirectSessionURL,
setReferrer, setServerIp, setServerPort, sessionId, sessionProtocol,
getSessionTimeout, getSessionType, setStream, setStreamExt, setStreamName,
setStreamPosition, setTimeoutSession, setTranscoderVODIndex, setTranscoderVODSession,
setUri, setUserAgent, setUserHTTPHeader, setUserQueryStr, setVHost,
setVODTranscodeNGRP, shutdown, touch, updateLoggingValues, validStreamDomainToString

```

Methods inherited from interface [com.wowza.wms.httpstreamer.model.IHTTPSessionNotify](#)

[notifyHTTPSessionCreate](#), [notifyHTTPSessionDestroy](#)

## Constructors

### HTTPStreamerSessionSmoothStreamer

```
public HTTPStreamerSessionSmoothStreamer()
```

## Methods

### shutdown

```
public void shutdown()
```

### containsIndex

```
public boolean containsIndex(String streamName)
```

## getIndex

```
public IHTTPStreamerSmoothStreamerIndex getIndex(IHTTPStreamerSession  
httpStreamerSession,  
    IHTTPStreamerApplicationContext appContext,  
    String streamExt,  
    String streamName,  
    long playStart,  
    long playDuration,  
    long playSeek,  
    String vodTranscodeNGRP)
```

---

## updateLoggingValues

```
public void updateLoggingValues()
```

---

## clearLoggingValues

```
public void clearLoggingValues()
```

---

## getLivePlaylist

```
public SmoothStreamingLivePlaylist getLivePlaylist()
```

---

## setLivePlaylist

```
public void setLivePlaylist(SmoothStreamingLivePlaylist livePlaylist)
```

---

## logLiveFragment

```
public void logLiveFragment(SmoothStreamerFragmentId fragmentId,  
    PacketFragmentList fragmentData)
```

---

## logVODFragment

```
public void logVODFragment(SmoothStreamerFragmentId fragmentId,  
    PacketFragmentList fragmentData)
```

---

## notifyHTTPSessionCreate

```
public void notifyHTTPSessionCreate(IApplicationInstance appInstance,  
    IHTTPStreamerSession httpStreamerSession)
```

## notifyHTTPSessionDestroy

```
public void notifyHTTPSessionDestroy(IApplicationInstance appInstance,  
IHTTPStreamerSession httpStreamerSession)
```

---

## indexTranscoderVOD

```
public void indexTranscoderVOD(IHTTPStreamerApplicationContext appContext,  
String vodTranscodeNGRP)
```

---

Package

**com.wowza.wms.httpstreamer.webmstreaming.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, notifyCreate, playDuration, playSeek,
playStart, properties, queryStr, redirectSession, redirectSessionBody, redirectSessionCode,
redirectSessionContentType, redirectSessionURL, referrer, serverIp, serverPort, sessionId,
sessionProtocol, sessionTimeout, sessionType, stream, streamDomainStrSet, streamExt,
streamName, streamNamePartMap, streamPosition, timeoutSession, totalIOPerformance2Last,
totalIOPerformanceLast, transcoderVODIndex, transcoderVODIndexDestinationsMap,
transcoderVODIndexDestinationsOrder, transcoderVODSession, uri, userAgent, userHTTPHeaders,
userQueryStr, vhost, vodTranscodeNGRP
```

### Fields inherited from interface com.wowza.wms.httpstreamer.model.IHTTPStreamerSession

```
SESSIONPROTOCOL_COUNT, SESSIONPROTOCOL_CUPERTINOSTREAMING,
SESSIONPROTOCOL_DIRECTSTREAMING, SESSIONPROTOCOL_DVRCHUNKSTREAMING,
SESSIONPROTOCOL_MPEGDASHSTREAMING, SESSIONPROTOCOL_SANJOSESTREAMING,
SESSIONPROTOCOL_SMOOTHSTREAMING, SESSIONPROTOCOL_UNKNOWN, SESSIONPROTOCOL_WEBMSTREAMING,
SESSIONTYPE_LIVE, SESSIONTYPE_LIVEDVR, SESSIONTYPE_UNKNOWN, SESSIONTYPE_VOD
```

## Constructor Summary

public	<a href="#">HTTPStreamerSessionWebM()</a>
--------	---

## Method Summary

WebMPacketizerSession Tracker	<a href="#">getSessionTracker()</a>
void	<a href="#">setSessionTracker(WebMPacketizerSessionTracker sessionTracker)</a>
void	<a href="#">shutdown()</a>

### Methods inherited from class com.wowza.wms.httpstreamer.model.HTTPStreamerSessionBase

```

acceptSession, addIOPerformance, addIOPerformance2, addStreamDomainStr,
addStreamDomainStrs, addUserHTTPHeaders, checkAndSetPlayLogged, clearLoggingValues,
containsStreamDomainStr, containsStreamNameParts, doSessionRedirect,
doSessionRedirect, extractHTTPRequestInfo, getAndClearNotifyCreate, getAppInstance,
getConnectionHolder, getCookieStr, getDvrSessionInfo, getElapsedTime, getFileInfo,
getHTTPDate, getHTTPHeader, getHTTPHeaderMap, getHTTPHeaderNames, getHTTPIntHeader,
getHTTPStreamerAdapter, getIOPerformanceCounter, getIpAddress, getLastRequest,
getLiveStreamingPacketizer, getLock, getPlayDuration, getPlaySeek, getPlayStart,
getProperties, getQueryStr, getRedirectSessionBody, getRedirectSessionCode,
getRedirectSessionContentType, getRedirectSessionURL, getReferrer, getServerIp,
getServerPort, getSessionId, getSessionProtocol, getSessionTimeout, getSessionType,
getStream, getStreamDomainStr, getStreamDomainStrList, getStreamExt, getStreamName,
getStreamNameParts, getStreamPosition, getTimeRunning, getTimeRunningSeconds,
getTranscoderVODIndex, getTranscoderVODIndex, getTranscoderVODIndexDestinationsMap,
getTranscoderVODIndexDestinationsOrder, getTranscoderVODSession, getUri,
getUserAgent, getUserHTTPHeaders, getUserQueryStr, getVHost, getVODTranscodeNGRP,
indexTranscoderVOD, isAcceptSession, isActive, isFileInfo, isHTTPOrigin,
isPlayLogged, isRedirectSession, isTimeout, isTimeoutSession, isvalidated,
isValidStreamDomainStr, lockRepeaterStreams, putFileInfo, putStreamNameParts,
redirectSession, redirectSession, rejectSession, removeStreamDomainStr,
setAcceptSession, setActive, setAppInstance, setCookieStr, setDvrSessionInfo,
setHTTPOrigin, setHTTPStreamerAdapter, setIpAddress, setLiveStreamingPacketizer,
setPlayDuration, setPlayLogged, setPlaySeek, setPlayStart, setQueryStr,
setRedirectSession, setRedirectSessionBody, setRedirectSessionCode,
setRedirectSessionContentType, setRedirectSessionURL, setReferrer, setServerIp,
setServerPort, sessionId, sessionProtocol, sessionTimeout, sessionType,
setStream, setStreamExt, setStreamName, setStreamPosition, setThreadContext,
setTimeoutSession, setTranscoderVODIndex, setTranscoderVODSession, setUri,
setUserAgent, setUserHTTPHeader, setUserQueryStr, setValidated, setVHost,
setVODTranscodeNGRP, shutdown, shutdownLocked, touch, updateLoggingValues,
validStreamDomainToString

```

**Methods inherited from class java.lang.Object**

```

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait,
wait

```

**Methods inherited from interface [com.wowza.wms.httpstreamer.model.IHTTPStreamerSession](#)**

```

acceptSession, addIOPerformance, addIOPerformance2, addStreamDomainStr,
addStreamDomainStrs, addUserHTTPHeaders, checkAndSetPlayLogged, clearLoggingValues,
containsStreamDomainStr, containsStreamNameParts, doSessionRedirect,
doSessionRedirect, extractHTTPRequestInfo, getAndClearNotifyCreate, getAppInstance,
getConnectionHolder, getCookieStr, getDvrSessionInfo, getElapsedTime, getFileInfo,
getHTTPHeader, getHTTPHeaderMap, getHTTPHeaderNames, getHTTPIntHeader,
getHTTPStreamerAdapter, getIOPerformanceCounter, getIpAddress, getLastRequest,
getLiveStreamingPacketizer, getLock, getPlayDuration, getPlaySeek, getPlayStart,
getProperties, getQueryStr, getRedirectSessionBody, getRedirectSessionCode,
getRedirectSessionContentType, getRedirectSessionURL, getReferrer, getServerIp,
getServerPort, getSessionId, getSessionProtocol, getSessionTimeout, getSessionType,
getStream, getStreamExt, getStreamName, getStreamNameParts, getStreamPosition,
getTimeRunning, getTimeRunningSeconds, getTranscoderVODIndex, getTranscoderVODIndex,
getTranscoderVODIndexDestinationsMap, getTranscoderVODIndexDestinationsOrder,
getTranscoderVODSession, getUri, getUserAgent, getUserHTTPHeaders, getUserQueryStr,
getVHost, getVODTranscodeNGRP, indexTranscoderVOD, isAcceptSession, isActive,
isFileInfo, isHTTPOrigin, isPlayLogged, isRedirectSession, isTimeout,
isTimeoutSession, isvalidated, isValidStreamDomainStr, lockRepeaterStreams,
putFileInfo, putStreamNameParts, redirectSession, redirectSession, rejectSession,
removeStreamDomainStr, setAcceptSession, setActive, setAppInstance, setCookieStr,
setDvrSessionInfo, setHTTPOrigin, setHTTPStreamerAdapter, setIpAddress,
setLiveStreamingPacketizer, setPlayDuration, setPlayLogged, setPlaySeek,
setPlayStart, setQueryStr, setRedirectSession, setRedirectSessionBody,
setRedirectSessionCode, setRedirectSessionContentType, setRedirectSessionURL,
setReferrer, setServerIp, setServerPort, sessionId, sessionProtocol,
getSessionTimeout, getSessionType, setStream, setStreamExt, setStreamName,
setStreamPosition, setTimeoutSession, setTranscoderVODIndex, setTranscoderVODSession,
setUri, setUserAgent, setUserHTTPHeader, setUserQueryStr, setVHost,
setVODTranscodeNGRP, shutdown, touch, updateLoggingValues, validStreamDomainToString

```

## Constructors

### HTTPStreamerSessionWebM

```
public HTTPStreamerSessionWebM()
```

## Methods

### getSessionTracker

```
public WebMPacketizerSessionTracker getSessionTracker()
```

### setSessionTracker

```
public void setSessionTracker(WebMPacketizerSessionTracker sessionTracker)
```

### shutdown

```
public void shutdown()
```

(continued from last page)

---

Package

**com.wowza.wms.livestreamrecord.manager**

## com.wowza.wms.livestreamrecord.manager Interface ILiveStreamRecordManager

public interface **ILiveStreamRecordManager**  
extends

This interface provides access to the functionality of the LiveStreamRecordManager. The LiveStreamRecordManger is responsible for maintaining the list of IStreamRecorders as well as managing the IstreamRecorder life cycle.

### Method Summary

<a href="#">ILiveStreamRecordManagerActionNotify</a>	<a href="#"><code>addListener(ILiveStreamRecordManagerActionNotify listener)</code></a> Used to add a listener to receive ILiveStreamRecordManagerActionNotify events from the LiveStreamRecordManager.
<a href="#">IStreamRecorder</a>	<a href="#"><code>getRecorder(IApplicationInstance appInstance, String streamName)</code></a> Used to get a specific IStreamRecorder from the LiveStreamRecordManager. Modifying the returned IStreamRecorder while a recording is active may result in undefined behavior.
<a href="#">java.util.List</a>	<a href="#"><code>getRecordersList(IApplicationInstance appInstance)</code></a> Used to get the list of IStreamRecorders for a specific application instance.
<a href="#">java.util.Map</a>	<a href="#"><code>getRecordersMap(IApplicationInstance appInstance)</code></a> Used to get the Map of stream Name to IStreamRecorder for a specific application instance.
<a href="#">void</a>	<a href="#"><code>init(IVHost vhost)</code></a> Called by vhost init to intialize the LiveStreamRecordManager
<a href="#">boolean</a>	<a href="#"><code>removeListener(ILiveStreamRecordManagerActionNotify listener)</code></a> Used to remove an ILiveStreamRecordManagerActionNotify listener from the LiveStreamRecordManager.
<a href="#">void</a>	<a href="#"><code>splitRecording(IApplicationInstance appInstance, String streamName)</code></a> Used to tell the LiveStreamRecordManager to split (a.k.a segment) a recording immediately for a specific stream in a specific application instance.
<a href="#">void</a>	<a href="#"><code>startRecording(IApplicationInstance appInstance, StreamRecorderParameters params)</code></a> Used to tell the LiveStreamRecordManager to create StreamRecorders for all streams which are published to the specified application instance.
<a href="#">void</a>	<a href="#"><code>startRecording(IApplicationInstance appInstance, String streamName, StreamRecorderParameters params)</code></a> Used to tell the LiveStreamRecordManager to create a StreamRecorder for a specific stream in a specific application instance.
<a href="#">void</a>	<a href="#"><code>stopRecording(IApplicationInstance appInstance)</code></a> Used to tell the LiveStreamRecordManager to stop all StreamRecorders on a specific application instance.
<a href="#">void</a>	<a href="#"><code>stopRecording(IApplicationInstance appInstance, String streamName)</code></a> Used to tell the LiveStreamRecordManager to stop a StreamRecorder for a specific stream in a specific application instance.

### Methods

(continued from last page)

## init

```
public void init(IVHost vhost)
```

Called by vhost init to intialize the LiveStreamRecordManager

**Parameters:**

vhost - The vhost to associated with the instance of LiveStreamRecordManager

---

## getRecorder

```
public IStreamRecorder getRecorder(IAplicationInstance appInstance,  
String streamName)
```

Used to get a specific IStreamRecorder from the LiveStreamRecordManager. Modifying the returned IStreamRecorder while a recording is active may result in undefined behavior.

**Parameters:**

appInstance - The application instance associated with the desired IStreamRecorder  
streamName - The stream name associated with the desired IStreamRecorder

**Returns:**

The requested IStreamRecorder or null if it does not exist

---

## getRecordersList

```
public java.util.List getRecordersList(IAplicationInstance appInstance)
```

Used to get the list of IStreamRecorders for a specific application instance. Making changes to the returned list (e.g. removing an IStreamRecorder from the list) does not affect the LiveStreamRecordManager's list, however, changes made to the IStreamRecorders themselves will be reflected in the LiveStreamRecordManager's IStreamRecorder. Therefore, changing IStreamRecorders while a recording is active may result in undefined behavior.

**Parameters:**

appInstance - The application instance associated with the desired IStreamRecorder List

**Returns:**

The requested IStreamRecorder List. The list can be empty if no IStreamRecorders are set.

---

## getRecordersMap

```
public java.util.Map getRecordersMap(IAplicationInstance appInstance)
```

Used to get the Map of stream Name to IStreamRecorder for a specific application instance. Making changes to the returned map (e.g. removing a mapping entry from the map) does not affect the LiveStreamRecordManager's map, however, changes made to the IStreamRecorders themselves will be reflected in the LiveStreamRecordManager's IStreamRecorder. Therefore, changing IStreamRecorders while a recording is active may result in undefined behavior.

**Parameters:**

appInstance - The application instance associated with the desired IStreamRecorder List

**Returns:**

The requested stream name to IStreamRecorder mappings. The map can be empty if no IStreamRecorders are set.

---

## startRecording

```
public void startRecording(IAplicationInstance appInstance,  
StreamRecorderParameters params)
```

(continued from last page)

Used to tell the LiveStreamRecordManager to create StreamRecorders for all streams which are published to the specified application instance. The StreamRecorderParameters specified will be used for each StreamRecorder created. If the StreamRecordParameters are null the default StreamRecordParameters will be used.

**Parameters:**

appInstance - The application instance on which to record all incoming streams  
params - The StreamRecorderParameters to use for all recorders

**See Also:**

[StreamRecorderParameters](#)  
[IStreamRecorderConstants](#)

---

## startRecording

```
public void startRecording(IAplicationInstance appInstance,  
                           String streamName,  
                           StreamRecorderParameters params)
```

Used to tell the LiveStreamRecordManager to create a StreamRecorder for a specific stream in a specific application instance. The StreamRecorderParameters specified will be used for the StreamRecorder. If the StreamRecordParameters are null the default StreamRecordParameters will be used.

**Parameters:**

appInstance - The application instance on which to record the requested stream  
streamName - The stream name associated with the incoming stream to record  
params - The StreamRecorderParameters to use for the StreamRecorder

**See Also:**

[StreamRecorderParameters](#)  
[IStreamRecorderConstants](#)

---

## stopRecording

```
public void stopRecording(IAplicationInstance appInstance,  
                           String streamName)
```

Used to tell the LiveStreamRecordManager to stop a StreamRecorder for a specific stream in a specific application instance. If a StreamRecorder already exists for the specified application instance and stream name, nothing is done.

**Parameters:**

appInstance - The application instance of the StreamRecorder to stop  
streamName - The stream name associated with the StreamRecorder to stop

---

## stopRecording

```
public void stopRecording(IAplicationInstance appInstance)
```

Used to tell the LiveStreamRecordManager to stop all StreamRecorders on a specific application instance. If no StreamRecorders exist for the specified application instance, nothing is done.

**Parameters:**

appInstance - The application instance on which to stop all StreamRecorders

---

## splitRecording

```
public void splitRecording(IAplicationInstance appInstance,  
                           String streamName)
```

Used to tell the LiveStreamRecordManager to split (a.k.a segment) a recording immediately for a specific stream in a specific application instance. If no StreamRecorder exists for the specified application instance and stream name, nothing is done.

---

(continued from last page)

**Parameters:**

- appInstance - The application instance of the StreamRecorder to split
  - streamName - The stream name associated with the StreamRecorder to split
- 

## addListener

```
public ILiveStreamRecordManagerActionNotify  
addListener(ILiveStreamRecordManagerActionNotify listener)
```

Used to add a listener to receive ILiveStreamRecordManagerActionNotify events from the LiveStreamRecordManager. Listeners are notified in the order in which they are added.

**Parameters:**

- listener - An instance of a class which implements the ILiveStreamRecordManagerActionNotify interface

**Returns:**

The listener instance passed in

**See Also:**

[ILiveStreamRecordManagerActionNotify](#)

---

## removeListener

```
public boolean removeListener(ILiveStreamRecordManagerActionNotify listener)
```

Used to remove an ILiveStreamRecordManagerActionNotify listener from the LiveStreamRecordManager.

**Parameters:**

- listener - The listener passed into the addListener method.

**Returns:**

true if the listener was found in the list otherwise false.

## com.wowza.wms.livestreamrecord.manager Interface ILiveStreamRecordManagerActionNotify

public interface **ILiveStreamRecordManagerActionNotify**  
extends

This interface provides notifications to registered listeners for LiveStreamRecordManager events as it operates on IStreamRecorders. These notification are sent prior to invoking the associated method for a recorder.

Multiple Listener's can be registered for the LiveStreamRecordManager. The listeners are notified in the order in which they were added.

### See Also:

[for API's to add and remove listeners](#)

## Method Summary

void	<a href="#">onCreateRecord(IStreamRecorder recorder)</a> This notification occurs when a new recorder is created due to a call to one of the ILiveStreamRecordManager startRecording API's.
void	<a href="#">onSplitRecord(IStreamRecorder recorder)</a> This notification occurs due to a call to the ILiveStreamRecordManager splitRecording API.
void	<a href="#">onStartRecord(IStreamRecorder recorder)</a> This notification occurs due to a call to one of the ILiveStreamRecordManager startRecording API's.
void	<a href="#">onStopRecord(IStreamRecorder recorder)</a> This notification occurs due to a call to the ILiveStreamRecordManager stopRecording API.
void	<a href="#">onSwitchRecord(IStreamRecorder recorder, IMediaStream newStream)</a> This notification occurs when an existing stream is replaced with a stream of the same name.
<a href="#">IStreamRecorder</a>	<a href="#">recordFactory(String streamName, StreamRecorderParameters recordParams)</a> This notification is the first one to occur when the ILiveStreamRecordManager is requested to start a recording via any of the ILiveStreamRecordManager startRecording API's. It gives the listener the opportunity to return a custom class which implements the IStreamRecorder interface to handle the recording. If the listener returns null, the LiveStreamRecordManager will internally create an new StreamRecorder to handle the recording.

## Methods

### recordFactory

```
public IStreamRecorder recordFactory\(String streamName, StreamRecorderParameters recordParams\)
```

(continued from last page)

This notification is the first one to occur when the ILiveStreamRecordManager is requested to start a recording via any of the ILiveStreamRecordManager startRecording API's

It gives the listener the opportunity to return a custom class which implements the IStreamRecorder interface to handle the recording.

If the listener returns null, the LiveStreamRecordManager will internally create an new StreamRecorder to handle the recording. Since multiple listeners can be registered, LiveStreamRecordManager notifies listeners in the order in which they were added, and will use the first IStreamRecorder returned which is not null, however, remaining listeners will receive the notification.

Called before LiveStreamRecordManager internal action is taken

---

**Parameters:**

recordParams - the parameters passed into IStreamRecordManager startRecording API  
streamName - the stream name passed into IStreamRecordManager startRecording API

**Returns:**

the IStreamRecorder which will handled recording or null

---

## onCreateRecord

```
public void onCreateRecord(IStreamRecorder recorder)
```

This notification occurs when a new recorder is created due to a call to one of the ILiveStreamRecordManager startRecording API's.

Called after the recordFactory notification is called, but before LiveStreamRecordManager calls the recorders own createRecorder method.

---

**Parameters:**

recorder - The new recorder

---

## onStartRecord

```
public void onStartRecord(IStreamRecorder recorder)
```

This notification occurs due to a call to one of the ILiveStreamRecordManager startRecording API's.

Called after the onCreateRecord notification is called, but before LiveStreamRecordManager calls the recorders own startRecorder method.

---

**Parameters:**

recorder - The recorder which is starting

---

## onSplitRecord

```
public void onSplitRecord(IStreamRecorder recorder)
```

This notification occurs due to a call to the ILiveStreamRecordManager splitRecording API.

Called before LiveStreamRecordManager calls the recorders own splitRecorder method.

---

**Parameters:**

recorder - The recorder for which the split has been requested.

## onStopRecord

```
public void onStopRecord(IStreamRecorder recorder)
```

This notification occurs due to a call to the ILiveStreamRecordManager stopRecording API.

Called before LiveStreamRecordManager calls the recorders own stopRecorder method.

### Parameters:

recorder - The recorder for which a stop has been requested.

---

## onSwitchRecord

```
public void onSwitchRecord(IStreamRecorder recorder,  
    IMediaStream newStream)
```

This notification occurs when an existing stream is replaced with a stream of the same name. This can occur when a stream is published from an encoder, and then a different encoder publishes a stream using the same stream name.

Called before LiveStreamRecordManager calls the recorders own switchRecorder method.

### Parameters:

recorder - The recorder for the existing stream

newStream - The new IMediaStream associated with the existing stream name

## com.wowza.wms.livestreamrecord.manager Interface IStreamRecorder

public interface **IStreamRecorder**  
extends

This interface provides access to into a StreamRecorder.

The StreamRecorder is responsible for performing the actual recording functionality of a live stream, as directed by the LiveStreamRecordManager.

### Method Summary

<a href="#">IStreamRecorderActionNotify</a>	<a href="#">addListener(IStreamRecorderActionNotify listener)</a> Called to add a listener for IStreamRecorderActionNotify events generated by the recorder. Multiple listeners can be added.
void	<a href="#">createRecorder()</a> Called to tell the recorder to initialize itself after it has been created.
<a href="#">IApplicationInstance</a>	<a href="#">getAppInstance()</a> Called to get the application Instance the recorder is associated with
String	<a href="#">getBaseFilePath()</a> Called to get the full path and filename used by the recorder as the PRIOR to any changes made due to the selected segmentationType and versioningOption.
long	<a href="#">getCurrentDuration()</a> Called to get the current segment size regardless of segmentationType. This value is updated continuously while the recorder state is RECORDER_STATE_RECORDING.
String	<a href="#">getCurrentFile()</a> Called to get the current segment size regardless of segmentationType. This value is updated continuously while the recorder state is RECORDER_STATE_RECORDING.
long	<a href="#">getCurrentSize()</a> Called to get the current segment size regardless of segmentationType. This value is updated continuously while the recorder state is RECORDER_STATE_RECORDING.
org.joda.time.DateTime	<a href="#">getEndTime()</a> Called to get the end time of the segment regardless of segmentationType. When the recording is still active, the returned value is for the previous segment. When the recording has been stopped, the returned value is for the last segment.
String	<a href="#">getErrorString()</a> Called to get an error string describing the error, when the recorder state is RECORDER_STATE_ERROR.
String	<a href="#">getFileFormat()</a> Called to get output file format being used by the recorder.
String	<a href="#">getFilePath()</a> Called to get the full path and filename used by the recorder AFTER changes made due to the selected segmentationType and versioningOption. NOTE: When the recorder is not in RECORDER_STATE_RECORDING state, the returned value will be the basePath value.
<a href="#">IStreamRecorderFileVersionDelegate</a>	<a href="#">getFileVersionDelegate()</a> Called to get the IStreamRecorderFileVersionDelegate being used by the recorder.

<a href="#"><u>StreamRecorderParameters</u></a>	<a href="#"><u>getRecorderParams()</u></a> Called to get the StreamRecorderParameters for the recorder. This is typically used to check or set the StreamRecorderParameters used to configure a recorder.
int	<a href="#"><u>getRecorderState()</u></a> Called to get the current state of the recorder
String	<a href="#"><u>getRecorderStateString()</u></a> Called to get the current state of the recorder as a text string.
String	<a href="#"><u>getSegmentationType()</u></a> Called to get the SegmentationType being used by the recorder.
long	<a href="#"><u>getSegmentDuration()</u></a> Called to get the segment duration used by the recorder when segmentationType is set to SEGMENT_BY_DURATION.
int	<a href="#"><u>getSegmentNumber()</u></a> Called to get the current segment number regardless of segmentationType. This value is updated continuously while the recorder state is RECORDER_STATE_RECORDING.
<a href="#"><u>CrontabEvent</u></a>	<a href="#"><u>getSegmentSchedule()</u></a> Called to get the CrontabEvent object associated with the recorder when the segmentationType is set to SEGMENT_BY_SCHEDULE. When the recorder state is RECORDER_STATE_ERROR, RECORDER_STATE_PENDING or RECORDER_STATE_WAITING, the return value is null, otherwise the CrontabEvent used by the recorder is returned.
String	<a href="#"><u>getSegmentScheduleString()</u></a> Called to get the segment schedule crontab string used by the recorder when segmentationType is set to SEGMENT_BY_SCHEDULE.
long	<a href="#"><u>getSegmentSize()</u></a> Called to get the segment size used by the recorder when segmentationType is set to SEGMENT_BY_SIZE.
org.joda.time.DateTime	<a href="#"><u>getStartTime()</u></a> Called to get the start time of the current segment regardless of segmentationType.
<a href="#"><u>IMediaStream</u></a>	<a href="#"><u>getStream()</u></a> Called to get the IMediaStream object being used by the recorder
String	<a href="#"><u>getStreamName()</u></a> Called to get the stream name being used by the recorder
String	<a href="#"><u>getVersioningOption()</u></a> Called to get the method used by the recorder to version a file when a file exists of the same name.
boolean	<a href="#"><u>isAppendFile()</u></a> Called to check the versioningOption used by the recorder. Users should use getVersioningOption instead.
boolean	<a href="#"><u>isMoveFirstVideoFrameToZero()</u></a> Called to check the value of MoveFirstVideoFrameToZero being used by the recorder.
boolean	<a href="#"><u>isRecordData()</u></a> Called to check the value of RecordData being used by the recorder.
boolean	<a href="#"><u>isSplitOnTcDiscontinuity()</u></a> Called to check the value of splitOnTcDiscontinuity being used by the recorder.

boolean	<a href="#"><u>isStartOnKeyFrame()</u></a> Called to check the value of StartOnKeyFrame being used by the recorder.
boolean	<a href="#"><u>isVersionFile()</u></a> Called to check the versioningOption used by the recorder. Users should use getVersioningOption instead.
boolean	<a href="#"><u>removeListener(IStreamRecorderActionNotify listener)</u></a> Called to remove a listener from the the recorder.
void	<a href="#"><u>setInternalRecorder(com.wowza.wms.livestreamrecord.model.ILiveStreamRecord internalRecorder)</u></a> Called to set the internal recorder used by the StreamRecorder This method should only be used if existing recorder workflows cannot be implemented using the IStreamRecorderActionNotify API's.
void	<a href="#"><u>setStream(IMediaStream stream)</u></a> Called to set the IMediaStream object the recorder will use NOTE: This method is used by LiveStreamRecordManager to manage recorders and associated IMediaStream objects and should NOT be used directly when a StreamRecorder is under the control of LiveStreamRecordManager
void	<a href="#"><u>splitRecorder()</u></a> Called to tell the recorder to stop writing to the current file, create a new file and write begin writing data to the new file. The recorder should start writing data to the new file beginning with a video key frame.
void	<a href="#"><u>startRecorder()</u></a> Called to tell the recorder to start recording.
void	<a href="#"><u>stopRecorder()</u></a> Called to tell the recorder to stop recording.
void	<a href="#"><u>switchRecorder(IMediaStream newStream)</u></a> Called to tell the recorder to switch over to the data from a new IMediaStream. This occurs when an existing stream is replaced with a stream of the same name.

## Methods

### createRecorder

```
public void createRecorder()
```

Called to tell the recorder to initalize itself after it has been created. This API should be called prior to calling the startRecorder() method.

### startRecorder

```
public void startRecorder()
```

Called to tell the recorder to start recording. This API should be called after createRecorder() method has been called to ensure the recorder has been properly initialized

### switchRecorder

```
public void switchRecorder(IMediaStream newStream)
```

---

(continued from last page)

Called to tell the recorder to switch over to the data from a new IMediaStream.  
This occurs when an existing stream is replaced with a stream of the same name.

For example: When a stream is published from an encoder, and then a different encoder publishes a stream using the same stream name. Called before LiveStreamRecordManager calls the recorders own switchRecorder method.

---

**Parameters:**

newStream - The new IMediaStream the recorder should switch to.

---

## stopRecorder

**public void stopRecorder()**

Called to tell the recorder to stop recording.

---

## splitRecorder

**public void splitRecorder()**

Called to tell the recorder to stop writing to the current file, create a new file and write begin writing data to the new file.  
The recorder should start writing data to the new file beginning with a video key frame.

---

## getRecorderParams

**public [StreamRecorderParameters](#) getRecorderParams()**

Called to get the StreamRecorderParameters for the recorder  
This is typically used to check or set the StreamRecorderParameters used to configure a recorder. To get the actual value used by the recorder, use the appropriate get method found below.

---

**See Also:**

[StreamRecorderParameters](#)

---

## getAppInstance

**public [IApplicationInstance](#) getAppInstance()**

Called to get the application Instance the recorder is associated with

---

**Returns:**

IApplicationInstance

---

## getStreamName

**public String getStreamName()**

Called to get the stream name being used by the recorder

---

**Returns:**

String Stream name

## getStream

```
public IMediaStream getStream()
```

Called to get the IMediaStream object being used by the recorder

### Returns:

IMediaStream Stream object

---

## setStream

```
public void setStream(IMediaStream stream)
```

Called to set the IMediaStream object the recorder will use

NOTE: This method is used by LiveStreamRecordManager to manage recorders and associated IMediaStream objects and should NOT be used directly when a StreamRecorder is under the control of LiveStreamRecordManager

### Parameters:

stream - IMediaStream The new Stream object

---

## getRecorderState

```
public int getRecorderState()
```

Called to get the current state of the recorder

### Returns:

int Current state

### See Also:

[IStreamRecorderConstants.RECORDER\\_STATE\\_ERROR](#)  
[IStreamRecorderConstants.RECORDER\\_STATE\\_PENDING](#)  
[IStreamRecorderConstants.RECORDER\\_STATE\\_WAITING](#)  
[IStreamRecorderConstants.RECORDER\\_STATE\\_RECORDING](#)

---

## getRecorderStateString

```
public String getRecorderStateString()
```

Called to get the current state of the recorder as a text string.

### Returns:

String Current state as a string

(continued from last page)

**See Also:**

[IStreamRecorderConstants.RECORDER\\_STATE\\_ERROR\\_STRING](#)  
[IStreamRecorderConstants.RECORDER\\_STATE\\_PENDING\\_STRING](#)  
[IStreamRecorderConstants.RECORDER\\_STATE\\_WAITING\\_STRING](#)  
[IStreamRecorderConstants.RECORDER\\_STATE\\_RECORDING\\_STRING](#)

---

## getFileFormat

```
public String getFileFormat()
```

Called to get output file format being used by the recorder.

**Returns:**

String Output file format

**See Also:**

[IStreamRecorderConstants.FORMAT\\_MP4](#)  
[IStreamRecorderConstants.FORMAT\\_FLV](#)

---

## isStartOnKeyFrame

```
public boolean isStartOnKeyFrame()
```

Called to check the value of StartOnKeyFrame being used by the recorder.

**Returns:**

boolean true, false

**See Also:**

[StreamRecorderParameters.startOnKeyFrame](#)

---

## isRecordData

```
public boolean isRecordData()
```

Called to check the value of RecordData being used by the recorder.

**Returns:**

boolean true, false

**See Also:**

[StreamRecorderParameters.recordData](#)

---

(continued from last page)

## isSplitOnTcDiscontinuity

```
public boolean isSplitOnTcDiscontinuity()
```

Called to check the value of splitOnTcDiscontinuity being used by the recorder.

**Returns:**

boolean true, false

**See Also:**

[StreamRecorderParameters.splitOnTcDiscontinuity](#)

## isMoveFirstVideoFrameToZero

```
public boolean isMoveFirstVideoFrameToZero()
```

Called to check the value of MoveFirstVideoFrameToZero being used by the recorder.

**Returns:**

boolean true, false

**See Also:**

[StreamRecorderParameters.moveFirstVideoFrameToZero](#)

## getSegmentationType

```
public String getSegmentationType()
```

Called to get the SegmentationType being used by the recorder.

**Returns:**

String Segmentation type

**See Also:**

[StreamRecorderParameters.segmentationType](#)  
[IStreamRecorderConstants.SEGMENT\\_NONE](#)  
[IStreamRecorderConstants.SEGMENT\\_BY\\_DURATION](#)  
[IStreamRecorderConstants.SEGMENT\\_BY\\_SIZE](#)  
[IStreamRecorderConstants.SEGMENT\\_BY\\_SCHEDULE](#)

## getVersioningOption

```
public String getVersioningOption()
```

(continued from last page)

Called to get the method used by the recorder to version a file when a file exists of the same name.

**Returns:**

String Versioning option

**See Also:**

[StreamRecorderParameters.versioningOption](#)  
[IStreamRecorderConstants.APPEND\\_FILE](#)  
[IStreamRecorderConstants.OVERWRITE\\_FILE](#)  
[IStreamRecorderConstants.VERSION\\_FILE](#)

---

**getSegmentSize**

public long **getSegmentSize()**

Called to get the segment size used by the recorder when segmentationType is set to SEGMENT\_BY\_SIZE.

**Returns:**

long Segment size in bytes

**See Also:**

[StreamRecorderParameters.segmentSize](#)

---

**getSegmentDuration**

public long **getSegmentDuration()**

Called to get the segment duration used by the recorder when segmentationType is set to SEGMENT\_BY\_DURATION.

**Returns:**

long Segment duration in milliseconds

**See Also:**

[StreamRecorderParameters.segmentDuration](#)

---

**getSegmentScheduleString**

public String **getSegmentScheduleString()**

Called to get the segment schedule crontab string used by the recorder when segmentationType is set to SEGMENT\_BY\_SCHEDULE.

**Returns:**

(continued from last page)

String Segment schedule

**See Also:**

[StreamRecorderParameters.segmentSchedule](#)

---

## getFilePath

**public String `getFilePath()`**

Called to get the full path and filename used by the recorder AFTER changes made due to the selected segmentationType and versioningOption.

NOTE: When the recorder is not in RECORDER\_STATE\_RECORDING state, the returned value will be the basePath value.

**Returns:**

String Full path and filename

**See Also:**

[getBaseFilePath\(\)](#)

---

## getBaseFilePath

**public String `getBaseFilePath()`**

Called to get the full path and filename used by the recorder as the PRIOR to any changes made due to the selected segmentationType and versioningOption.

**Returns:**

String Full path and filename

**See Also:**

[getFilePath\(\)](#)

---

## getCurrentSize

**public long `getCurrentSize()`**

Called to get the current segment size regardless of segmentationType

This value is updated continuously while the recorder state is RECORDER\_STATE\_RECORDING.

**Returns:**

long Segment size in bytes

---

## getCurrentDuration

**public long `getCurrentDuration()`**

---

(continued from last page)

Called to get the current segment size regardless of segmentationType.  
This value is updated continuously while the recorder state is RECORDER\_STATE\_RECORDING.

**Returns:**

long Segment duration in milliseconds

---

## getCurrentFile

```
public String getCurrentFile()
```

Called to get the current segment size regardless of segmentationType.  
This value is updated continuously while the recorder state is RECORDER\_STATE\_RECORDING.

**Returns:**

String The current fully qualified file name, packets are being written to.

---

## getSegmentNumber

```
public int getSegmentNumber()
```

Called to get the current segment number regardless of segmentationType.  
This value is updated continuously while the recorder state is RECORDER\_STATE\_RECORDING.

**Returns:**

int Current segment number.

---

## getStartTime

```
public org.joda.time.DateTime getStartTime()
```

Called to get the start time of the current segment regardless of segmentationType.

**Returns:**

DateTime Current segment's start time.

---

## getEndTime

```
public org.joda.time.DateTime getEndTime()
```

Called to get the end time of the segment regardless of segmentationType.  
When the recording is still active, the returned value is for the previous segment.  
When the recording has been stopped, the returned value is for the last segment.

**Returns:**

DateTime Segment end time.

---

## getErrorString

```
public String getErrorString()
```

---

(continued from last page)

Called to get an error string describing the error, when the recorder state is is RECORDER\_STATE\_ERROR.

**Returns:**

String Error description string

---

## getFileVersionDelegate

```
public IStreamRecorderFileVersionDelegate getFileVersionDelegate()
```

Called to get the IStreamRecorderFileVersionDelegate being used by the recorder.

**Returns:**

IStreamRecorderFileVersionDelegate

**See Also:**

[IStreamRecorderFileVersionDelegate](#)  
[StreamRecorderFileVersionDelegate](#)  
[StreamRecorderSimpleFileVersionDelegate](#)

---

## addListener

```
public IStreamRecorderActionNotify addListener(IStreamRecorderActionNotify listener)
```

Called to add a listener for IStreamRecorderActionNotify events generated by the recorder.

Multiple listeners can be added. All listeners will be called for each event, in the order in which they were added.

**Parameters:**

listener - An object which implements the IStreamRecorderActionNotify interface

**Returns:**

IStreamRecorderActionNotify The listener object passed in.

**See Also:**

[IStreamRecorderActionNotify](#)

---

## removeListener

```
public boolean removeListener(IStreamRecorderActionNotify listener)
```

Called to remove a listener from the the recorder.

**Parameters:**

listener - The listener which is to be removed

**Returns:**

(continued from last page)

boolean true, if listener was found in list, else false

---

## isVersionFile

```
public boolean isVersionFile()
```

Called to check the versioningOption used by the recorder.  
Users should use getVersioningOption instead.

**Returns:**

boolean true when versioningOption is set to VERISON\_FILE, otherwise false

---

## isAppendFile

```
public boolean isAppendFile()
```

Called to check the versioningOption used by the recorder.  
Users should use getVersioningOption instead.

**Returns:**

boolean true when versioningOption is set to APPEND\_FILE, otherwise false

---

## getSegmentSchedule

```
public CrontabEvent getSegmentSchedule()
```

Called to get the CrontabEvent object associated with the recorder when the segmentationType is set to SEGMENT\_BY\_SCHEDULE  
When the recorder state is RECORDER\_STATE\_ERROR, RECORDER\_STATE\_PENDING or RECORDER\_STATE\_WAITING, the return value is null,  
otherwise the CrontabEvent use by the recorder is returned.

**Returns:**

CrontabEvent

---

## setInternalRecorder

```
public void setInternalRecorder(com.wowza.wms.livestreamrecord.model.ILiveStreamRecord  
internalRecorder)
```

Called to set the internal recorder used by the StreamRecorder  
This method should only be used if existing recorder workflows cannot be implemented using the IStreamRecorderActionNotify API's.

**Parameters:**

internalRecorder

## com.wowza.wms.livestreamrecord.manager Interface IStreamRecorderActionNotify

public interface **IStreamRecorderActionNotify**  
extends

This interface provides notifications to registered listeners throughout the life cycle of a StreamRecorder. These events are similar to the ILiveStreamRecordManagerActionNotify events, however, they are called by the StreamRecorder deeper in the action processing.

Multiple Listener's can be registered for each StreamRecorder. The listeners are notified in the order in which they were added.

### See Also:

[for API's to add and remove listeners](#)

## Method Summary

void	<a href="#">onCreateRecorder(IStreamRecorder recorder)</a> This notification occurs when a new recorder has been created, but before recording has started.
void	<a href="#">onSegmentEnd(IStreamRecorder recorder)</a> This notification occurs when a recorder stops writing to the current file
void	<a href="#">onSegmentStart(IStreamRecorder recorder)</a> This notification occurs when a recorder starts writing to the current file
void	<a href="#">onSplitRecorder(IStreamRecorder recorder)</a> This notification occurs when a recorder is directed to split the active recording.
void	<a href="#">onStartRecorder(IStreamRecorder recorder)</a> This notification occurs when a recorder starts recording.
void	<a href="#">onStopRecorder(IStreamRecorder recorder)</a> This notification occurs when a recorder stops recording.
void	<a href="#">onSwitchRecorder(IStreamRecorder recorder, IMediaStream newStream)</a> This notification occurs when an existing stream has been replaced by a new stream of the same name.

## Methods

### onCreateRecorder

public void [onCreateRecorder\(IStreamRecorder recorder\)](#)

This notification occurs when a new recorder has been created, but before recording has started. Within this method, changes can be made to the StreamRecorder's StreamRecorderParameters, which will then take affect when the recorder is started.

Called before action is taken

#### Parameters:

recorder - the new recorder

(continued from last page)

## onStartRecorder

```
public void onStartRecorder(IStreamRecorder recorder)
```

This notification occurs when a recorder starts recording.

Called after all configuration of the recorder is complete, but before the recording actually starts.

**Parameters:**

recorder - the recorder which is starting

---

## onSplitRecorder

```
public void onSplitRecorder(IStreamRecorder recorder)
```

This notification occurs when a recorder is directed to split the active recording.

Called after all configuration of the recorder is complete, but before the recording actually starts.

**Parameters:**

recorder - the recorder which is starting

---

## onStopRecorder

```
public void onStopRecorder(IStreamRecorder recorder)
```

This notification occurs when a recorder stops recording.

NOTE: Recorders are stopped whenever a stream is unpublished, or when a stream is "reset".

Within this method, calling; getCurrentFile(), getCurrentDuration(), getCurrentSize(), segmentNumber(), getStartTime(), getEndTime() will return the values accumulated from the time the onStartRecorder notification was called, to the time this notification was called.

Called after the recorder is stopped.

**Parameters:**

recorder - the recorder which is starting

---

## onSwitchRecorder

```
public void onSwitchRecorder(IStreamRecorder recorder,  
    IMediaStream newStream)
```

This notification occurs when an existing stream has been replaced by a new stream of the same name. The StreamRecorder will stop the recording associated with the existing stream and start a new recording for the new stream, using the same parameters as those used to originally start the recording.

Called before any action is taken

**Parameters:**

recorder - the current recorder for existing stream

newStream - the new stream which will be recorded

---

## onSegmentStart

```
public void onSegmentStart(IStreamRecorder recorder)
```

This notification occurs when a recorder starts writing to the current file

Called after a the file has been opened for writing, but before any data is written to the file.

**Parameters:**

(continued from last page)

recorder - the recorder which is writing the segment

---

## onSegmentEnd

public void **onSegmentEnd**([IStreamRecorder](#) recorder)

This notification occurs when a recorder stops writing to the current file

Called after all packets have been written to the file and the file has been closed.

**Parameters:**

recorder - the recorder which is wrote the segment

## com.wowza.wms.livestreamrecord.manager Interface IStreamRecorderConstants

public interface IStreamRecorderConstants  
extends

Defines the static constants used by the StreamRecorder system

The Properties for StreamRecorders should be placed in application.xml within the StreamRecorder properties container

for example:

```
</StreamRecorder>
</Properties>
</Property>
</Name>streamRecorderDebugEnable</Name>
</Value>true</Value>
</Type>Boolean</Type>
</Property>
</Properties>
</StreamRecorder>
```

### Field Summary

public static final	<a href="#">APPEND_FILE</a> Appends new stream data to the existing file. Value: <b>append</b>
public static final	<a href="#">FORMAT_FLV</a> Value: <b>flv</b>
public static final	<a href="#">FORMAT_MP4</a> Value: <b>mp4</b>
public static final	<a href="#">OVERWRITE_FILE</a> Overwrites the existing file with the new file Value: <b>overwrite</b>
public static final	<a href="#">PROPERTY_DEBUG_ENABLE</a> Application Property "streamRecorderDebugEnable": for enabling additional log statements throughout StreamRecorder Value: <b>streamRecorderDebugEnable</b>
public static final	<a href="#">PROPERTY_FILE_FORMAT</a> Application Property "streamRecorderFileFormat": for specifying the default file format used by StreamRecorder. Value: <b>streamRecorderFileFormat</b>
public static final	<a href="#">PROPERTY_FILE_VERSION_DELEGATE</a> Application Property "streamRecorderFileVersionDelegate": for specifying a default custom file version delegate. Value: <b>streamRecorderFileVersionDelegate</b>

public static final	<u><a href="#">PROPERTY_FILE_VERSION_TEMPLATE</a></u> Application Property "streamRecorderFileVersionTemplate": for specifying a default file version template string. Value: <b>streamRecorderFileVersionTemplate</b>
public static final	<u><a href="#">PROPERTY_MOVE_FIRST_VIDEO_FRAME_TO_ZERO</a></u> Application Property "streamRecorderMoveFirstVideoFrameToZero": for specifying the default moveFirstVideoFrameToZero value used by StreamRecorder. Value: <b>streamRecorderMoveFirstVideoFrameToZero</b>
public static final	<u><a href="#">PROPERTY_NOTIFY_LISTENER</a></u> Application Property "streamRecorderNotifyListener": for specifying the default listener which will receive StreamRecorder notifications. Value: <b>streamRecorderNotifyListener</b>
public static final	<u><a href="#">PROPERTY_OUTPUT_FILE</a></u> Application Property "streamRecorderOutputFile": for specifying the default filename used by StreamRecorder for recordings. Value: <b>streamRecorderOutputFile</b>
public static final	<u><a href="#">PROPERTY_OUTPUT_PATH</a></u> Application Property "streamRecorderOutputPath": for specifying the default path used by StreamRecorder to write recordings. Value: <b>streamRecorderOutputPath</b>
public static final	<u><a href="#">PROPERTY_RECORD_DATA</a></u> Application Property "streamRecorderRecordData": for specifying the default recordData value used by StreamRecorders. Value: <b>streamRecorderRecordData</b>
public static final	<u><a href="#">PROPERTY_SEGMENT_DURATION</a></u> Application Property "streamRecorderSegmentDuration": for specifying the default SegmentDuration used by StreamRecorder when the segmentationType is set to SEGMENT_BY_DURATION. Value: <b>streamRecorderSegmentDuration</b>
public static final	<u><a href="#">PROPERTY_SEGMENT_SCHEDULE</a></u> Application Property "streamRecorderSegmentSchedule": for specifying the default SegmentSchedule string used by StreamRecorder when the segmentationType is set to SEGMENT_BY_SCHEDULE. Value: <b>streamRecorderSegmentSchedule</b>
public static final	<u><a href="#">PROPERTY_SEGMENT_SIZE</a></u> Application Property "streamRecorderSegmentSize": for specifying the default SegmentSize used by StreamRecorder when the segmentationType is set to SEGMENT_BY_SIZE. Value: <b>streamRecorderSegmentSize</b>
public static final	<u><a href="#">PROPERTY_SEGMENTATION_TYPE</a></u> Application Property "streamRecorderSegmentationType": for specifying the default segmentation type used by StreamRecorder. Value: <b>streamRecorderSegmentationType</b>
public static final	<u><a href="#">PROPERTY_SPLIT_ON_TC_DISCONTINUITY</a></u> Application Property "streamRecorderSplitOnTcDiscontinuity": for specifying the default splitOnTcDiscontinuity value used by StreamRecorder. Value: <b>streamRecorderSplitOnTcDiscontinuity</b>

public static final	<u><a href="#">PROPERTY_START_ON_KEYFRAME</a></u> Application Property "streamRecorderStartOnKeyFrame": for specifying the default startOnKeyFrame value used by StreamRecorder. Value: <b>streamRecorderStartOnKeyFrame</b>
public static final	<u><a href="#">PROPERTY_USE_SIMPLE_FILE_VERSION_NAMING</a></u> Application Property "streamRecorderUseSimpleFileVersionNaming": for using StreamRecorderSimpleFileVersionDelegate instead of StreamRecorderFileVersionDelegate as the default file version naming delegate Value: <b>streamRecorderUseSimpleFileVersionNaming</b>
public static final	<u><a href="#">PROPERTY_VERSIONING_OPTION</a></u> Application Property "streamRecorderVersioningOption": for specifying the default file versioning method used by StreamRecorders Value: <b>streamRecorderVersioningOption</b>
public static final	<u><a href="#">RECORDER_STATE_ERROR</a></u> Value returned by IStreamRecorder getRecorderState() to indicate an error has occurred in the recorder. Value: <b>-1</b>
public static final	<u><a href="#">RECORDER_STATE_ERROR_STRING</a></u> String value returned by IStreamRecorder getRecorderStateString() when the recorder state is RECORDER_STATE_ERROR. Value: <b>Recorder Error</b>
public static final	<u><a href="#">RECORDER_STATE_PENDING</a></u> Value returned by IStreamRecorder getRecorderState() to indicate the recorder has been created but it's IStreamRecorder startRecording() method has not yet been called. Value: <b>0</b>
public static final	<u><a href="#">RECORDER_STATE_PENDING_STRING</a></u> String value returned by IStreamRecorder getRecorderStateString() when the recorder state is RECORDER_STATE_PENDING. Value: <b>Pending</b>
public static final	<u><a href="#">RECORDER_STATE_RECORDING</a></u> Value returned by IStreamRecorder getRecorderState() to indicate the recorder is actively recording data to file. Value: <b>2</b>
public static final	<u><a href="#">RECORDER_STATE_RECORDING_STRING</a></u> String value returned by IStreamRecorder getRecorderStateString() when the recorder state is RECORDER_STATE_RECORDING. Value: <b>Recording</b>
public static final	<u><a href="#">RECORDER_STATE_WAITING</a></u> Value returned by IStreamRecorder getRecorderState() to indicate the recorder has been started but the associated stream is not currently being received by the server (for example the stream is not being published from the encoder) Value: <b>1</b>
public static final	<u><a href="#">RECORDER_STATE_WAITING_STRING</a></u> String value returned by IStreamRecorder getRecorderStateString() when the recorder state is RECORDER_STATE_WAITING. Value: <b>Waiting for Stream</b>
public static final	<u><a href="#">SEGMENT_BY_DURATION</a></u> Value: <b>duration</b>

public static final	<u><a href="#">SEGMENT_BY_SCHEDULE</a></u> Value: <b>schedule</b>
public static final	<u><a href="#">SEGMENT_BY_SIZE</a></u> Value: <b>size</b>
public static final	<u><a href="#">SEGMENT_NONE</a></u> Value: <b>none</b>
public static final	<u><a href="#">VERSION_FILE</a></u> Uses the file name returned by the configured fileVersionDelegate to version the new file Value: <b>version</b>

## Fields

### PROPERTY\_DEBUG\_ENABLE

public static final java.lang.String **PROPERTY\_DEBUG\_ENABLE**

Application Property "streamRecorderDebugEnable": for enabling additional log statements throughout StreamRecorder  
Add the property to the StreamRecorder Properties section of Application.xml.

Default value is false  
Valid values are true or false  
Constant value: **streamRecorderDebugEnable**

### PROPERTY\_USE\_SIMPLE\_FILE\_VERSION\_NAMING

public static final java.lang.String **PROPERTY\_USE\_SIMPLE\_FILE\_VERSION\_NAMING**

Application Property "streamRecorderUseSimpleFileVersionNaming": for using StreamRecorderSimpleFileVersionDelegate instead of StreamRecorderFileVersionDelegate as the default file version naming delegate

Add the property to the StreamRecorder Properties section of Application.xml.

Default value is false  
Valid values are true, false

Constant value: **streamRecorderUseSimpleFileVersionNaming**

See Also:

[StreamRecorderSimpleFileVersionDelegate](#)

### PROPERTY\_SEGMENTATION\_TYPE

public static final java.lang.String **PROPERTY\_SEGMENTATION\_TYPE**

Application Property "streamRecorderSegmentationType": for specifying the default segmentation type used by StreamRecorder

Add the property to the StreamRecorder Properties section of Application.xml.

Valid values are: none, schedule, size, duration

(continued from last page)

Constant value: **streamRecorderSegmentationType****See Also:**[StreamRecorderParameters.segmentationType](#)

---

## SEGMENT\_NONE

```
public static final java.lang.String SEGMENT_NONE
```

Constant value: **none**

---

## SEGMENT\_BY\_SCHEDULE

```
public static final java.lang.String SEGMENT_BY_SCHEDULE
```

Constant value: **schedule**

---

## SEGMENT\_BY\_SIZE

```
public static final java.lang.String SEGMENT_BY_SIZE
```

Constant value: **size**

---

## SEGMENT\_BY\_DURATION

```
public static final java.lang.String SEGMENT_BY_DURATION
```

Constant value: **duration**

---

## PROPERTY\_FILE\_FORMAT

```
public static final java.lang.String PROPERTY_FILE_FORMAT
```

Application Property "streamRecorderFileFormat": for specifying the default file format used by StreamRecorder.

Add the property to the StreamRecorder Properties section of Application.xml.

Valid values are: flv, mp4

Constant value: **streamRecorderFileFormat****See Also:**[StreamRecorderParameters.fileFormat](#)

---

## FORMAT\_FLV

```
public static final java.lang.String FORMAT_FLV
```

Constant value: **flv**

---

## FORMAT\_MP4

```
public static final java.lang.String FORMAT_MP4
```

(continued from last page)

Constant value: **mp4**

---

## PROPERTY\_VERSIONING\_OPTION

```
public static final java.lang.String PROPERTY_VERSIONING_OPTION
```

Application Property "streamRecorderVersioningOption": for specifying the default file versioning method used by StreamRecorders

Add the property to the StreamRecorder Properties section of Application.xml.

Valid values are: version, append, overwrite

Constant value: **streamRecorderVersioningOption**

See Also:

[StreamRecorderParameters.versioningOption](#)

---

## VERSION\_FILE

```
public static final java.lang.String VERSION_FILE
```

Uses the file name returned by the configured fileVersionDelegate to version the new file  
Constant value: **version**

See Also:

[IStreamRecorderFileVersionDelegate](#)  
[StreamRecorderFileVersionDelegate](#)  
[StreamRecorderSimpleFileVersionDelegate](#)

---

## APPEND\_FILE

```
public static final java.lang.String APPEND_FILE
```

Appends new stream data to the existing file. NOTE: To ensure playback compatibility, it is highly recommended that this option only be used if all streams recorded to the file will be using identical video and audio codecs and configurations.  
Constant value: **append**

---

## OVERWRITE\_FILE

```
public static final java.lang.String OVERWRITE_FILE
```

Overwrites the existing file with the new file  
Constant value: **overwrite**

---

## PROPERTY\_START\_ON\_KEYFRAME

```
public static final java.lang.String PROPERTY_START_ON_KEYFRAME
```

Application Property "streamRecorderStartOnKeyFrame": for specifying the default startOnKeyFrame value used by StreamRecorder.

Add the property to the StreamRecorder Properties section of Application.xml.

Constant value: **streamRecorderStartOnKeyFrame**

See Also:

[StreamRecorderParameters.startOnKeyFrame](#)

---

## PROPERTY\_RECORD\_DATA

```
public static final java.lang.String PROPERTY_RECORD_DATA
```

Application Property "streamRecorderRecordData": for specifying the default recordData value used by StreamRecorders.

Add the property to the StreamRecorder Properties section of Application.xml.

Constant value: **streamRecorderRecordData**

See Also:

[StreamRecorderParameters.recordData](#)

---

## PROPERTY\_MOVE\_FIRST\_VIDEO\_FRAME\_TO\_ZERO

```
public static final java.lang.String PROPERTY_MOVE_FIRST_VIDEO_FRAME_TO_ZERO
```

Application Property "streamRecorderMoveFirstVideoFrameToZero": for specifying the default moveFirstVideoFrameToZero value used by StreamRecorder.

Add the property to the StreamRecorder Properties section of Application.xml.

Constant value: **streamRecorderMoveFirstVideoFrameToZero**

See Also:

[StreamRecorderParameters.moveFirstVideoFrameToZero](#)

---

## PROPERTY\_SPLIT\_ON\_TC\_DISCONTINUITY

```
public static final java.lang.String PROPERTY_SPLIT_ON_TC_DISCONTINUITY
```

Application Property "streamRecorderSplitOnTcDiscontinuity": for specifying the default splitOnTcDiscontinuity value used by StreamRecorder.

Add the property to the StreamRecorder Properties section of Application.xml.

Constant value: **streamRecorderSplitOnTcDiscontinuity**

See Also:

[StreamRecorderParameters.splitOnTcDiscontinuity](#)

---

## PROPERTY\_SEGMENT\_SIZE

```
public static final java.lang.String PROPERTY_SEGMENT_SIZE
```

Application Property "streamRecorderSegmentSize": for specifying the default SegmentSize used by StreamRecorder when the segmentationType is set to SEGMENT\_BY\_SIZE.

Add the property to the StreamRecorder Properties section of Application.xml.

Constant value: **streamRecorderSegmentSize**

See Also:

[StreamRecorderParameters.segmentSize](#)

---

(continued from last page)

## PROPERTY\_SEGMENT\_DURATION

```
public static final java.lang.String PROPERTY_SEGMENT_DURATION
```

Application Property "streamRecorderSegmentDuration": for specifying the default SegmentDuration used by StreamRecorder when the segmentationType is set to SEGMENT\_BY\_DURATION.

Add the property to the StreamRecorder Properties section of Application.xml.

Constant value: **streamRecorderSegmentDuration**

See Also:

[StreamRecorderParameters.segmentDuration](#)

---

## PROPERTY\_SEGMENT\_SCHEDULE

```
public static final java.lang.String PROPERTY_SEGMENT_SCHEDULE
```

Application Property "streamRecorderSegmentSchedule": for specifying the default SegmentSchedule string used by StreamRecorder when the segmentationType is set to SEGMENT\_BY\_SCHEDULE.

Add the property to the StreamRecorder Properties section of Application.xml.

Constant value: **streamRecorderSegmentSchedule**

See Also:

[StreamRecorderParameters.segmentSchedule](#)

---

## PROPERTY\_OUTPUT\_PATH

```
public static final java.lang.String PROPERTY_OUTPUT_PATH
```

Application Property "streamRecorderOutputPath": for specifying the default path used by StreamRecorder to write recordings.

Add the property to the StreamRecorder Properties section of Application.xml.

Constant value: **streamRecorderOutputPath**

See Also:

[StreamRecorderParameters.outputPath](#)

---

## PROPERTY\_OUTPUT\_FILE

```
public static final java.lang.String PROPERTY_OUTPUT_FILE
```

Application Property "streamRecorderOutputFile": for specifying the default filename used by StreamRecorder for recordings.

Add the property to the StreamRecorder Properties section of Application.xml.

Constant value: **streamRecorderOutputFile**

See Also:

[StreamRecorderParameters.outputFile](#)

(continued from last page)

## PROPERTY\_NOTIFY\_LISTENER

```
public static final java.lang.String PROPERTY_NOTIFY_LISTENER
```

Application Property "streamRecorderNotifyListener": for specifying the default listener which will receive StreamRecorder notifications.

Add the property to the StreamRecorder Properties section of Application.xml.

Constant value: **streamRecorderNotifyListener**

See Also:

[StreamRecorderParameters.notifyListener](#)

---

## PROPERTY\_FILE\_VERSION\_DELEGATE

```
public static final java.lang.String PROPERTY_FILE_VERSION_DELEGATE
```

Application Property "streamRecorderFileVersionDelegate": for specifying a default custom file version delegate.

Add the property to the StreamRecorder Properties section of Application.xml.

Constant value: **streamRecorderFileVersionDelegate**

See Also:

[StreamRecorderParameters.fileVersionDelegate](#)

---

## PROPERTY\_FILE\_VERSION\_TEMPLATE

```
public static final java.lang.String PROPERTY_FILE_VERSION_TEMPLATE
```

Application Property "streamRecorderFileVersionTemplate": for specifying a default file version template string.  
NOTE: only applicable when using the default fileVersionDelegate

Add the property to the StreamRecorder Properties section of Application.xml.

Constant value: **streamRecorderFileVersionTemplate**

See Also:

[StreamRecorderParameters.fileTemplate](#)  
[StreamRecorderParameters.fileVersionDelegate](#)

---

## RECORDER\_STATE\_ERROR

```
public static final int RECORDER_STATE_ERROR
```

Value returned by IStreamRecorder getRecorderState() to indicate an error has occurred in the recorder.

Constant value: **-1**

See Also:

[RECORDER\\_STATE\\_ERROR\\_STRING](#)

---

## RECORDER\_STATE\_PENDING

```
public static final int RECORDER_STATE_PENDING
```

(continued from last page)

Value returned by IStreamRecorder getRecorderState() to indicate the recorder has been created but it's IStreamRecorder startRecording() method has not yet been called.

Constant value: **0**

See Also:

[RECORDER\\_STATE\\_PENDING\\_STRING](#)

---

## RECORDER\_STATE\_WAITING

public static final int **RECORDER\_STATE\_WAITING**

Value returned by IStreamRecorder getRecorderState() to indicate the recorder has been started but the associated stream is not currently being received by the server (for example the stream is not being published from the encoder)

Constant value: **1**

See Also:

[RECORDER\\_STATE\\_WAITING\\_STRING](#)

---

## RECORDER\_STATE\_RECORDING

public static final int **RECORDER\_STATE\_RECORDING**

Value returned by IStreamRecorder getRecorderState() to indicate the recorder is actively recording data to file.

Constant value: **2**

See Also:

[RECORDER\\_STATE\\_RECORDING\\_STRING](#)

---

## RECORDER\_STATE\_ERROR\_STRING

public static final java.lang.String **RECORDER\_STATE\_ERROR\_STRING**

String value returned by IStreamRecorder getRecorderStateString() when the recorder state is RECORDER\_STATE\_ERROR.

Constant value: **Recorder Error**

See Also:

[RECORDER\\_STATE\\_ERROR](#)

---

## RECORDER\_STATE\_PENDING\_STRING

public static final java.lang.String **RECORDER\_STATE\_PENDING\_STRING**

String value returned by IStreamRecorder getRecorderStateString() when the recorder state is RECORDER\_STATE\_PENDING.

Constant value: **Pending**

See Also:

[RECORDER\\_STATE\\_PENDING](#)

---

(continued from last page)

## RECORDER\_STATE\_WAITING\_STRING

public static final java.lang.String RECORDER\_STATE\_WAITING\_STRING

String value returned by IStreamRecorder getRecorderStateString() when the recorder state is RECORDER\_STATE\_WAITING.

Constant value: Waiting for Stream

See Also:

[RECORDER\\_STATE\\_PENDING](#)

---

## RECORDER\_STATE\_RECORDING\_STRING

public static final java.lang.String RECORDER\_STATE\_RECORDING\_STRING

String value returned by IStreamRecorder getRecorderStateString() when the recorder state is RECORDER\_STATE\_RECORDING.

Constant value: Recording

See Also:

[RECORDER\\_STATE\\_RECORDING](#)

## com.wowza.wms.livestreamrecord.manager Interface IStreamRecorderFileVersionDelegate

All Known Implementing Classes:

[StreamRecorderSimpleFileVersionDelegate](#), [StreamRecorderFileVersionDelegate](#)

---

public interface **IStreamRecorderFileVersionDelegate**

extends

This interface is used to implement a custom file versioning scheme during live stream recording.

The `getFilename` method is called under 2 conditions

- 1 - when a recorder needs to create a new output file, and an file already exists by the same name.
- 2 - when a recorder creates a new file and has been configured to use any `segmentationType` other than `none`

The string returned must be a valid and fully qualified filename, including the full path.

For example:

`c:\test\myFile.flv`  
`usr/home/content/myFile.mp4`

See Also:

[StreamRecorderParameters.segmentationType](#)

---

### Method Summary

String	<a href="#">getFilename(IStreamRecorder recContext)</a>
--------	---

### Methods

#### **getFilename**

public String [getFilename\(IStreamRecorder recContext\)](#)

**Parameters:**

`recContext` - The `IStreamRecorder` requesting the file name

**Returns:**

`String` The versioned file name

## com.wowza.wms.livestreamrecord.manager Class StreamRecorderFileVersionDelegate

```
java.lang.Object
+-com.wowza.wms.livestreamrecord.model.LiveStreamRecordFileVersionDelegate
+-com.wowza.wms.livestreamrecord.manager.StreamRecorderFileVersionDelegate
```

### All Implemented Interfaces:

[IStreamRecorderFileVersionDelegate](#), com.wowza.wms.livestreamrecord.model.ILiveStreamRecordFileVersionDelegate

---

```
public class StreamRecorderFileVersionDelegate
extends com.wowza.wms.livestreamrecord.model.LiveStreamRecordFileVersionDelegate
implements com.wowza.wms.livestreamrecord.model.ILiveStreamRecordFileVersionDelegate,
IStreamRecorderFileVersionDelegate
```

Default FileVersionDelegate class

This delegate allows the configuration of a custom FileVersionName using a template string. The tags in the template string are replaced with their associated values, to generate a filename.

Valid template tags are;

\${SourceStreamName} - The name of the stream  
 \${BaseFileName} - The basefile name, which is the filename prior to any versioning changes  
 \${SegmentNumber} - An incrementing value representing the current number of segments which have been created for this recording, starting at 0  
 \${RecordingStartTime} - The time at which the recording was started formatted as yyyy-MM-dd-HH.mm.ss.SSS-zzz  
 \${SegmentTime} - Time time at which the segment was created

The default template string is

`${BaseFileName}_${RecordingStartTime}_${SegmentNumber}`

### Fields inherited from class

com.wowza.wms.livestreamrecord.model.LiveStreamRecordFileVersionDelegate

ALT\_DATETIME\_FORMAT, BASE\_NAME\_TAG, DATETIME\_FORMAT, SEGMENT\_NUMBER\_TAG, SEGMENT\_TIME\_TAG,  
START\_TIME\_TAG, STREAM\_NAME\_TAG

## Constructor Summary

public	<a href="#">StreamRecorderFileVersionDelegate()</a>
--------	---

## Method Summary

String	<a href="#">getFilename(IStreamRecorder recContext)</a> Creates a new filename using the default template string,
--------	--

### Methods inherited from class

com.wowza.wms.livestreamrecord.model.LiveStreamRecordFileVersionDelegate

`createFileName, getFilename, getFileTemplate, setFileTemplate`

### Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait`

**Methods inherited from interface**`com.wowza.wms.livestreamrecord.model.ILiveStreamRecordFileVersionDelegate``getFilename`**Methods inherited from interface**`com.wowza.wms.livestreamrecord.manager.IStreamRecorderFileVersionDelegate``getFilename`

## Constructors

### StreamRecorderFileVersionDelegate

`public StreamRecorderFileVersionDelegate()`

## Methods

### getFilename

`public String getFilename(IStreamRecorder recContext)`

Creates a new filename using the default template string,

**Parameters:**

`recContext` - The IStreamRecorder requesting the file name

**Returns:**

`String` The versioned file name

## com.wowza.wms.livestreamrecord.manager Class StreamRecorderParameters

```
java.lang.Object
+-com.wowza.wms.livestreamrecord.manager.StreamRecorderParameters
```

```
public class StreamRecorderParameters
extends Object
```

This class is used to configure the recording parameters for the StreamRecorder class.  
All members of this class are public and therefore can be accessed and modified directly.

### Field Summary

public	<a href="#"><u>fileFormat</u></a> Configures the output file format
public	<a href="#"><u>fileTemplate</u></a> Configures the file version template string used by the default fileVersionDelegate, to generate a versioned file name for the recorder NOTE: if set to null then a default file template string, as defined by StreamRecorderFileVersionDelegate, will be used
public	<a href="#"><u>fileVersionDelegate</u></a> Configures the fileVersionDelegate for the recorder Valid values are the fully qualified class path to the class which implements the IStreamRecorderFileVersionDelegate interface. Default value is com.wowza.wms.livestreamrecord.manager.StreamRecorderFileVersionDelegate
public	<a href="#"><u>moveFirstVideoFrameToZero</u></a> Configures whether or not the recorder will set the first packet's timecode to 0, when recording fileFormat is FORMAT_MP4
public	<a href="#"><u>notifyListener</u></a> Configures the StreamRecorder listener for the recorder Valid values are the fully qualified class path to the class which implements the IStreamRecorderActionNotify interface.
public	<a href="#"><u>outputFile</u></a> Configures the output file name used for the recording If segmentationType is not SEGMENT_NONE, then this value is used as the base file name for the file segments when using either of the fileVersionDelegates NOTE 1: When set to null or "", the recorder will create the filename based upon the stream name and the value of fileFormat.
public	<a href="#"><u>outputPath</u></a> Configures the output directory where recorded file will be written NOTE: When set to null or "", the recorder will write files to the default content directory for the application.
public	<a href="#"><u>recordData</u></a> Configures whether or not the recorder will write data packets to the output file.
public	<a href="#"><u>segmentationType</u></a> Configures the type of recording segmentation the recorder will use Segmentation provides a means by which the recording of an incoming live stream can be split into a series of files in order to manage them more easily.

public	<a href="#"><u>segmentDuration</u></a>
	Configures the output file size in milliseconds for each recording segment, when segmentationType is SEGMENT_BY_DURATION. This value is specified in milliseconds
public	<a href="#"><u>segmentSchedule</u></a>
	Configures the crontab schedule used to segment a recording, when segmentationType is SEGMENT_BY_SCHEDULE.
public	<a href="#"><u>segmentSize</u></a>
	Configures the output file size in bytes for each recording segment, when segmentationType is SEGMENT_BY_SIZE. This value is specified in bytes
public	<a href="#"><u>splitOnTcDiscontinuity</u></a>
	Configures whether or not the recorder will create a new segment for a recording when packet timecode discontinuities are detected. A discontinuity is said to occur when 10 consecutive packets have timecodes which jump backward in time.
public	<a href="#"><u>startOnKeyFrame</u></a>
	Configures whether or not the recorder will wait until a video key frame is received to start writing packets to the output file, thus ensuring the first packet in the file contains a video key frame.
public	<a href="#"><u>versioningOption</u></a>
	Configures the behavior when a new recording file is created with a name that already exists

## Constructor Summary

public	<a href="#"><u>StreamRecorderParameters (IAplicationInstance appInstance)</u></a>
	The constructor will read any defaults set via properties in application.xml and will use those values as the default values for this class.

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait

## Fields

### segmentationType

public java.lang.String **segmentationType**

Configures the type of recording segmentation the recorder will use

Segmentation provides a means by which the recording of an incoming live stream can be split into a series of files in order to manage them more easily. The files can be limited by; size, duration, or split at a specific time (e.g. at the top of every hour). If SEGMENT\_NONE is used, then the recording is written to a single file.

Valid values are defined in IStreamRecorderConstants

#### See Also:

[IStreamRecorderConstants.SEGMENT\\_NONE](#)  
[IStreamRecorderConstants.SEGMENT\\_BY\\_DURATION](#)  
[IStreamRecorderConstants.SEGMENT\\_BY\\_SIZE](#)  
[IStreamRecorderConstants.SEGMENT\\_BY\\_SCHEDULE](#)

(continued from last page)

## versioningOption

public java.lang.String **versioningOption**

Configures the behavior when a new recording file is created with a name that already exists

NOTE: APPEND\_FILE and OVERWRITE\_FILE are only applicable when the segmentationType is SEGMENT\_NONE

Valid values are defined in IStreamRecorderConstants

See Also:

[IStreamRecorderConstants.VERSION\\_FILE](#)  
[IStreamRecorderConstants.APPEND\\_FILE](#)  
[IStreamRecorderConstants.OVERWRITE\\_FILE](#)  
[segmentSize](#)  
[segmentDuration](#)  
[segmentSchedule](#)

---

## fileFormat

public java.lang.String **fileFormat**

Configures the output file format

Valid values are defined in IStreamRecorderConstants

See Also:

[IStreamRecorderConstants.FORMAT\\_FLV](#)  
[IStreamRecorderConstants.FORMAT\\_MP4](#)

---

## startOnKeyFrame

public boolean **startOnKeyFrame**

Configures whether or not the recorder will wait until a video key frame is received to start writing packets to the output file, thus ensuring the first packet in the file contains a video key frame.

Valid values are true, false  
default value is true

---

## recordData

public boolean **recordData**

Configures whether or not the recorder will write data packets to the output file.

Valid values are true, false  
default value is false

---

## moveFirstVideoFrameToZero

public boolean **moveFirstVideoFrameToZero**

Configures whether or not the recorder will set the first packet's timecode to 0, when recording fileFormat is FORMAT\_MP4

Valid values are true, false  
default value is true

(continued from last page)

## splitOnTcDiscontinuity

```
public boolean splitOnTcDiscontinuity
```

Configures whether or not the recorder will create a new segment for a recording when packet timecode discontinuities are detected

A discontinuity is said to occur when 10 consecutive packets have timecodes which jump backward in time. i.e. the current packet's timecode is less than the previous packet's timecode.

Valid values are true, false  
default value is false

---

## segmentSize

```
public long segmentSize
```

Configures the output file size in bytes for each recording segment, when segmentationType is SEGMENT\_BY\_SIZE.  
This value is specified in bytes

Valid values are any positive long value greater than 0, however a minimum size of 1048576 bytes (1 MB) is highly recommended to ensure the recorded file segments can start on a video key frame if configured to do so.  
default value is 10485760 bytes (10 MB)

---

## segmentDuration

```
public long segmentDuration
```

Configures the output file size in milliseconds for each recording segment, when segmentationType is SEGMENT\_BY\_DURATION.

This value is specified in milliseconds

Valid values are any positive long value greater than 0, however a minimum size of 60000 ms (1 minute) is highly recommended to ensure the recorded file segments can start on a video key frame if configured to do so.  
default value is 900000 ms (15 minutes)

---

## segmentSchedule

```
public java.lang.String segmentSchedule
```

(continued from last page)

Configures the crontab schedule used to segment a recording, when segmentationType is SEGMENT\_BY\_SCHEDULE.

#### Crontab expressions

A crontab expression specifies the segmentation interval in minutes, hours, months, years, days of the month, or days of the week. The available crontab fields are (in order):

[Minute] [Hour] [Day\_of\_the\_Month] [Month\_of\_the\_Year] [Day\_of\_the\_Week] [Year]  
where:

Minute - Specifies the minute value (between 0 and 59) at which the segmentation occurs.

Hour - Specifies the hour (between 0 and 23 based on a 24-hour clock) at which the segmentation occurs.

Day\_of\_the\_Month - Specifies the day (between 0 and 31) at which the split occurs. To specify that live recordings always be segmentation on the last day of the month, specify 31.

Month\_of\_the\_Year - Specifies the month at which the segmentation occurs. You can specify the month by using the abbreviated name of the month (for example, Jan) or a numerical equivalent (for example, 1).

Day\_of\_the\_Week - Specifies the day in a week at which the segmentation occurs. You can specify the day by using the abbreviated name of the day (for example, Mon) or a numerical equivalent (for example, 1).

Year - Specifies the year at which the segmentation occurs (for example, 2013).

#### Example

To segmentation a live recording every 30 minutes (at the top of each hour and every half-hour):

0,30 \* \* \* \*

0,30 (simplified version)

for more details see (<http://www.nncron.ru/help/EN/working/cron-format.htm>).

Note: The nnCron nonstandard question mark character (?), which specifies that the server startup time is substituted into the cron expression, is NOT currently supported.

Valid values are any valid crontab expression exception for '?'

default value is "0 \* \* \* \* \* (segment every hour at the top of the hour)

## **outputFile**

```
public java.lang.String outputFile
```

Configures the output file name used for the recording

If segmentationType is not SEGMENT\_NONE, then this value is used as the base file name for the file segments when using either of the fileVersionDelegates

NOTE 1: When set to null or "", the recorder will create the filename based upon the stream name and the value of fileFormat.  
NOTE 2: The fileFormat will be appended to the file name to ensure the file extension matches the actual file format written.

Example mytestRecording, with no fileFormat specified will result in the default fileFormat being used and a recorded file name of mytestRecording.mp4 myOtherTest.mp4, with the the fileFormat specified as 1 (FLV), will result in the recorded file name of myOtherTest.mp4.flv

Valid values are any valid filename

default value is null

## **outputPath**

```
public java.lang.String outputPath
```

Configures the output directory where recorded file will be written

NOTE: When set to null or "", the recorder will write files to the default content directory for the application.

Examples c:\myfiles\recordings /home/user/myfiles/recordings

Valid values are any valid path.

default value is null

## **notifyListener**

```
public com.wowza.wms.livestreamrecord.manager.IStreamRecorderActionNotify  
notifyListener
```

(continued from last page)

Configures the StreamRecorder listener for the recorder

Valid values are the fully qualified class path to the class which implements the IStreamRecorderActionNotify interface. default value is null

**See Also:**

[IStreamRecorderActionNotify](#)

## fileVersionDelegate

```
public com.wowza.wms.livestreamrecord.manager.IStreamRecorderFileVersionDelegate  
fileVersionDelegate
```

Configures the fileVersionDelegate for the recorder

Valid values are the fully qualified class path to the class which implements the IStreamRecorderFileVersionDelegate interface. Default value is com.wowza.wms.livestreamrecord.manager.StreamRecorderFileVersionDelegate

**See Also:**

[IStreamRecorderFileVersionDelegate](#)

[StreamRecorderFileVersionDelegate](#)

[StreamRecorderSimpleFileVersionDelegate](#)

## fileTemplate

```
public java.lang.String fileTemplate
```

Configures the file version template string used by the default fileVersionDelegate, to generate a versioned file name for the recorder

NOTE: if set to null then a default file template string, as defined by StreamRecorderFileVersionDelegate, will be used

Valid values are null or a valid file version template string

Default value is null

**See Also:**

[StreamRecorderFileVersionDelegate](#)

## Constructors

### StreamRecorderParameters

```
public StreamRecorderParameters(IAplicationInstance appInstance)
```

The constructor will read any defaults set via properties in application.xml and will use those values as the default values for this class.

**Parameters:**

appInstance

## com.wowza.wms.livestreamrecord.manager Class StreamRecorderSimpleFileVersionDelegate

```
java.lang.Object
+-com.wowza.wms.livestreamrecord.model.LiveStreamRecordSimpleFileVersionDelegate
  +-com.wowza.wms.livestreamrecord.manager.StreamRecorderSimpleFileVersionDelegate
```

### All Implemented Interfaces:

[IStreamRecorderFileVersionDelegate](#), com.wowza.wms.livestreamrecord.model.ILiveStreamRecordFileVersionDelegate

```
public class StreamRecorderSimpleFileVersionDelegate
extends com.wowza.wms.livestreamrecord.model.LiveStreamRecordSimpleFileVersionDelegate
implements com.wowza.wms.livestreamrecord.model.ILiveStreamRecordFileVersionDelegate,
IStreamRecorderFileVersionDelegate
```

StreamRecorderSimpleFileVersionDelegate class

This delegate uses the simple file versioning method to generate a file version name. The simple method is to version the existing file and always use the "basename" for the current file

#### Simple File Versioning:

The simple file versioning scheme attempts to resolve file name conflicts by appending "\_1" to the end of the existing file, so that the current recording can use the requested file name. If a file by that name already exists the appended number will be incremented and renaming will be tried again. This will continue until a non-conflicting file name is found.

For example: if the file name is "myStream.flv", and that file already exists, "myStream\_1.flv" will be tried. If that still conflicts with an existing file, "myStream\_2.flv" will be tried. This process continues until a non conflicting name is found. The recorder will then record to "mystream.flv"

## Constructor Summary

public	<a href="#">StreamRecorderSimpleFileVersionDelegate()</a>
--------	---

## Method Summary

String	<a href="#">getFilename(IStreamRecorder recContext)</a> Versions the existing file by appending _, and returns the original file name for use as the current file.
--------	---

### Methods inherited from class

com.wowza.wms.livestreamrecord.model.LiveStreamRecordSimpleFileVersionDelegate

[createFileName](#), [getFilename](#)

### Methods inherited from class java.lang.Object

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#)

### Methods inherited from interface

com.wowza.wms.livestreamrecord.model.ILiveStreamRecordFileVersionDelegate

[getFilename](#)

**Methods inherited from interface**[com.wowza.wms.livestreamrecord.manager.IStreamRecorderFileVersionDelegate](#)[getFilename](#)

## Constructors

### StreamRecorderSimpleFileVersionDelegate

```
public StreamRecorderSimpleFileVersionDelegate()
```

## Methods

### getFilename

```
public String getFilename(IStreamRecorder recContext)
```

Versions the existing file by appending \_#, and returns the original file name for use as the current file.

**Parameters:**

recContext - The IStreamRecorder requesting the file name

**Returns:**

String The original file name

---

Package

**com.wowza.wms.logging**

## com.wowza.wms.logging Interface Constants

public interface **Constants**  
extends

Constants used internally throughout log4j.

### Field Summary

public static final	<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="#"><u>HOSTNAME_KEY</u></a> hostname string literal. Value: <b>hostname</b>
public static final	<a href="#"><u>ISO8601_FORMAT</u></a> ISO8601 string literal. Value: <b>ISO8601</b>
public static final	<a href="#"><u>ISO8601_PATTERN</u></a> SimpleTimePattern for ISO8601. Value: <b>YYYY-MM-dd HH:mm:ss,sss</b>
public static final	<a href="#"><u>JNDI_CONTEXT_NAME</u></a> JNDI context name string literal. Value: <b>java:comp/env/log4j/context-name</b>
public static final	<a href="#"><u>LOG4J_ID_KEY</u></a> log4jid string literal. Value: <b>log4jid</b>
public static final	<a href="#"><u>LOG4J_PACKAGE_NAME</u></a> log4j package name string literal. Value: <b>org.apache.log4j</b>
public static final	<a href="#"><u>RECEIVER_NAME_KEY</u></a> receiver string literal. Value: <b>receiver</b>
public static final	<a href="#"><u>TEMP_CONSOLE_APPENDER_NAME</u></a> TEMP_CONSOLE_APPENDER string literal. Value: <b>TEMP_CONSOLE_APPENDER</b>
public static final	<a href="#"><u>TEMP_LIST_APPENDER_NAME</u></a> TEMP_LIST_APPENDER string literal. Value: <b>TEMP_LIST_APPENDER</b>
public static final	<a href="#"><u>TIMESTAMP_RULE_FORMAT</u></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
```

Next, edit **[install-dir]/conf/log4j.properties** and add the new field names to any **log4j.appenders.[appender-name].layout.Fields** lists to which you want to log these values. For example:

```
log4j.appenders.stdout.layout.Fields=x-severity,x-category,x-event,x-ctx,x-comment,systime-long,systime-string
```

## Method Summary

void	<a href="#"><b>onLog</b></a> (org.apache.log4j.Level level, String comment, <a href="#">IMediaStream</a> stream, String category, String event, int status, String context)
	Called each time the server logs a message.

## Methods

### onLog

```
public void onLog(org.apache.log4j.Level level,
    String comment,
    IMediaStream stream,
    String category,
    String event,
    int status,
    String context)
```

Called each time the server logs a message.

#### Parameters:

**level** - log level as defined by (org.apache.log4j.Level)  
**comment** - comment part of the log statement  
**stream** - if stream category log message it's the source stream  
**category** - log category as defined by WMSLoggerIDs.CAT\_\*

**event** - log event as defined by WMSLoggerIDs.EVT\_\*

**status** - log status (same as HTTP status field) as defined by WMSLoggerIDs.STAT\_\*

**context** - log context value like stream name, vhost name, application name

## com.wowza.wms.logging Class LogNotifyCalculateIncremental

```
java.lang.Object
+-com.wowza.wms.logging.LogNotifyCalculateIncremental
```

All Implemented Interfaces:

[ILogNotify](#)

```
public class LogNotifyCalculateIncremental
extends Object
implements ILogNotify
```

LogNotifyCalculateIncremental: pre-built implementation of ILogNotify that adds 5 new log fields:

- cs-bytes-inc (client to server bytes streamed since last log entry)
- sc-bytes-inc (server to client bytesstreamed since last log entry)
- cs-stream-bytes-inc (client to server stream bytesstreamed since last log entry)
- sc-stream-bytes-inc (server to client stream bytesstreamed since last log entry)
- x-duration-inc (time in seconds that have passed since the last log event)

These are incremental versions of the regular log values. To add this to Wowza Pro edit:

- [install-dir]/bin/setenv.bat (Windows)
- [install-dir]/bin/setenv.sh (Linux, OSX)

Add **-Dcom.wowza.wms.logging.LogNotify=com.wowza.wms.logging.LogNotifyCalculateIncremental** to the **JAVA\_OPTS**. Also, edit **[install-dir]/conf/log4j.properties** and add these field names to any **log4j.appenders.[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(org.apache.log4j.Level level, String comment, IMediaStream stream, String category, String event, int status, String context)</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.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

<code>public static final</code>	<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

<code>public</code>	<a href="#">UDPAppender()</a>
<code>public</code>	<a href="#">UDPAppender(java.net.InetAddress address, int port)</a> Sends UDP packets to the address and port.
<code>public</code>	<a href="#">UDPAppender(String host, int port)</a> Sends UDP packets to the address and port.

### Method Summary

<code>void</code>	<a href="#">activateOptions()</a> Open the UDP sender for the <b>RemoteHost</b> and <b>Port</b> .
<code>void</code>	<a href="#">append(org.apache.log4j.spi.LoggingEvent event)</a>

void	<a href="#"><u>cleanUp()</u></a> Close the UDP Socket and release the underlying connector thread if it has been created
void	<a href="#"><u>close()</u></a> Close this appender.
String	<a href="#"><u>getApplication()</u></a> Returns value of the <b>App</b> option.
String	<a href="#"><u>getEncoding()</u></a> Returns value of the <b>Encoding</b> option.
int	<a href="#"><u>getPort()</u></a> Returns value of the <b>Port</b> option.
String	<a href="#"><u>getRemoteHost()</u></a> Returns value of the <b>RemoteHost</b> option.
boolean	<a href="#"><u>isActive()</u></a>
boolean	<a href="#"><u>requiresLayout()</u></a> The UDPAppender uses layouts.
void	<a href="#"><u>setApplication(String app)</u></a> The <b>App</b> option takes a string value which should be the name of the application getting logged.
void	<a href="#"><u>setEncoding(String encoding)</u></a> The <b>Encoding</b> option specifies how the bytes are encoded.
void	<a href="#"><u>setPort(int port)</u></a> The <b>Port</b> option takes a positive integer representing the port where UDP packets will be sent.
void	<a href="#"><u>setRemoteHost(String host)</u></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> (Object o)
void	<a href="#">debug</a> (Object o, Throwable e)
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="#">debug</a> (String comment, Throwable e)
void	<a href="#">error</a> (Object o)
void	<a href="#">error</a> (Object o, Throwable e)

void	<a href="#"><u>error</u></a> (String comment)
void	<a href="#"><u>error</u></a> (String comment, <a href="#"><u>IMediaStream</u></a> stream)
void	<a href="#"><u>error</u></a> (String comment, <a href="#"><u>IMediaStream</u></a> stream, String category, String event, int status, String context)
void	<a href="#"><u>error</u></a> (String comment, String category, String event)
void	<a href="#"><u>error</u></a> (String comment, String category, String event, int status, String context)
void	<a href="#"><u>error</u></a> (String comment, Throwable e)
void	<a href="#"><u>fatal</u></a> (Object o)
void	<a href="#"><u>fatal</u></a> (Object o, Throwable e)
void	<a href="#"><u>fatal</u></a> (String comment)
void	<a href="#"><u>fatal</u></a> (String comment, <a href="#"><u>IMediaStream</u></a> stream)
void	<a href="#"><u>fatal</u></a> (String comment, <a href="#"><u>IMediaStream</u></a> stream, String category, String event, int status, String context)
void	<a href="#"><u>fatal</u></a> (String comment, String category, String event)
void	<a href="#"><u>fatal</u></a> (String comment, String category, String event, int status, String context)
void	<a href="#"><u>fatal</u></a> (String comment, Throwable e)
static <a href="#"><u>WMSLogger</u></a>	<a href="#"><u>getLogger</u></a> (String name)
void	<a href="#"><u>info</u></a> (Object o)
void	<a href="#"><u>info</u></a> (Object o, Throwable e)
void	<a href="#"><u>info</u></a> (String comment)
void	<a href="#"><u>info</u></a> (String comment, <a href="#"><u>IMediaStream</u></a> stream)
void	<a href="#"><u>info</u></a> (String comment, <a href="#"><u>IMediaStream</u></a> stream, String category, String event, int status, String context)
void	<a href="#"><u>info</u></a> (String comment, String category, String event)
void	<a href="#"><u>info</u></a> (String comment, String category, String event, int status, String context)

void	<a href="#"><u>info</u></a> (String comment, Throwable e)
boolean	<a href="#"><u>isDebugEnabled</u></a> ()
boolean	<a href="#"><u>isEnabledFor</u></a> (org.apache.log4j.Priority level)
boolean	<a href="#"><u>isErrorEnabled</u></a> ()
boolean	<a href="#"><u>isInfoEnabled</u></a> ()
boolean	<a href="#"><u>isTraceEnabled</u></a> ()
boolean	<a href="#"><u>isWarnEnabled</u></a> ()
void	<a href="#"><u>log</u></a> (org.apache.log4j.Level level, String comment)
void	<a href="#"><u>log</u></a> (org.apache.log4j.Level level, String comment, <a href="#"><u>IMediaStream</u></a> stream)
void	<a href="#"><u>log</u></a> (org.apache.log4j.Level level, String comment, <a href="#"><u>IMediaStream</u></a> stream, String category, String event)
void	<a href="#"><u>log</u></a> (org.apache.log4j.Level level, String comment, <a href="#"><u>IMediaStream</u></a> stream, String category, String event, int status, String context)
void	<a href="#"><u>log</u></a> (org.apache.log4j.Level level, String comment, <a href="#"><u>IMediaStream</u></a> stream, String category, String event, int status, String context, Throwable e)
void	<a href="#"><u>log</u></a> (org.apache.log4j.Level level, String comment, String category, String event)
void	<a href="#"><u>warn</u></a> (Object o)
void	<a href="#"><u>warn</u></a> (Object o, Throwable e)
void	<a href="#"><u>warn</u></a> (String comment)
void	<a href="#"><u>warn</u></a> (String comment, <a href="#"><u>IMediaStream</u></a> stream)
void	<a href="#"><u>warn</u></a> (String comment, <a href="#"><u>IMediaStream</u></a> stream, String category, String event, int status, String context)
void	<a href="#"><u>warn</u></a> (String comment, String category, String event)
void	<a href="#"><u>warn</u></a> (String comment, String category, String event, int status, String context)
void	<a href="#"><u>warn</u></a> (String comment, Throwable e)

**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, 17dlog, 17dlog, 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,
    Throwable e)
```

---

**log**

```
public void log(org.apache.log4j.Level level,
    String comment,
    IMediaStream stream,
    String category,
    String event,
    int status,
    String context)
```

---

**log**

```
public void log(org.apache.log4j.Level level,
    String comment,
    IMediaStream stream,
    String category,
    String event)
```

---

(continued from last page)

**log**

```
public void log(org.apache.log4j.Level level,
               String comment,
               String category,
               String event)
```

**log**

```
public void log(org.apache.log4j.Level level,
               String comment)
```

**log**

```
public void log(org.apache.log4j.Level level,
               String comment,
               IMediaStream stream)
```

**info**

```
public void info(Object o)
```

**info**

```
public void info(Object o,
                 Throwable e)
```

**info**

```
public void info(String comment)
```

**info**

```
public void info(String comment,
                 Throwable e)
```

**info**

```
public void info(String comment,
                 IMediaStream stream)
```

(continued from last page)

**info**

```
public void info(String comment,
                 String category,
                 String event)
```

**info**

```
public void info(String comment,
                 String category,
                 String event,
                 int status,
                 String context)
```

**info**

```
public void info(String comment,
                 IMediaStream stream,
                 String category,
                 String event,
                 int status,
                 String context)
```

**error**

```
public void error(Object o)
```

**error**

```
public void error(Object o,
                  Throwable e)
```

**error**

```
public void error(String comment)
```

**error**

```
public void error(String comment,
                  Throwable e)
```

**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(Object o)
```

---

**fatal**

```
public void fatal(Object o,
                  Throwable e)
```

---

**fatal**

```
public void fatal(String comment)
```

---

**fatal**

```
public void fatal(String comment,
                  Throwable e)
```

---

**fatal**

```
public void fatal(String comment,
                  IMediaStream stream)
```

---

(continued from last page)

---

**fatal**

```
public void fatal(String comment,  
                     String category,  
                     String event)
```

---

**fatal**

```
public void fatal(String comment,  
                     String category,  
                     String event,  
                     int status,  
                     String context)
```

---

**fatal**

```
public void fatal(String comment,  
                     IMediaStream stream,  
                     String category,  
                     String event,  
                     int status,  
                     String context)
```

---

**debug**

```
public void debug(Object o)
```

---

**debug**

```
public void debug(Object o,  
                     Throwable e)
```

---

**debug**

```
public void debug(String comment)
```

---

**debug**

```
public void debug(String comment,  
                     Throwable e)
```

---

(continued from last page)

**debug**

```
public void debug(String comment,  
                  IMediaStream stream)
```

**debug**

```
public void debug(String comment,  
                  String category,  
                  String event)
```

**debug**

```
public void debug(String comment,  
                  String category,  
                  String event,  
                  int status,  
                  String context)
```

**debug**

```
public void debug(String comment,  
                  IMediaStream stream,  
                  String category,  
                  String event,  
                  int status,  
                  String context)
```

**warn**

```
public void warn(Object o)
```

**warn**

```
public void warn(Object o,  
                 Throwable e)
```

**warn**

```
public void warn(String comment)
```

**warn**

```
public void warn(String comment,  
                 Throwable e)
```

---

**warn**

```
public void warn(String comment,  
                 IMediaStream stream)
```

---

**warn**

```
public void warn(String comment,  
                 String category,  
                 String event)
```

---

**warn**

```
public void warn(String comment,  
                 String category,  
                 String event,  
                 int status,  
                 String context)
```

---

**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(String name, org.apache.log4j.Logger tmpLogger)</a>
static Object	<a href="#">getGlobalLogValue(String key)</a>
static <a href="#">WMSLoggerFactory</a>	<a href="#">getInstance()</a>
static <a href="#">WMSLogger</a>	<a href="#"> getLogger(Class classObj)</a>
static <a href="#">WMSLogger</a>	<a href="#"> getLoggerObj(IApplication application)</a>
static <a href="#">WMSLogger</a>	<a href="#"> getLoggerObj(IApplicationInstance appInstance)</a>
static <a href="#">WMSLogger</a>	<a href="#"> getLoggerObj(IVHost vhost)</a>
<a href="#">WMSLogger</a>	<a href="#"> getLoggerObj(String name)</a>
static void	<a href="#">initContextLogging(IApplication application)</a>
static void	<a href="#">initContextLogging(IApplicationInstance appInstance)</a>
static void	<a href="#">initContextLogging(IVHost vhost)</a>

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(IAplication application)
```

---

**initContextLogging**

```
public static void initContextLogging(IAplicationInstance 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="#"><u>CAT_vhost</u></a> Value: <b>vhost</b>
public static final	<a href="#"><u>CAT_webm</u></a> Value: <b>webm</b>
public static final	<a href="#"><u>CTRL_playlist_node</u></a> Value: <b>CTRL-playlist-node</b>
public static final	<a href="#"><u>EVT_ALL</u></a>
public static final	<a href="#"><u>EVT_announce</u></a> Value: <b>announce</b>
public static final	<a href="#"><u>EVT_app_start</u></a> Value: <b>app-start</b>
public static final	<a href="#"><u>EVT_app_stop</u></a> Value: <b>app-stop</b>
public static final	<a href="#"><u>EVT_comment</u></a> Value: <b>comment</b>
public static final	<a href="#"><u>EVT_connect</u></a> Value: <b>connect</b>
public static final	<a href="#"><u>EVT_connect_burst</u></a> Value: <b>connect-burst</b>
public static final	<a href="#"><u>EVT_connect_pending</u></a> Value: <b>connect-pending</b>
public static final	<a href="#"><u>EVT_create</u></a> Value: <b>create</b>
public static final	<a href="#"><u>EVT_decoderaudiostart</u></a> Value: <b>decoder-audio-start</b>
public static final	<a href="#"><u>EVT_decoderaudiostop</u></a> Value: <b>decoder-audio-stop</b>
public static final	<a href="#"><u>EVT_decodervideostart</u></a> Value: <b>decoder-video-start</b>
public static final	<a href="#"><u>EVT_decodervideostop</u></a> Value: <b>decoder-video-stop</b>

public static final	<a href="#"><u>EVT_describe</u></a> Value: <b>describe</b>
public static final	<a href="#"><u>EVT_destroy</u></a> Value: <b>destroy</b>
public static final	<a href="#"><u>EVT_disconnect</u></a> Value: <b>disconnect</b>
public static final	<a href="#"><u>EVT_encoderaudiostart</u></a> Value: <b>encoder-audio-start</b>
public static final	<a href="#"><u>EVT_encoderaudiostop</u></a> Value: <b>encoder-audio-stop</b>
public static final	<a href="#"><u>EVT_encodervideostart</u></a> Value: <b>encoder-video-start</b>
public static final	<a href="#"><u>EVT_encodervideostop</u></a> Value: <b>encoder-video-stop</b>
public static final	<a href="#"><u>EVT_pause</u></a> Value: <b>pause</b>
public static final	<a href="#"><u>EVT_play</u></a> Value: <b>play</b>
public static final	<a href="#"><u>EVT_publish</u></a> Value: <b>publish</b>
public static final	<a href="#"><u>EVT_record</u></a> Value: <b>record</b>
public static final	<a href="#"><u>EVT_recordstop</u></a> Value: <b>recordstop</b>
public static final	<a href="#"><u>EVT_seek</u></a> Value: <b>seek</b>
public static final	<a href="#"><u>EVT_server_start</u></a> Value: <b>server-start</b>
public static final	<a href="#"><u>EVT_server_stop</u></a> Value: <b>server-stop</b>
public static final	<a href="#"><u>EVT_setbuffertime</u></a> Value: <b>setbuffertime</b>

public static final	<a href="#"><u>EVT_setstreamtype</u></a> Value: <b>setstreamtype</b>
public static final	<a href="#"><u>EVT_stop</u></a> Value: <b>stop</b>
public static final	<a href="#"><u>EVT_unpause</u></a> Value: <b>unpause</b>
public static final	<a href="#"><u>EVT_unpublish</u></a> Value: <b>unpublish</b>
public static final	<a href="#"><u>EVT_vhost_start</u></a> Value: <b>vhost-start</b>
public static final	<a href="#"><u>EVT_vhost_stop</u></a> Value: <b>vhost-stop</b>
public static final	<a href="#"><u>FD_ALL</u></a>
public static final	<a href="#"><u>FD_c_client_id</u></a> Value: <b>c-client-id</b>
public static final	<a href="#"><u>FD_c_ip</u></a> Value: <b>c-ip</b>
public static final	<a href="#"><u>FD_c_proto</u></a> Value: <b>c-proto</b>
public static final	<a href="#"><u>FD_c_referrer</u></a> Value: <b>c-referrer</b>
public static final	<a href="#"><u>FD_c_user_agent</u></a> Value: <b>c-user-agent</b>
public static final	<a href="#"><u>FD_cs_bytes</u></a> Value: <b>cs-bytes</b>
public static final	<a href="#"><u>FD_cs_stream_bytes</u></a> Value: <b>cs-stream-bytes</b>
public static final	<a href="#"><u>FD_cs_uri_query</u></a> Value: <b>cs-uri-query</b>
public static final	<a href="#"><u>FD_cs_uri_stem</u></a> Value: <b>cs-uri-stem</b>

public static final	<u>FD_date</u> Value: <b>date</b>
public static final	<u>FD_s_ip</u> Value: <b>s-ip</b>
public static final	<u>FD_s_port</u> Value: <b>s-port</b>
public static final	<u>FD_s_uri</u> Value: <b>s-uri</b>
public static final	<u>FD_sc_bytes</u> Value: <b>sc-bytes</b>
public static final	<u>FD_sc_stream_bytes</u> Value: <b>sc-stream-bytes</b>
public static final	<u>FD_time</u> Value: <b>time</b>
public static final	<u>FD_tz</u> Value: <b>tz</b>
public static final	<u>FD_x_app</u> Value: <b>x-app</b>
public static final	<u>FD_x_appinst</u> Value: <b>x-appinst</b>
public static final	<u>FD_x_category</u> Value: <b>x-category</b>
public static final	<u>FD_x_comment</u> Value: <b>x-comment</b>
public static final	<u>FD_x_ctx</u> Value: <b>x-ctx</b>
public static final	<u>FD_x_ctx_override</u> Value: <b>x-ctx-override</b>
public static final	<u>FD_x_duration</u> Value: <b>x-duration</b>
public static final	<u>FD_x_event</u> Value: <b>x-event</b>

public static final	<u>FD_x_file_ext</u> Value: <b>x-file-ext</b>
public static final	<u>FD_x_file_length</u> Value: <b>x-file-length</b>
public static final	<u>FD_x_file_name</u> Value: <b>x-file-name</b>
public static final	<u>FD_x_file_size</u> Value: <b>x-file-size</b>
public static final	<u>FD_x_severity</u> Value: <b>x-severity</b>
public static final	<u>FD_x_sname</u> Value: <b>x-sname</b>
public static final	<u>FD_x_sname_query</u> Value: <b>x-sname-query</b>
public static final	<u>FD_x_spos</u> Value: <b>x-spos</b>
public static final	<u>FD_x_status</u> Value: <b>x-status</b>
public static final	<u>FD_x_stream_id</u> Value: <b>x-stream-id</b>
public static final	<u>FD_x_suri</u> Value: <b>x-suri</b>
public static final	<u>FD_x_suri_query</u> Value: <b>x-suri-query</b>
public static final	<u>FD_x_suri_stem</u> Value: <b>x-suri-stem</b>
public static final	<u>FD_x_vhost</u> Value: <b>x-vhost</b>
public static final	<u>PROTO_HTTPCUPERTINO</u> Value: <b>http (cupertino)</b>
public static final	<u>PROTO_HTTPDVRCHUNK</u> Value: <b>http (dvr)</b>

public static final	<a href="#"><u>PROTO_HTTPMPG DASH</u></a> Value: <b>http (mpegdash)</b>
public static final	<a href="#"><u>PROTO_HTTPPSANJOSE</u></a> Value: <b>http (sanjose)</b>
public static final	<a href="#"><u>PROTO_HTTPSCUPERTINO</u></a> Value: <b>https (cupertino)</b>
public static final	<a href="#"><u>PROTO_HTTPSDVRCHUNK</u></a> Value: <b>https (dvr)</b>
public static final	<a href="#"><u>PROTO_HTTPPSMOOTH</u></a> Value: <b>http (smooth)</b>
public static final	<a href="#"><u>PROTO_HTTPSSANJOSE</u></a> Value: <b>https (sanjose)</b>
public static final	<a href="#"><u>PROTO_HTTPSSMOOTH</u></a> Value: <b>https (smooth)</b>
public static final	<a href="#"><u>PROTO_HTTPSSTREAMER</u></a> Value: <b>https (streamer)</b>
public static final	<a href="#"><u>PROTO_HTTPSTREAMER</u></a> Value: <b>http (streamer)</b>
public static final	<a href="#"><u>PROTO_RTMP</u></a> Value: <b>rtmp</b>
public static final	<a href="#"><u>PROTO_RTMPE</u></a> Value: <b>rtmpe</b>
public static final	<a href="#"><u>PROTO_RTMPS</u></a> Value: <b>rtmps</b>
public static final	<a href="#"><u>PROTO_RTMPT</u></a> Value: <b>rtmpt (HTTP-1.1)</b>
public static final	<a href="#"><u>PROTO_RTMPTE</u></a> Value: <b>rtmpte (HTTP-1.1)</b>
public static final	<a href="#"><u>PROTO_RTMPTS</u></a> Value: <b>rtmpts (HTTP-1.1)</b>
public static final	<a href="#"><u>PROTO_RTSP</u></a> Value: <b>rtsp</b>

public static final	<a href="#"><u>PROTO_WOWZ</u></a> Value: <b>wowz</b>
public static final	<a href="#"><u>PROTO_WOWZE</u></a> Value: <b>wowze</b>
public static final	<a href="#"><u>PROTO_WOWZS</u></a> Value: <b>wowzs</b>
public static final	<a href="#"><u>STAT_connect_application_not_available</u></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="#"><u>STAT_play_rejected_by_application</u></a> Value: <b>401</b>
public static final	<a href="#"><u>STAT_play_rejected_by_module</u></a> Value: <b>403</b>
public static final	<a href="#"><u>STAT_play_stream_not_found</u></a> Value: <b>404</b>
public static final	<a href="#"><u>STAT_play_successful</u></a> Value: <b>200</b>
public static final	<a href="#"><u>STAT_play_unsupported_media_type</u></a> Value: <b>415</b>
public static final	<a href="#"><u>STAT_publish_bad_request</u></a> Value: <b>400</b>
public static final	<a href="#"><u>STAT_publish_in_use</u></a> Value: <b>409</b>
public static final	<a href="#"><u>STAT_publish_internal_error</u></a> Value: <b>500</b>
public static final	<a href="#"><u>STAT_publish_rejected_by_application</u></a> Value: <b>401</b>
public static final	<a href="#"><u>STAT_publish_successful</u></a> Value: <b>200</b>
public static final	<a href="#"><u>STAT_publish_unsupported_media_type</u></a> Value: <b>415</b>
public static final	<a href="#"><u>STAT_stop_client_disconnect</u></a> Value: <b>408</b>

public static final	<a href="#"><u>STAT_stop_successful</u></a>
	Value: <b>200</b>

## Constructor Summary

public	<a href="#"><u>WMSLoggerIDs()</u></a>
--------	---------------------------------------

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait

## Fields

### FD\_x\_event

public static final java.lang.String **FD\_x\_event**

Constant value: **x-event**

### FD\_x\_category

public static final java.lang.String **FD\_x\_category**

Constant value: **x-category**

### FD\_date

public static final java.lang.String **FD\_date**

Constant value: **date**

### FD\_time

public static final java.lang.String **FD\_time**

Constant value: **time**

### FD\_tz

public static final java.lang.String **FD\_tz**

Constant value: **tz**

### FD\_x\_ctx

public static final java.lang.String **FD\_x\_ctx**

Constant value: **x-ctx**

**FD\_x\_ctx\_override**

```
public static final java.lang.String FD_x_ctx_override
```

Constant value: **x-ctx-override**

---

**FD\_x\_vhost**

```
public static final java.lang.String FD_x_vhost
```

Constant value: **x-vhost**

---

**FD\_x\_app**

```
public static final java.lang.String FD_x_app
```

Constant value: **x-app**

---

**FD\_x\_appinst**

```
public static final java.lang.String FD_x_appinst
```

Constant value: **x-appinst**

---

**FD\_c\_ip**

```
public static final java.lang.String FD_c_ip
```

Constant value: **c-ip**

---

**FD\_c\_proto**

```
public static final java.lang.String FD_c_proto
```

Constant value: **c-proto**

---

**FD\_s\_uri**

```
public static final java.lang.String FD_s_uri
```

Constant value: **s-uri**

---

**FD\_c\_referrer**

```
public static final java.lang.String FD_c_referrer
```

Constant value: **c-referrer**

---

(continued from last page)

## FD\_c\_user\_agent

```
public static final java.lang.String FD_c_user_agent
```

Constant value: **c-user-agent**

---

## FD\_c\_client\_id

```
public static final java.lang.String FD_c_client_id
```

Constant value: **c-client-id**

---

## FD\_cs\_bytes

```
public static final java.lang.String FD_cs_bytes
```

Constant value: **cs-bytes**

---

## FD\_sc\_bytes

```
public static final java.lang.String FD_sc_bytes
```

Constant value: **sc-bytes**

---

## FD\_x\_sname

```
public static final java.lang.String FD_x_sname
```

Constant value: **x-sname**

---

## FD\_x\_file\_size

```
public static final java.lang.String FD_x_file_size
```

Constant value: **x-file-size**

---

## FD\_x\_file\_length

```
public static final java.lang.String FD_x_file_length
```

Constant value: **x-file-length**

---

## FD\_x\_spos

```
public static final java.lang.String FD_x_spos
```

Constant value: **x-spos**

---

## FD\_cs\_stream\_bytes

```
public static final java.lang.String FD_cs_stream_bytes
```

(continued from last page)

Constant value: **cs-stream-bytes**

---

### **FD\_sc\_stream\_bytes**

```
public static final java.lang.String FD_sc_stream_bytes
```

Constant value: **sc-stream-bytes**

---

### **FD\_s\_ip**

```
public static final java.lang.String FD_s_ip
```

Constant value: **s-ip**

---

### **FD\_x\_duration**

```
public static final java.lang.String FD_x_duration
```

Constant value: **x-duration**

---

### **FD\_x\_status**

```
public static final java.lang.String FD_x_status
```

Constant value: **x-status**

---

### **FD\_cs\_uri\_stem**

```
public static final java.lang.String FD_cs_uri_stem
```

Constant value: **cs-uri-stem**

---

### **FD\_cs\_uri\_query**

```
public static final java.lang.String FD_cs_uri_query
```

Constant value: **cs-uri-query**

---

### **FD\_x\_sname\_query**

```
public static final java.lang.String FD_x_sname_query
```

Constant value: **x-sname-query**

---

### **FD\_x\_file\_name**

```
public static final java.lang.String FD_x_file_name
```

Constant value: **x-file-name**

---

**FD\_x\_file\_ext**

```
public static final java.lang.String FD_x_file_ext
```

Constant value: **x-file-ext**

---

**FD\_x\_suri\_query**

```
public static final java.lang.String FD_x_suri_query
```

Constant value: **x-suri-query**

---

**FD\_x\_suri\_stem**

```
public static final java.lang.String FD_x_suri_stem
```

Constant value: **x-suri-stem**

---

**FD\_x\_suri**

```
public static final java.lang.String FD_x_suri
```

Constant value: **x-suri**

---

**FD\_x\_severity**

```
public static final java.lang.String FD_x_severity
```

Constant value: **x-severity**

---

**FD\_x\_comment**

```
public static final java.lang.String FD_x_comment
```

Constant value: **x-comment**

---

**FD\_s\_port**

```
public static final java.lang.String FD_s_port
```

Constant value: **s-port**

---

**FD\_x\_stream\_id**

```
public static final java.lang.String FD_x_stream_id
```

Constant value: **x-stream-id**

---

(continued from last page)

## CAT\_server

```
public static final java.lang.String CAT_server
```

Constant value: **server**

---

## CAT\_vhost

```
public static final java.lang.String CAT_vhost
```

Constant value: **vhost**

---

## CAT\_application

```
public static final java.lang.String CAT_application
```

Constant value: **application**

---

## CAT\_session

```
public static final java.lang.String CAT_session
```

Constant value: **session**

---

## CAT\_stream

```
public static final java.lang.String CAT_stream
```

Constant value: **stream**

---

## CAT\_rtsp

```
public static final java.lang.String CAT_rtsp
```

Constant value: **rtsp**

---

## CAT\_cupertino

```
public static final java.lang.String CAT_cupertino
```

Constant value: **cupertino**

---

## CAT\_sanjose

```
public static final java.lang.String CAT_sanjose
```

Constant value: **sanjose**

---

## CAT\_smoothstreaming

```
public static final java.lang.String CAT_smoothstreaming
```

---

(continued from last page)

Constant value: **smoothstreaming**

---

## CAT\_dvrchunk

```
public static final java.lang.String CAT_dvrchunk
```

Constant value: **dvrchunk**

---

## CAT\_webm

```
public static final java.lang.String CAT_webm
```

Constant value: **webm**

---

## CAT\_transcoder

```
public static final java.lang.String CAT_transcoder
```

Constant value: **transcoder**

---

## CAT\_mpegdash

```
public static final java.lang.String CAT_mpegdash
```

Constant value: **mpegdash**

---

## EVT\_connect\_pending

```
public static final java.lang.String EVT_connect_pending
```

Constant value: **connect-pending**

---

## EVT\_connect

```
public static final java.lang.String EVT_connect
```

Constant value: **connect**

---

## EVT\_connect\_burst

```
public static final java.lang.String EVT_connect_burst
```

Constant value: **connect-burst**

---

## EVT\_disconnect

```
public static final java.lang.String EVT_disconnect
```

Constant value: **disconnect**

---

## EVT\_publish

```
public static final java.lang.String EVT_publish
```

Constant value: **publish**

---

## EVT\_unpublish

```
public static final java.lang.String EVT_unpublish
```

Constant value: **unpublish**

---

## EVT\_play

```
public static final java.lang.String EVT_play
```

Constant value: **play**

---

## EVT\_pause

```
public static final java.lang.String EVT_pause
```

Constant value: **pause**

---

## EVT\_setbuffertime

```
public static final java.lang.String EVT_setbuffertime
```

Constant value: **setbuffertime**

---

## EVT\_create

```
public static final java.lang.String EVT_create
```

Constant value: **create**

---

## EVT\_destroy

```
public static final java.lang.String EVT_destroy
```

Constant value: **destroy**

---

## EVT\_setstreamtype

```
public static final java.lang.String EVT_setstreamtype
```

Constant value: **setstreamtype**

---

---

(continued from last page)

## EVT\_unpause

```
public static final java.lang.String EVT_unpause
```

Constant value: **unpause**

---

## EVT\_seek

```
public static final java.lang.String EVT_seek
```

Constant value: **seek**

---

## EVT\_stop

```
public static final java.lang.String EVT_stop
```

Constant value: **stop**

---

## EVT\_record

```
public static final java.lang.String EVT_record
```

Constant value: **record**

---

## EVT\_recordstop

```
public static final java.lang.String EVT_recordstop
```

Constant value: **recordstop**

---

## EVT\_server\_start

```
public static final java.lang.String EVT_server_start
```

Constant value: **server-start**

---

## EVT\_server\_stop

```
public static final java.lang.String EVT_server_stop
```

Constant value: **server-stop**

---

## EVT\_vhost\_start

```
public static final java.lang.String EVT_vhost_start
```

Constant value: **vhost-start**

---

## EVT\_vhost\_stop

```
public static final java.lang.String EVT_vhost_stop
```

(continued from last page)

Constant value: **vhost-stop**

---

### EVT\_app\_start

```
public static final java.lang.String EVT_app_start
```

Constant value: **app-start**

---

### EVT\_app\_stop

```
public static final java.lang.String EVT_app_stop
```

Constant value: **app-stop**

---

### EVT\_comment

```
public static final java.lang.String EVT_comment
```

Constant value: **comment**

---

### EVT\_announce

```
public static final java.lang.String EVT_announce
```

Constant value: **announce**

---

### EVT\_describe

```
public static final java.lang.String EVT_describe
```

Constant value: **describe**

---

### EVT\_decoderaudiostart

```
public static final java.lang.String EVT_decoderaudiostart
```

Constant value: **decoder-audio-start**

---

### EVT\_decoderaudiostop

```
public static final java.lang.String EVT_decoderaudiostop
```

Constant value: **decoder-audio-stop**

---

### EVT\_decodervideostart

```
public static final java.lang.String EVT_decodervideostart
```

Constant value: **decoder-video-start**

---

**EVT\_decodervideostop**

```
public static final java.lang.String EVT_decodervideostop
```

Constant value: **decoder-video-stop**

---

**EVT\_encoderaudiostart**

```
public static final java.lang.String EVT_encoderaudiostart
```

Constant value: **encoder-audio-start**

---

**EVT\_encoderaudiostop**

```
public static final java.lang.String EVT_encoderaudiostop
```

Constant value: **encoder-audio-stop**

---

**EVT\_encodervideostart**

```
public static final java.lang.String EVT_encodervideostart
```

Constant value: **encoder-video-start**

---

**EVT\_encodervideostop**

```
public static final java.lang.String EVT_encodervideostop
```

Constant value: **encoder-video-stop**

---

**CTRL\_playlist\_node**

```
public static final java.lang.String CTRL_playlist_node
```

Constant value: **CTRL-playlist-node**

---

**STAT\_connect\_pending\_wating**

```
public static final int STAT_connect_pending_wating
```

Constant value: **100**

---

**STAT\_connect\_successful**

```
public static final int STAT_connect_successful
```

Constant value: **200**

---

(continued from last page)

**STAT\_connect\_application\_not\_available**

```
public static final int STAT_connect_application_not_available
```

Constant value: **302**

---

**STAT\_connect\_unknown\_protocol**

```
public static final int STAT_connect_unknown_protocol
```

Constant value: **400**

---

**STAT\_connect\_rejected\_by\_application**

```
public static final int STAT_connect_rejected_by_application
```

Constant value: **401**

---

**STAT\_connect\_rejected\_by\_module**

```
public static final int STAT_connect_rejected_by_module
```

Constant value: **403**

---

**STAT\_connect\_application\_not\_found**

```
public static final int STAT_connect_application_not_found
```

Constant value: **404**

---

**STAT\_connect\_resource\_limit**

```
public static final int STAT_connect_resource_limit
```

Constant value: **409**

---

**STAT\_connect\_license\_limit**

```
public static final int STAT_connect_license_limit
```

Constant value: **413**

---

**STAT\_connect\_redirect**

```
public static final int STAT_connect_redirect
```

Constant value: **302**

---

**STAT\_connect\_internal\_error**

```
public static final int STAT_connect_internal_error
```

(continued from last page)

Constant value: **500**

---

### **STAT\_connect\_bad\_gateway**

```
public static final int STAT_connect_bad_gateway
```

Constant value: **502**

---

### **STAT\_connect\_service\_unavailable**

```
public static final int STAT_connect_service_unavailable
```

Constant value: **503**

---

### **STAT\_play\_successful**

```
public static final int STAT_play_successful
```

Constant value: **200**

---

### **STAT\_play\_bad\_request**

```
public static final int STAT_play_bad_request
```

Constant value: **400**

---

### **STAT\_play\_rejected\_by\_application**

```
public static final int STAT_play_rejected_by_application
```

Constant value: **401**

---

### **STAT\_play\_rejected\_by\_module**

```
public static final int STAT_play_rejected_by_module
```

Constant value: **403**

---

### **STAT\_play\_stream\_not\_found**

```
public static final int STAT_play_stream_not_found
```

Constant value: **404**

---

### **STAT\_play\_unsupported\_media\_type**

```
public static final int STAT_play_unsupported_media_type
```

Constant value: **415**

---

**STAT\_play\_internal\_error**

```
public static final int STAT_PLAY_INTERNAL_ERROR
```

Constant value: 500

---

**STAT\_publish\_successful**

```
public static final int STAT_PUBLISH_SUCCESSFUL
```

Constant value: 200

---

**STAT\_publish\_bad\_request**

```
public static final int STAT_PUBLISH_BAD_REQUEST
```

Constant value: 400

---

**STAT\_publish\_rejected\_by\_application**

```
public static final int STAT_PUBLISH_REJECTED_BY_APPLICATION
```

Constant value: 401

---

**STAT\_publish\_in\_use**

```
public static final int STAT_PUBLISH_IN_USE
```

Constant value: 409

---

**STAT\_publish\_unsupported\_media\_type**

```
public static final int STAT_PUBLISH_UNSUPPORTED_MEDIA_TYPE
```

Constant value: 415

---

**STAT\_publish\_internal\_error**

```
public static final int STAT_PUBLISH_INTERNAL_ERROR
```

Constant value: 500

---

**STAT\_stop\_successful**

```
public static final int STAT_STOP_SUCCESSFUL
```

Constant value: 200

---

---

(continued from last page)

## **STAT\_stop\_client\_disconnect**

```
public static final int STAT_stop_client_disconnect
```

Constant value: **408**

---

## **STAT\_general\_successful**

```
public static final int STAT_general_successful
```

Constant value: **200**

---

## **STAT\_general\_internal\_error**

```
public static final int STAT_general_internal_error
```

Constant value: **500**

---

## **PROTO\_RTMP**

```
public static final java.lang.String PROTO_RTMP
```

Constant value: **rtmp**

---

## **PROTO\_RTMPS**

```
public static final java.lang.String PROTO_RTMPS
```

Constant value: **rtmps**

---

## **PROTO\_RTMPT**

```
public static final java.lang.String PROTO_RTMPT
```

Constant value: **rtmpt (HTTP-1.1)**

---

## **PROTO\_RTMPTS**

```
public static final java.lang.String PROTO_RTMPTS
```

Constant value: **rtmpts (HTTP-1.1)**

---

## **PROTO\_RTMPE**

```
public static final java.lang.String PROTO_RTMPE
```

Constant value: **rtmpe**

---

## **PROTO\_RTMPTE**

```
public static final java.lang.String PROTO_RTMPTE
```

---

(continued from last page)

Constant value: **rtmpte (HTTP-1.1)**

---

## PROTO\_WOWZ

```
public static final java.lang.String PROTO_WOWZ
```

Constant value: **wowz**

---

## PROTO\_WOWZE

```
public static final java.lang.String PROTO_WOWZE
```

Constant value: **wowze**

---

## PROTO\_WOWZS

```
public static final java.lang.String PROTO_WOWZS
```

Constant value: **wowzs**

---

## PROTO\_RTSP

```
public static final java.lang.String PROTO_RTSP
```

Constant value: **rtsp**

---

## PROTO\_HTTPSTREAMER

```
public static final java.lang.String PROTO_HTTPSTREAMER
```

Constant value: **http (streamer)**

---

## PROTO\_HTTPCUPERTINO

```
public static final java.lang.String PROTO_HTTPCUPERTINO
```

Constant value: **http (cupertino)**

---

## PROTO\_HTTPSMOOTH

```
public static final java.lang.String PROTO_HTTPSMOOTH
```

Constant value: **http (smooth)**

---

## PROTO\_HTTPSANJOSE

```
public static final java.lang.String PROTO_HTTPSANJOSE
```

Constant value: **http (sanjose)**

---

## PROTO\_HTTPDVRCHUNK

```
public static final java.lang.String PROTO_HTTPDVRCHUNK
```

Constant value: **http (dvr)**

---

## PROTO\_HTTPMPEGDASH

```
public static final java.lang.String PROTO_HTTPMPEGDASH
```

Constant value: **http (mpegdash)**

---

## PROTO\_HTTPSSTREAMER

```
public static final java.lang.String PROTO_HTTPSSTREAMER
```

Constant value: **https (streamer)**

---

## PROTO\_HTTPSCUPERTINO

```
public static final java.lang.String PROTO_HTTPSCUPERTINO
```

Constant value: **https (cupertino)**

---

## PROTO\_HTTPSSMOOTH

```
public static final java.lang.String PROTO_HTTPSSMOOTH
```

Constant value: **https (smooth)**

---

## PROTO\_HTTPSSANJOSE

```
public static final java.lang.String PROTO_HTTPSSANJOSE
```

Constant value: **https (sanjose)**

---

## PROTO\_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
```

(continued from last page)

---

## **EVT\_ALL**

```
public static final java.lang.String EVT_ALL
```

## Constructors

### **WMSLoggerIDs**

```
public WMSLoggerIDs()
```

---

Package

**com.wowza.wms.media.mp3.model.idtags**

## com.wowza.wms.media.mp3.model.idtags Class ID3Frames

```
java.lang.Object
+-com.wowza.wms.media.mp3.model.idtags.ID3Frames
```

---

```
public class ID3Frames
extends Object
```

### Field Summary

public static final	<a href="#">ID3FOOTER_SIZE</a>
	Value: <b>10</b>
public static final	<a href="#">ID3HEADER_SIZE</a>
	Value: <b>10</b>
public static final	<a href="#">ID3HEADER_VERSION</a>
	Value: <b>1024</b>
public static final	<a href="#">ID3HEADERFLAGS_DEFAULT</a>
	Value: <b>0</b>
public static final	<a href="#">ID3HEADERFLAGS_EXPERIMENTAL</a>
	Value: <b>32</b>
public static final	<a href="#">ID3HEADERFLAGS_EXTENDED</a>
	Value: <b>64</b>
public static final	<a href="#">ID3HEADERFLAGS_FOOTERPRESENT</a>
	Value: <b>16</b>
public static final	<a href="#">ID3HEADERFLAGS_UNSYNC</a>
	Value: <b>128</b>

### Constructor Summary

public	<a href="#">ID3Frames()</a>
--------	-----------------------------

### Method Summary

void	<a href="#">clear()</a>
java.util.List	<a href="#">getFrameMapIds()</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](#)

**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	<u>TAG_TBPM</u> Value: <b>TBPM</b>
public static final	<u>TAG_TCOM</u> Value: <b>TCOM</b>
public static final	<u>TAG_TCON</u> Value: <b>TCON</b>
public static final	<u>TAG_TCOP</u> Value: <b>TCOP</b>
public static final	<u>TAG_TDEN</u> Value: <b>TDEN</b>
public static final	<u>TAG_TDLY</u> Value: <b>TDLY</b>
public static final	<u>TAG_TDOR</u> Value: <b>TDOR</b>
public static final	<u>TAG_TDRC</u> Value: <b>TDRC</b>
public static final	<u>TAG_TDRL</u> Value: <b>TDRL</b>
public static final	<u>TAG_TDTG</u> Value: <b>TDTG</b>
public static final	<u>TAG_TEWC</u> Value: <b>TEWC</b>
public static final	<u>TAG_TEXT</u> Value: <b>TEXT</b>
public static final	<u>TAG_TFLT</u> Value: <b>TFLT</b>
public static final	<u>TAG_TIPL</u> Value: <b>TIPL</b>
public static final	<u>TAG_TIT1</u> Value: <b>TIT1</b>
public static final	<u>TAG_TIT2</u> Value: <b>TIT2</b>

public static final	<u>TAG_TIT3</u> Value: <b>TIT3</b>
public static final	<u>TAG_TKEY</u> Value: <b>TKEY</b>
public static final	<u>TAG_TLAN</u> Value: <b>TLAN</b>
public static final	<u>TAG_TLEN</u> Value: <b>TLEN</b>
public static final	<u>TAG_TMCL</u> Value: <b>TMCL</b>
public static final	<u>TAG_TMED</u> Value: <b>TMED</b>
public static final	<u>TAG_TMOO</u> Value: <b>TMOO</b>
public static final	<u>TAG_TOAL</u> Value: <b>TOAL</b>
public static final	<u>TAG_TOFN</u> Value: <b>TOFN</b>
public static final	<u>TAG_TOLY</u> Value: <b>TOLY</b>
public static final	<u>TAG_TOPE</u> Value: <b>TOPE</b>
public static final	<u>TAG_TOWN</u> Value: <b>TOWN</b>
public static final	<u>TAG_TPE1</u> Value: <b>TPE1</b>
public static final	<u>TAG_TPE2</u> Value: <b>TPE2</b>
public static final	<u>TAG_TPE3</u> Value: <b>TPE3</b>
public static final	<u>TAG_TPE4</u> Value: <b>TPE4</b>

public static final	<a href="#"><u>TAG_TPOS</u></a> Value: <b>TPOS</b>
public static final	<a href="#"><u>TAG_TPRO</u></a> Value: <b>TPRO</b>
public static final	<a href="#"><u>TAG_TPUB</u></a> Value: <b>TPUB</b>
public static final	<a href="#"><u>TAG_TRCK</u></a> Value: <b>TRCK</b>
public static final	<a href="#"><u>TAG_TRSN</u></a> Value: <b>TRSN</b>
public static final	<a href="#"><u>TAG_TRSO</u></a> Value: <b>TRSO</b>
public static final	<a href="#"><u>TAG_TSOA</u></a> Value: <b>TSOA</b>
public static final	<a href="#"><u>TAG_TSOP</u></a> Value: <b>TSOP</b>
public static final	<a href="#"><u>TAG_TSOT</u></a> Value: <b>TSOT</b>
public static final	<a href="#"><u>TAG_TSRC</u></a> Value: <b>TSRC</b>
public static final	<a href="#"><u>TAG_TSSE</u></a> Value: <b>TSSE</b>
public static final	<a href="#"><u>TAG_TSST</u></a> Value: <b>TSST</b>
public static final	<a href="#"><u>TAG_TXXX</u></a> Value: <b>TXXX</b>
public static final	<a href="#"><u>TAG_UNKN</u></a> Value: <b>UNKN</b>
public static final	<a href="#"><u>TAG_WCOM</u></a> Value: <b>WCOM</b>
public static final	<a href="#"><u>TAG_WCOP</u></a> Value: <b>WCOP</b>

public static final	<a href="#"><u>TAG_WOAF</u></a> Value: <b>WOAF</b>
public static final	<a href="#"><u>TAG_WOAR</u></a> Value: <b>WOAR</b>
public static final	<a href="#"><u>TAG_WOAS</u></a> Value: <b>WOAS</b>
public static final	<a href="#"><u>TAG_WORS</u></a> Value: <b>WORS</b>
public static final	<a href="#"><u>TAG_WPAY</u></a> Value: <b>WPAY</b>
public static final	<a href="#"><u>TAG_WPUB</u></a> Value: <b>WPUB</b>
public static final	<a href="#"><u>TAG_WXXX</u></a> Value: <b>WXXX</b>
public static final	<a href="#"><u>TAGS_TEXTINFORMATION</u></a>
public static final	<a href="#"><u>TEXTENCODING_ISO_8859_1</u></a> Value: <b>0</b>
public static final	<a href="#"><u>TEXTENCODING_UTF16</u></a> Value: <b>1</b>
public static final	<a href="#"><u>TEXTENCODING_UTF16BE</u></a> Value: <b>2</b>
public static final	<a href="#"><u>TEXTENCODING_UTF8</u></a> Value: <b>3</b>

## Constructor Summary

public	<a href="#"><u>ID3V2FrameBase</u></a> (String idStr, int flags)
--------	---

## Method Summary

static int	<a href="#"><u>byteStringLen</u></a> (String value)
static <a href="#"><u>ID3V2FrameBase</u></a>	<a href="#"><u>deserializeFrame</u></a> (byte[] buffer, int offset, int len)
static String	<a href="#"><u>deserializeString</u></a> (byte[] buffer, int offset, int len)
int	<a href="#"><u>getFlags()</u></a>

String	<a href="#">getIdStr()</a>
String	<a href="#">getMapIdStr()</a>
int	<a href="#">getSize()</a>
int	<a href="#">serialize(byte[] buffer, int offset)</a>
static int	<a href="#">serializeString(String value, byte[] buffer, int offset)</a>
static int	<a href="#">serializeStringLen(String value)</a>
void	<a href="#">setFlags(int flags)</a>
void	<a href="#">setIdStr(String idStr)</a>
static String	<a href="#">trimTrailingZero(String value)</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.media.mp3.model.idtags.IID3V2Frame**

[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
+- com.wowza.wms.media.mp3.model.idtags.ID3V2FrameBase
  +- com.wowza.wms.media.mp3.model.idtags.ID3V2FrameComment
```

All Implemented Interfaces:

[IID3V2Frame](#)

public class **ID3V2FrameComment**

extends [ID3V2FrameBase](#)

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_TDTR, 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](#)

**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
+- com.wowza.wms.media.mp3.model.idtags.ID3V2FrameBase
  +- com.wowza.wms.media.mp3.model.idtags.ID3V2FrameLinkedInformation
```

All Implemented Interfaces:

[IID3V2Frame](#)

```
public class ID3V2FrameLinkedInformation
extends ID3V2FrameBase
```

Fields inherited from class [com.wowza.wms.media.mp3.model.idtags.ID3V2FrameBase](#)

```
FLAGS_DEFAULT, HEADERSIZE, TAG_APIC, TAG_COMM, TAG_LINK, TAG_PRIV, TAG_RBUF, TAG_SYLT, TAG_TALB,
TAG_TBPM, TAG_TCOM, TAG_TCON, TAG_TCOP, TAG_TDEN, TAG_TDLY, TAG_TDOR, TAG_TDRC, TAG_TDRL,
TAG_TDTR, 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="#">setData(byte[] data)</a>
void	<a href="#">setDescription(String description)</a>
void	<a href="#">setFrameIdentifier(long frameIdentifier)</a>
void	<a href="#">setURL(String url)</a>

**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](#)

**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
+- com.wowza.wms.media.mp3.model.idtags.ID3V2FrameBase
  +- com.wowza.wms.media.mp3.model.idtags.ID3V2FramePrivate
```

All Implemented Interfaces:

[ID3V2Frame](#)

```
public class ID3V2FramePrivate
extends ID3V2FrameBase
```

Fields inherited from class [com.wowza.wms.media.mp3.model.idtags.ID3V2FrameBase](#)

```
FLAGS_DEFAULT, HEADERSIZE, TAG_APIC, TAG_COMM, TAG_LINK, TAG_PRIV, TAG_RBUF, TAG_SYLT, TAG_TALB,
TAG_TBPM, TAG_TCOM, TAG_TCON, TAG_TCOP, TAG_TDEN, TAG_TDLY, TAG_TDOR, TAG_TDRC, TAG_TDRL,
TAG_TDTR, 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>
byte[]	<a href="#">getData()</a>
String	<a href="#">getOwnerIdentifier()</a>
int	<a href="#">serializeBody</a> (byte[] buffer, int offset)
void	<a href="#">setData</a> (byte[] data)
void	<a href="#">setOwnerIdentifier</a> (String ownerIdentifier)

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()
```

---

### getOwnerIdentifier

```
public String getOwnerIdentifier()
```

(continued from last page)

---

### **setOwnerIdentifier**

```
public void setOwnerIdentifier(String ownerIdentifier)
```

---

### **getData**

```
public byte[] getData()
```

---

### **setData**

```
public void setData(byte[] data)
```

## com.wowza.wms.media.mp3.model.idtags Class ID3V2FrameRawBytes

```
java.lang.Object
+- com.wowza.wms.media.mp3.model.idtags.ID3V2FrameBase
  +- com.wowza.wms.media.mp3.model.idtags.ID3V2FrameRawBytes
```

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_TCOP, TAG_TDEN, TAG_TDLY, TAG_TDOR, TAG_TDRC, TAG_TDRL,
TAG_TDTR, 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,
```

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
+- com.wowza.wms.media.mp3.model.idtags.ID3V2FrameBase
  +- com.wowza.wms.media.mp3.model.idtags.ID3V2FrameRecommendedBufferSize
```

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_TCOP, TAG_TDEN, TAG_TDLY, TAG_TDOR, TAG_TDRC, TAG_TDRL,
TAG_TDTR, 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](#)

**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: 5
public static final	<a href="#">CONTENTTYPE_EVENTS</a>
	Value: 4
public static final	<a href="#">CONTENTTYPE LYRICS</a>
	Value: 1
public static final	<a href="#">CONTENTTYPE_MOVEMENT</a>
	Value: 3
public static final	<a href="#">CONTENTTYPE_OTHER</a>
	Value: 0
public static final	<a href="#">CONTENTTYPE_TRANSCRPTION</a>
	Value: 2
public static final	<a href="#">CONTENTTYPE_TRIVIA</a>
	Value: 6
public static final	<a href="#">CONTENTTYPE_URLIMAGES</a>
	Value: 8
public static final	<a href="#">CONTENTTYPE_URLWEBPAGES</a>
	Value: 7
public static final	<a href="#">TIMESTAMPFORMAT_MILLISECONDS</a>
	Value: 2

public static final	<a href="#">TIMESTAMPFORMAT_MPEG</a>
---------------------	--------------------------------------

Value: 1
----------

#### Fields inherited from class [com.wowza.wms.media.mp3.model.idtags.ID3V2FrameBase](#)

FLAGS_DEFAULT, HEADERSIZE, TAG_APIC, TAG_COMM, TAG_LINK, TAG_PRIV, TAG_RBUF, TAG_SYLT, TAG_TALB, TAG_TBPM, TAG_TCOM, TAG_TCON, TAG_TCOP, TAG_TDEN, TAG_TDLY, TAG_TDOR, TAG_TDRC, TAG_TDRL, TAG_TDTG, TAG_TENC, TAG_TEXT, TAG_TFLT, TAG_TIPL, TAG_TIT1, TAG_TIT2, TAG_TIT3, TAG_TKEY, TAG_TLAN, TAG_TLEN, TAG_TMCL, TAG_TMED, TAG_TMOO, TAG_TOAL, TAG_TOFN, TAG_TOLY, TAG_TOPE, TAG_TOWN, TAG_TPE1, TAG_TPE2, TAG_TPE3, TAG_TPE4, TAG_TPOS, TAG_TPRO, TAG_TPUB, TAG_TRCK, TAG_TRSN, TAG_TRSO, TAG_TSOA, TAG_TSOP, TAG_TSOT, TAG_TSRC, TAG_TSSE, TAG_TSST, TAG_TXXX, TAG_UNKN, TAG_WCOM, TAG_WCOP, TAG_WOAF, TAG_WOAR, TAG_WOAS, TAG_WORS, TAG_WPAY, TAG_WPUB, TAG_WXXX, TAGS_TEXTINFORMATION, TEXTENCODING_ISO_8859_1, TEXTENCODING_UTF16, TEXTENCODING_UTF16BE, TEXTENCODING_UTF8
---

## Constructor Summary

public	<a href="#">ID3V2FrameSynchronizedText</a> (String idStr, int flags)
public	<a href="#">ID3V2FrameSynchronizedText</a> ()

## Method Summary

void	<a href="#">addContentDescriptor</a> ( <a href="#">ID3V2FrameSynchronizedTextDescriptor</a> contentDescriptor)
void	<a href="#">addContentDescriptor</a> (long timecode, String value)
void	<a href="#">deserializeBody</a> (byte[] buffer, int offset, int len)
int	<a href="#">getBodySize</a> ()
java.util.List	<a href="#">getContentDescriptors</a> ()
byte	<a href="#">getContentType</a> ()
String	<a href="#">getLanguage</a> ()
int	<a href="#">getTextEncoding</a> ()
byte	<a href="#">getTimeStampFormat</a> ()
int	<a href="#">serializeBody</a> (byte[] buffer, int offset)
void	<a href="#">setContent-Type</a> (byte contentType)
void	<a href="#">setLanguage</a> (String language)
void	<a href="#">setTextEncoding</a> (int textEncoding)

void	<a href="#">setTimeStampFormat</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](#)

**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,
```

### 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
+- com.wowza.wms.media.mp3.model.idtags.ID3V2FrameBase
  +- com.wowza.wms.media.mp3.model.idtags.ID3V2FrameTextInformation
```

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_TDTR, 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(int textEncoding)</a>
void	<a href="#">setValue(String value)</a>

**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](#)

**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
+- com.wowza.wms.media.mp3.model.idtags.ID3V2FrameBase
  +- com.wowza.wms.media.mp3.model.idtags.ID3V2FrameURLLink
```

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_TCOP, TAG_TDEN, TAG_TDLY, TAG_TDOR, TAG_TDRC, TAG_TDRL,
TAG_TDTR, 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(byte[] b, int offset, int count)</a>
static long	<a href="#">byteArrayToLongSafeSync(byte[] b, int offset, int count)</a>
static void	<a href="#">intToByteArraySafeSync(int value, byte[] buffer, int offset, int size)</a>

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:

[ID3V2FrameBase](#)

---

public interface **IID3V2Frame**  
extends

### Method Summary

void	<a href="#">deserializeBody(byte[] buffer, int offset, int len)</a>
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(byte[] buffer, int offset)</a>
int	<a href="#">serializeBody(byte[] buffer, int offset)</a>
void	<a href="#">setFlags(int flags)</a>
void	<a href="#">setIdStr(String idStr)</a>

### Methods

#### getIdStr

public String **getIdstr()**

---

#### setIdStr

public void **setIdstr(String idStr)**

(continued from last page)

**getMapIdStr**

```
public String getMapIdstr()
```

**getFlags**

```
public int getFlags()
```

**setFlags**

```
public void setFlags(int flags)
```

**getSize**

```
public int getSize()
```

**getBodySize**

```
public int getBodySize()
```

**deserializeBody**

```
public void deserializeBody(byte[] buffer,  
                           int offset,  
                           int len)
```

**serializeBody**

```
public int serializeBody(byte[] buffer,  
                        int offset)
```

**serialize**

```
public int serialize(byte[] buffer,  
                    int offset)
```

---

Package

**com.wowza.wms.mediacache.impl**

## com.wowza.wms.mediocache.impl Interface IMediaCacheHTTPByteReader

public interface **IMediaCacheHTTPByteReader**  
extends

### Method Summary

	void <a href="#"><u>close()</u></a>
MediaCacheHTTPByteReaderResult	<a href="#"><u>head(String path)</u></a>
	void <a href="#"><u>init()</u></a>
MediaCacheHTTPByteReaderResult	<a href="#"><u>read(String path, long pos, int flen)</u></a>
	void <a href="#"><u>setConnectionTimeout(int connectionTimeout)</u></a>
	void <a href="#"><u>setDestinationAddress(String host, int port)</u></a>
	void <a href="#"><u>setMaxRetries(int maxRetries)</u></a>
	void <a href="#"><u>setProxyAddress(String host, int port)</u></a>
	void <a href="#"><u>setReadTimeout(int readTimeout)</u></a>
	void <a href="#"><u>setReceiveBufferSize(int receiveBufferSize)</u></a>
	void <a href="#"><u>setRequestFullURL(boolean requestFullURL)</u></a>
	void <a href="#"><u>setSendBufferSize(int sendBufferSize)</u></a>

### Methods

#### head

```
public MediaCacheHTTPByteReaderResult head(String path)
```

#### read

```
public MediaCacheHTTPByteReaderResult read(String path,
    long pos,
    int flen)
```

(continued from last page)

---

**close**

```
public void close()
```

---

**setDestinationAddress**

```
public void setDestinationAddress(String host,  
                                int port)
```

---

**setProxyAddress**

```
public void setProxyAddress(String host,  
                            int port)
```

---

**setRequestFullURL**

```
public void setRequestFullURL(boolean requestFullURL)
```

---

**setReadTimeout**

```
public void setReadTimeout(int readTimeout)
```

---

**setMaxRetries**

```
public void setMaxRetries(int maxRetries)
```

---

**setConnectionTimeout**

```
public void setConnectionTimeout(int connectionTimeout)
```

---

**setSendBufferSize**

```
public void setSendBufferSize(int sendBufferSize)
```

---

**setReceiveBufferSize**

```
public void setReceiveBufferSize(int receiveBufferSize)
```

---

---

## init

```
public void init()
```

## com.wowza.wms.mediocache.impl Interface IMediaCacheItemHTTPReaderFactory

public interface **IMediaCacheItemHTTPReaderFactory**  
extends **IGenericPoolFactory**

### Method Summary

void	<a href="#">addProperties(WMSProperties properties)</a>
<a href="#">IMediaCacheHTTPByteReader</a>	<a href="#">createInstance()</a>
void	<a href="#">destroyInstance(IMediaCacheHTTPByteReader instance)</a>
int	<a href="#">getConnectionTimeout()</a>
<a href="#">IMediaCacheItem</a>	<a href="#">getMediaCacheItem()</a>
int	<a href="#">getPort()</a>
int	<a href="#">getReadRetries()</a>
int	<a href="#">getReadTimeout()</a>
int	<a href="#">getReceiveBufferSize()</a>
int	<a href="#">getSendBufferSize()</a>
void	<a href="#">init()</a>
boolean	<a href="#">isRequestFullURL()</a>
void	<a href="#">setConnectionTimeout(int connectionTimeout)</a>
void	<a href="#">setDestinationAddress(String host, int port)</a>
void	<a href="#">setMediaCacheItem(IMediaCacheItem mediaCacheItem)</a>
void	<a href="#">setPort(int port)</a>
void	<a href="#">setProxyAddress(String host, int port)</a>
void	<a href="#">setReadRetries(int readRetries)</a>
void	<a href="#">setReadTimeout(int readTimeout)</a>

void	<a href="#">setReceiveBufferSize(int receiveBufferSize)</a>
void	<a href="#">setRequestFullURL(boolean requestFullURL)</a>
void	<a href="#">setSendBufferSize(int sendBufferSize)</a>

Methods inherited from interface com.wowza.util.IGenericPoolFactory

`createInstance, destroyInstance`

## Methods

### **setProxyAddress**

```
public void setProxyAddress(String host,  
    int port)
```

### **createInstance**

```
public IMediaCacheHTTPByteReader createInstance()
```

### **destroyInstance**

```
public void destroyInstance(IMediaCacheHTTPByteReader instance)
```

### **isRequestFullURL**

```
public boolean isRequestFullURL()
```

### **setRequestFullURL**

```
public void setRequestFullURL(boolean requestFullURL)
```

### **getReadTimeout**

```
public int getReadTimeout()
```

### **setReadTimeout**

```
public void setReadTimeout(int readTimeout)
```

---

### **getConnectionTimeout**

```
public int getConnectionTimeout()
```

---

### **setConnectionTimeout**

```
public void setConnectionTimeout(int connectionTimeout)
```

---

### **getPort**

```
public int getPort()
```

---

### **setPort**

```
public void setPort(int port)
```

---

### **getReadRetries**

```
public int getReadRetries()
```

---

### **setReadRetries**

```
public void setReadRetries(int readRetries)
```

---

### **getSendBufferSize**

```
public int getSendBufferSize()
```

---

### **setSendBufferSize**

```
public void setSendBufferSize(int sendBufferSize)
```

---

### **getReceiveBufferSize**

```
public int getReceiveBufferSize()
```

---

### **setReceiveBufferSize**

```
public void setReceiveBufferSize(int receiveBufferSize)
```

---

(continued from last page)

---

### setDestinationAddress

```
public void setDestinationAddress(String host,  
                                int port)
```

---

### getMediaCacheItem

```
public IMediaCacheItem getMediaCacheItem()
```

---

### setMediaCacheItem

```
public void setMediaCacheItem(IMediaCacheItem mediaCacheItem)
```

---

### init

```
public void init()
```

---

### addProperties

```
public void addProperties(WMSProperties properties)
```

## com.wowza.wms.mediocache.impl Interface IMediaCacheNameHandler

public interface **IMediaCacheNameHandler**  
extends

### Method Summary

void	<a href="#">addToEnvMap(String key, String value)</a>
String	<a href="#">getBasePath()</a>
String	<a href="#">getCanHandleFilter()</a>
String	<a href="#">getMediaNameExpression()</a>
String	<a href="#">getPathDelimiter()</a>
void	<a href="#">setBasePath(String basePath)</a>
void	<a href="#">setCanHandleFilter(String canHandleFilter)</a>
void	<a href="#">setMediaNameExpression(String mediaNameExpression)</a>
void	<a href="#">setPathDelimiter(String pathDelimiter)</a>

### Methods

#### **addToEnvMap**

```
public void addToEnvMap(String key,  
String value)
```

#### **getbasePath**

```
public String getbasePath()
```

#### **setbasePath**

```
public void setbasePath(String basePath)
```

(continued from last page)

**getCanHandleFilter**

```
public String getCanHandleFilter()
```

---

**setCanHandleFilter**

```
public void setCanHandleFilter(String canHandleFilter)
```

---

**getMediaNameExpression**

```
public String getMediaNameExpression()
```

---

**setMediaNameExpression**

```
public void setMediaNameExpression(String mediaNameExpression)
```

---

**getPathDelimiter**

```
public String getPathDelimiter()
```

---

**setPathDelimiter**

```
public void setPathDelimiter(String pathDelimiter)
```

---

## com.wowza.wms.mediocache.impl Interface IMediaCacheSourceBasicPerSourceInfo

---

public interface IMediaCacheSourceBasicPerSourceInfo  
extends

### Method Summary

String	<a href="#">getName()</a>
void	<a href="#">setName(String name)</a>

### Methods

#### **setName**

public void **setName**(String name)

---

#### **getName**

public String **getName**()

## com.wowza.wms.mediocache.impl Interface IMediaCacheSourceBasicPerSourceInfoListener

public interface IMediaCacheSourceBasicPerSourceInfoListener  
extends

### Method Summary

void	<a href="#">onPerSourceInfoAdd( IMediaCacheSourceBasicPerSourceInfo perSourceInfo)</a>
void	<a href="#">onPerSourceInfoRemove( IMediaCacheSourceBasicPerSourceInfo perSourceInfo)</a>

### Methods

#### onPerSourceInfoAdd

public void [onPerSourceInfoAdd\( IMediaCacheSourceBasicPerSourceInfo perSourceInfo\)](#)

#### onPerSourceInfoRemove

public void [onPerSourceInfoRemove\( IMediaCacheSourceBasicPerSourceInfo perSourceInfo\)](#)

---

Package

**com.wowza.wms.mediocache.model**

**com.wowza.wms.mediocache.model  
Interface IMediaCacheContextMapper**

public interface **IMediaCacheContextMapper**  
extends

**Method Summary**

String	<a href="#"><u>getMediaNameFromContext</u></a> ( <a href="#"><u>IApplicationInstance</u></a> appInstance, <a href="#"><u>IMediaStream</u></a> stream, String basePath, String mediaName, String mediaExtension)
void	<a href="#"><u>init</u></a> (MediaCache mediaCache)

**Methods****init**

public void [init](#)(MediaCache mediaCache)

**getMediaNameFromContext**

public String [getMediaNameFromContext](#)([IApplicationInstance](#) appInstance, [IMediaStream](#) stream, String basePath, String mediaName, String mediaExtension)

## com.wowza.wms.mediocache.model Interface IMediaCacheItem

public interface **IMediaCacheItem**  
extends

### Field Summary

public static final	<a href="#">PROPS_ITEM_LASTMODIFIED</a> Value: <b>mediaCacheItem.lastModified</b>
public static final	<a href="#">PROPS_ITEM_LENGTH</a> Value: <b>mediaCacheItem.length</b>
public static final	<a href="#">PROPS_ITEM_MEDIANAME</a> Value: <b>mediaCacheItem.mediaName</b>
public static final	<a href="#">PROPS_SOURCE_BLOCKSIZE</a> Value: <b>mediaCacheSource.blockSize</b>
public static final	<a href="#">PROPS_SOURCE_NAME</a> Value: <b>mediaCacheSource.name</b>

### Method Summary

void	<a href="#">addPendingWrites(MediaCacheBlockPtr blockPtr)</a>
long	<a href="#">blockSize()</a>
void	<a href="#">close()</a>
void	<a href="#">decPendingWriteCount()</a>
boolean	<a href="#">exists()</a>
long	<a href="#">getBlockCount()</a>
MediaCacheBlockPtr	<a href="#">getBlockPtr(long offset, long length)</a>
long	<a href="#">getBlockSize()</a>
MediaCacheItemInfo	<a href="#">getInfo()</a>
String	<a href="#">getLocalPath()</a>

int	<a href="#">getLockCount()</a>
long	<a href="#">getMaxTimeToLive()</a>
<a href="#">IMediaCacheSource</a>	<a href="#">getMediaCacheSource()</a>
java.io.File	<a href="#">getMediaFile()</a>
String	<a href="#">getMediaName()</a>
long	<a href="#">getMinTimeToLive()</a>
long	<a href="#">getPendingWriteCount()</a>
MediaCacheBlockPtr	<a href="#">getPendingWrites(long blockId)</a>
<a href="#">IMediaCacheStoreItem</a>	<a href="#">getStoreItem()</a>
<a href="#">IMediaCacheStoreItem</a>	<a href="#">getStoreItem(boolean doLoad)</a>
edu.emory.mathcs.backport.java.util.concurrent.locks.WMSReadWriteLock	<a href="#">getSyncLock()</a>
void	<a href="#">incPendingWriteCount()</a>
boolean	<a href="#">init(String mediaName, IMediaCacheStoreItem storeItem)</a>
boolean	<a href="#">isBlockInCache(long block)</a>
boolean	<a href="#">isLocal()</a>
boolean	<a href="#">isReadAheadTrigger(long pos, int threshold)</a>
long	<a href="#">lastModified()</a>
long	<a href="#">length()</a>
int	<a href="#">lock()</a>
long	<a href="#">posToBlock(long pos)</a>
void	<a href="#">preload()</a>
long	<a href="#">read(MediaCacheReadContext readContext, long foffset, byte[] buffer, long boffset, long length)</a>
MediaCacheReadResult	<a href="#">read(MediaCacheReadContext readContext, long foffset, long length)</a>

void	<a href="#">readAhead()</a>
MediaCacheBlockPtr	<a href="#">removePendingWrites(long blockId)</a>
void	<a href="#">saveBlock(MediaCacheBlockPtr blockPtr)</a>
void	<a href="#">setLastModified(long lastModified)</a>
void	<a href="#">setLength(long length)</a>
void	<a href="#">setLocal(boolean isLocal)</a>
void	<a href="#">setLocalPath(String localPath)</a>
void	<a href="#">setMaxTimeToLive(long maxTimeToLive)</a>
void	<a href="#">setMediaCacheSource(IMediaCacheSource mediaCacheSource)</a>
void	<a href="#">setMinTimeToLive(long minTimeToLive)</a>
void	<a href="#">triggerReadAhead(long pos)</a>
int	<a href="#">unlock()</a>
boolean	<a href="#">validate()</a>
boolean	<a href="#">validate(java.util.Properties props)</a>

## Fields

### PROPS\_ITEM\_LENGTH

public static final java.lang.String PROPS\_ITEM\_LENGTH

Constant value: `mediaCacheItem.length`

### PROPS\_ITEM\_LASTMODIFIED

public static final java.lang.String PROPS\_ITEM\_LASTMODIFIED

Constant value: `mediaCacheItem.lastModified`

### PROPS\_ITEM\_MEDIANAME

public static final java.lang.String PROPS\_ITEM\_MEDIANAME

Constant value: `mediaCacheItem.mediaName`

## PROPS\_SOURCE\_NAME

```
public static final java.lang.String PROPS_SOURCE_NAME
```

Constant value: `mediaCacheSource.name`

---

## PROPS\_SOURCE\_BLOCKSIZE

```
public static final java.lang.String PROPS_SOURCE_BLOCKSIZE
```

Constant value: `mediaCacheSource.blockSize`

## Methods

### getMediaCacheSource

```
public IMediaCacheSource getMediaCacheSource()
```

---

### setMediaCacheSource

```
public void setMediaCacheSource(IMediaCacheSource mediaCacheSource)
```

---

### lock

```
public int lock()
```

---

### unlock

```
public int unlock()
```

---

### getLockCount

```
public int getLockCount()
```

---

### getSyncLock

```
public edu.emory.mathcs.backport.java.util.concurrent.locks.WMSReadWriteLock  
getSyncLock()
```

---

### init

```
public boolean init(String mediaName,  
                   IMediaCacheStoreItem storeItem)
```

(continued from last page)

---

**close**

```
public void close()
```

---

**getBlockPtr**

```
public MediaCacheBlockPtr getBlockPtr(long offset,  
                                     long length)
```

---

**saveBlock**

```
public void saveBlock(MediaCacheBlockPtr blockPtr)
```

---

**readAhead**

```
public void readAhead()
```

---

**getMediaFile**

```
public java.io.File getMediaFile()
```

---

**isBlockInCache**

```
public boolean isBlockInCache(long block)
```

---

**blockSize**

```
public long blocksize()
```

---

**validate**

```
public boolean validate()
```

---

**validate**

```
public boolean validate(java.util.Properties props)
```

---

(continued from last page)

**getInfo**

```
public MediaCacheItemInfo getInfo()
```

---

**getMediaName**

```
public String getMediaName()
```

---

**getStoreItem**

```
public IMediaCacheStoreItem getStoreItem()
```

---

**getStoreItem**

```
public IMediaCacheStoreItem getStoreItem(boolean doLoad)
```

---

**addPendingWrites**

```
public void addPendingWrites(MediaCacheBlockPtr blockPtr)
```

---

**removePendingWrites**

```
public MediaCacheBlockPtr removePendingWrites(long blockId)
```

---

**getPendingWrites**

```
public MediaCacheBlockPtr getPendingWrites(long blockId)
```

---

**incPendingWriteCount**

```
public void incPendingWriteCount()
```

---

**decPendingWriteCount**

```
public void decPendingWriteCount()
```

---

**getPendingWriteCount**

```
public long getPendingWriteCount()
```

(continued from last page)

---

**getMinTimeToLive**

```
public long getMinTimeToLive()
```

---

**setMinTimeToLive**

```
public void setMinTimeToLive(long minTimeToLive)
```

---

**getMaxTimeToLive**

```
public long getMaxTimeToLive()
```

---

**setMaxTimeToLive**

```
public void setMaxTimeToLive(long maxTimeToLive)
```

---

**getBlockSize**

```
public long getBlockSize()
```

---

**getBlockCount**

```
public long getBlockCount()
```

---

**isReadAheadTrigger**

```
public boolean isReadAheadTrigger(long pos,  
                                int threshold)
```

---

**triggerReadAhead**

```
public void triggerReadAhead(long pos)
```

---

**posToBlock**

```
public long posToBlock(long pos)
```

---

(continued from last page)

**preload**

```
public void preload()
```

---

**getLocalPath**

```
public String getLocalPath()
```

---

**setLocalPath**

```
public void setLocalPath(String localPath)
```

---

**isLocal**

```
public boolean isLocal()
```

---

**setLocal**

```
public void setLocal(boolean isLocal)
```

---

**exists**

```
public boolean exists()
```

---

**length**

```
public long length()
```

---

**setLength**

```
public void setLength(long length)
```

---

**lastModified**

```
public long lastModified()
```

---

**setLastModified**

```
public void setLastModified(long lastModified)
```

---

(continued from last page)

---

**read**

```
public long read(MediaCacheReadContext readContext,  
                  long foffset,  
                  byte[] buffer,  
                  long boffset,  
                  long length)
```

---

**read**

```
public MediaCacheReadResult read(MediaCacheReadContext readContext,  
                                 long foffset,  
                                 long length)
```

## com.wowza.wms.mediocache.model Interface IMediaCacheReader

public interface **IMediaCacheReader**  
extends

### Method Summary

void	<a href="#">close()</a>
long	<a href="#">getFilePointer()</a>
<a href="#">IMediaCacheItem</a>	<a href="#">getMediaCacheItem()</a>
Object	<a href="#">getOpaqueObject()</a>
long	<a href="#">lastModified()</a>
long	<a href="#">length()</a>
void	<a href="#">open()</a>
int	<a href="#">read(byte[] buffer, int offset, int len)</a>
void	<a href="#">seek(long pos)</a>
void	<a href="#">setMediaCacheItem(<a href="#">IMediaCacheItem</a> mediaCacheItem)</a>
void	<a href="#">setOpaqueObject(Object object)</a>

### Methods

#### seek

```
public void seek(long pos)
```

#### getFilePointer

```
public long getFilePointer()
```

(continued from last page)

**read**

```
public int read(byte[] buffer,  
                 int offset,  
                 int len)
```

---

**setMediaCacheItem**

```
public void setMediaCacheItem(IMediaCacheItem mediaCacheItem)
```

---

**getMediaCacheItem**

```
public IMediaCacheItem getMediaCacheItem()
```

---

**open**

```
public void open()
```

---

**close**

```
public void close()
```

---

**length**

```
public long length()
```

---

**lastModified**

```
public long lastModified()
```

---

**getOpaqueObject**

```
public Object getOpaqueObject()
```

---

**setOpaqueObject**

```
public void setOpaqueObject(Object object)
```

---

## com.wowza.wms.mediocache.model Interface IMediaCacheSource

public interface **IMediaCacheSource**  
extends

### Method Summary

boolean	<a href="#">canHandle(String mediaName)</a>
String	<a href="#">getBaseClass()</a>
int	<a href="#">getDefaultValue()</a>
String	<a href="#">getDescription()</a>
long	<a href="#">getMaxTimeToLive()</a>
MediaCache	<a href="#">getMediaCache()</a>
long	<a href="#">getMinTimeToLive()</a>
String	<a href="#">getName()</a>
MediaCachePerformance Counter	<a href="#">getPerformanceCount()</a>
<a href="#">WMSProperties</a>	<a href="#">getProperties()</a>
int	<a href="#">getReadAheadThreshold()</a>
String	<a href="#">getReaderClass()</a>
String	<a href="#">getSourcePath()</a>
boolean	<a href="#">isPassThru()</a>
boolean	<a href="#">isReadAhead()</a>
<a href="#">IMediaCacheItem</a>	<a href="#">loadCacheItem(String mediaName, IMediaCacheStoreItem storeItem)</a>
<a href="#">IMediaCacheItem</a>	<a href="#">loadCacheItem(String mediaName, IMediaCacheStoreItem storeItem, long length, long lastModified)</a>
<a href="#">IMediaCacheItem</a>	<a href="#">newCacheItem(String mediaName)</a>

void	<a href="#"><u>setBaseClass</u></a> (String baseClass)
void	<a href="#"><u>setDefaultBlockSize</u></a> (int defaultBlockSize)
void	<a href="#"><u>setDescription</u></a> (String description)
void	<a href="#"><u>setMaxTimeToLive</u></a> (long maxTimeToLive)
void	<a href="#"><u>setMediaCache</u></a> (MediaCache mediaCache)
void	<a href="#"><u>setMinTimeToLive</u></a> (long minTimeToLive)
void	<a href="#"><u>setName</u></a> (String name)
void	<a href="#"><u>setPassThru</u></a> (boolean isPassThru)
void	<a href="#"><u>setProperties</u></a> (WMSProperties properties)
void	<a href="#"><u>setReadAhead</u></a> (boolean readAhead)
void	<a href="#"><u>setReadAheadThreshold</u></a> (int readAheadThreshold)
void	<a href="#"><u>setReaderClass</u></a> (String readerClass)
void	<a href="#"><u>setSourcePath</u></a> (String sourcePath)

## Methods

### canHandle

```
public boolean canHandle(String mediaName)
```

### newCacheItem

```
public IMediaCacheItem newCacheItem(String mediaName)
```

### loadCacheItem

```
public IMediaCacheItem loadCacheItem(String mediaName,  
IMediaCacheStoreItem storeItem)
```

(continued from last page)

**loadCacheItem**

```
public IMediaCacheItem loadCacheItem(String mediaName,  
    IMediaCacheStoreItem storeItem,  
    long length,  
    long lastModified)
```

**getName**

```
public String getName()
```

**setName**

```
public void setName(String name)
```

**getDescription**

```
public String getDescription()
```

**setDescription**

```
public void setDescription(String description)
```

**getBaseClass**

```
public String getBaseClass()
```

**setBaseClass**

```
public void setBaseClass(String baseClass)
```

**getSourcePath**

```
public String getSourcePath()
```

**setSourcePath**

```
public void setSourcePath(String sourcePath)
```

(continued from last page)

**getDefaultBlockSize**

```
public int getDefaultBlockSize()
```

---

**setDefaultBlockSize**

```
public void setDefaultBlockSize(int defaultBlockSize)
```

---

**getProperties**

```
public WMSProperties getProperties()
```

---

**setProperties**

```
public void setProperties(WMSProperties properties)
```

---

**getReaderClass**

```
public String getReaderClass()
```

---

**setReaderClass**

```
public void setReaderClass(String readerClass)
```

---

**getMediaCache**

```
public MediaCache getMediaCache()
```

---

**setMediaCache**

```
public void setMediaCache(MediaCache mediaCache)
```

---

**getPerformanceCount**

```
public MediaCachePerformanceCounter getPerformanceCount()
```

---

**getMinTimeToLive**

```
public long getMinTimeToLive()
```

(continued from last page)

---

**setMinTimeToLive**

```
public void setMinTimeToLive(long minTimeToLive)
```

---

**getMaxTimeToLive**

```
public long getMaxTimeToLive()
```

---

**setMaxTimeToLive**

```
public void setMaxTimeToLive(long maxTimeToLive)
```

---

**isReadAhead**

```
public boolean isReadAhead()
```

---

**setReadAhead**

```
public void setReadAhead(boolean readAhead)
```

---

**getReadAheadThreshold**

```
public int getReadAheadThreshold()
```

---

**setReadAheadThreshold**

```
public void setReadAheadThreshold(int readAheadThreshold)
```

---

**isPassThru**

```
public boolean isPassThru()
```

---

**setPassThru**

```
public void setPassThru(boolean isPassThru)
```

## com.wowza.wms.mediocache.model Interface IMediaCacheSourceNotify

---

public interface **IMediaCacheSourceNotify**  
extends

### Method Summary

void	<a href="#">onMediaCacheSourceCreate(<u>IMediaCacheSource</u> mediaCacheSource)</a>
void	<a href="#">onMediaCacheSourceDestory(<u>IMediaCacheSource</u> mediaCacheSource)</a>

### Methods

#### **onMediaCacheSourceCreate**

public void **onMediaCacheSourceCreate**([IMediaCacheSource](#) mediaCacheSource)

---

#### **onMediaCacheSourceDestory**

public void **onMediaCacheSourceDestory**([IMediaCacheSource](#) mediaCacheSource)

## com.wowza.wms.mediocache.model Interface IMediaCacheStoreItem

public interface **IMediaCacheStoreItem**  
extends

### Method Summary

byte[]	<a href="#">getBlockMask()</a>
String	<a href="#">getBlockPath()</a>
String	<a href="#">getMediaPath()</a>
java.util.Properties	<a href="#">getProperties()</a>
String	<a href="#">getPropertiesPath()</a>
MediaCacheStore	<a href="#">getStore()</a>
void	<a href="#">initMedia(long length)</a>
long	<a href="#">length()</a>
void	<a href="#">setBlockMask(byte[] blockMask)</a>
void	<a href="#">setLength(long length)</a>
void	<a href="#">setProperties(java.util.Properties props)</a>
String	<a href="#">toString()</a>
void	<a href="#">updateBlockMask(byte[] blockMask, long offset, long len)</a>

### Methods

#### **getMediaPath**

```
public String getMediaPath()
```

#### **getPropertiesPath**

```
public String getPropertiesPath()
```

(continued from last page)

---

**getBlockPath**

```
public String getBlockPath()
```

---

**getProperties**

```
public java.util.Properties getProperties()
```

---

**setProperties**

```
public void setProperties(java.util.Properties props)
```

---

**getBlockMask**

```
public byte[] getBlockMask()
```

---

**updateBlockMask**

```
public void updateBlockMask(byte[] blockMask,  
                           long offset,  
                           long len)
```

---

**setBlockMask**

```
public void setBlockMask(byte[] blockMask)
```

---

**initMedia**

```
public void initMedia(long length)
```

---

**toString**

```
public String toString()
```

---

**getStore**

```
public MediaCacheStore getStore()
```

---

---

### **length**

```
public long length()
```

---

### **setLength**

```
public void setLength(long length)
```

## com.wowza.wms.mediocache.model Interface IMediaCacheStoreNotify

---

public interface **IMediaCacheStoreNotify**  
extends

### Method Summary

void	<a href="#">onMediaCacheStoreCreate</a> (MediaCacheStore mediaCacheStore)
void	<a href="#">onMediaCacheStoreDestory</a> (MediaCacheStore mediaCacheStore)

### Methods

#### **onMediaCacheStoreCreate**

public void **onMediaCacheStoreCreate**(MediaCacheStore mediaCacheStore)

---

#### **onMediaCacheStoreDestory**

public void **onMediaCacheStoreDestory**(MediaCacheStore mediaCacheStore)

**com.wowza.wms.mediocache.model  
Interface IMediaCacheStoreTraverseFile**

public interface **IMediaCacheStoreTraverseFile**  
extends

---

**Method Summary**

void	<a href="#"><u>onFile</u></a> (java.io.File item)
------	---

**Methods****onFile**

public void **onFile**(java.io.File item)

**com.wowza.wms.mediocache.model  
Interface IMediaCacheStoreTraverseItem**

public interface **IMediaCacheStoreTraverseItem**  
extends

---

**Method Summary**

void	<a href="#">onItem(<b>IMediaCacheStoreItem</b> item)</a>
------	--

---

**Methods****onItem**

public void [onItem\(\*\*IMediaCacheStoreItem\*\* item\)](#)

---

Package

**com.wowza.wms.mediacaster**

## com.wowza.wms.mediacaster Interface IMediaCaster

public interface **IMediaCaster**  
extends

### Field Summary

public static final	<a href="#">MEDIACASTERTYPE_LIVEREPEATER</a>
	Value: <b>1</b>
public static final	<a href="#">MEDIACASTERTYPE_RTPLIVE</a>
	Value: <b>3</b>
public static final	<a href="#">MEDIACASTERTYPE_SHOUTCAST</a>
	Value: <b>2</b>
public static final	<a href="#">MEDIACASTERTYPE_UNKNOWN</a>
	Value: <b>0</b>
public static final	<a href="#">STREAMTIMEOUTREASON_GOOD</a>
	Value: <b>100</b>
public static final	<a href="#">STREAMTIMEOUTREASON_MISSING</a>
	Value: <b>101</b>
public static final	<a href="#">STREAMTIMEOUTREASON_NORTSPSESSION</a>
	Value: <b>6</b>
public static final	<a href="#">STREAMTIMEOUTREASON_NOSESSION</a>
	Value: <b>2</b>
public static final	<a href="#">STREAMTIMEOUTREASON_NOSTREAM</a>
	Value: <b>4</b>
public static final	<a href="#">STREAMTIMEOUTREASON_NOTIMEOUT</a>
	Value: <b>1</b>
public static final	<a href="#">STREAMTIMEOUTREASON_NOURL</a>
	Value: <b>3</b>
public static final	<a href="#">STREAMTIMEOUTREASON_RECONNECTRUNNING</a>
	Value: <b>5</b>
public static final	<a href="#">STREAMTIMEOUTREASON_UNKNOWN</a>
	Value: <b>0</b>

## Method Summary

boolean	<a href="#">doWatchdog()</a> Idle processor
void	<a href="#">forceReset()</a> Force a reset/reconnect of this media caster
<a href="#">IApplicationInstance</a>	<a href="#">getAppInstance()</a> Get the application instance this media caster is associated with
long	<a href="#">getConnectLastAttempt()</a> Get system time in milliseconds of last connection attempt
long	<a href="#">getConnectLastForceReset()</a> Get system time in milliseconds of last time forceReset was called
long	<a href="#">getConnectLastSuccess()</a> Get system time in milliseconds of last connection success
int	<a href="#">getIdleTimeout()</a> Get the idle timeout for this media caster (milliseconds)
Object	<a href="#">getLock()</a> Get the MediaCaster lock for an individual MediaCaster item.
<a href="#">MediaCasterItem</a>	<a href="#">getMediaCasterDef()</a> Get the media caster definition
String	<a href="#">getMediaCasterId()</a> Get the media caster id
<a href="#">MediaCasterStreamItem</a>	<a href="#">getMediaCasterStreamItem()</a> Get the media caster item associated with this media caster
int	<a href="#">getMediaCasterType()</a> Get the media caster type.
int	<a href="#">getReconnectWaitTime()</a> Get the minimum time between reconnect attempts (milliseconds)
<a href="#">IMediaStream</a>	<a href="#">getStream()</a> Get the underlying stream being used by this media caster
long	<a href="#">getStreamLastSeq()</a> Get the AMFPacket sequence number of last watchdog processed packet
long	<a href="#">getStreamMissingTime()</a> Get the time in milliseconds the stream has been missing
long	<a href="#">getStreamRunningIndex()</a> Each time a MediaCaster stream switches from not running to running this value is incremented
int	<a href="#">getStreamTimeout()</a> Get the watchdog stream timeout (milliseconds)
long	<a href="#">getStreamTimeoutLastReset()</a> Get system time in milliseconds of last time stream was reset due to stream timeout (debug)

long	<a href="#"><code>getStreamTimeoutLastTime()</code></a> Get system time in milliseconds of last time stream was considered in missing state (debug)
int	<a href="#"><code>getStreamTimeoutReason()</code></a> Get the reason the stream is in timeout condition (debug)
<a href="#"><u>IVHost</u></a>	<a href="#"><code>getVHost()</code></a> Get the virtual host associated with this media caster
void	<a href="#"><code>init(MediaCasterStreamItem mediaCasterStreamItem, MediaCasterItem mediaCasterDef, IApplicationInstance appInstance, String mediaCasterId, String streamExt)</code></a> Initialize the media caster
boolean	<a href="#"><code>isSession()</code></a> Is there current a session attached to this MediaCaster
boolean	<a href="#"><code>isStream()</code></a> Is there a stream associated with this MediaCaster
boolean	<a href="#"><code>isStreamIsRunning()</code></a> Return true if stream is currently running
void	<a href="#"><code>registerPlayer(IMediaStreamPlay player)</code></a> Register a player with this media caster
void	<a href="#"><code>sessionClosed(org.apache.mina.common.IoSession session)</code></a> sessionClosed callback
void	<a href="#"><code>sessionOpened(org.apache.mina.common.IoSession session)</code></a> sessionOpened callback
void	<a href="#"><code>setAppInstance(IApplicationInstance appInstance)</code></a> Set the application instance this media caster is assoiated with
void	<a href="#"><code>setMediaCasterDef(MediaCasterItem mediaCasterDef)</code></a> Set the media caster definition
void	<a href="#"><code>setMediaCasterId(String mediaCasterId)</code></a> Get the media caster id
void	<a href="#"><code>setMediaCasterType(int mediaCasterType)</code></a> Set the media caster type.
void	<a href="#"><code>setReconnectWaitTime(int reconnectWaitTime)</code></a> Set the minimum time between reconnect attempts (milliseconds)
void	<a href="#"><code>setStream(IMediaStream stream)</code></a> Set the underlying stream being used by this media caster
void	<a href="#"><code>setStreamRunning(boolean streamRunning)</code></a> Called internally to track when streams are running or are stopped.
void	<a href="#"><code>setStreamTimeout(int streamTimeout)</code></a> Set the watchdog stream timeout (milliseconds)
void	<a href="#"><code>shutdown(boolean isAppInstanceShutdown)</code></a> Shutdown media caster

```
void unregisterPlayer\(IMediaStreamPlay player\)
    Unregister a player with this media caster
```

## Fields

### **STREAMTIMEOUTREASON\_UNKNOWN**

```
public static final int STREAMTIMEOUTREASON_UNKNOWN
```

Constant value: **0**

### **STREAMTIMEOUTREASON\_NOTIMEOUT**

```
public static final int STREAMTIMEOUTREASON_NOTIMEOUT
```

Constant value: **1**

### **STREAMTIMEOUTREASON\_NOSESSION**

```
public static final int STREAMTIMEOUTREASON_NOSESSION
```

Constant value: **2**

### **STREAMTIMEOUTREASON\_NOURL**

```
public static final int STREAMTIMEOUTREASON_NOURL
```

Constant value: **3**

### **STREAMTIMEOUTREASON\_NOSTREAM**

```
public static final int STREAMTIMEOUTREASON_NOSTREAM
```

Constant value: **4**

### **STREAMTIMEOUTREASON\_RECONNECTRUNNING**

```
public static final int STREAMTIMEOUTREASON_RECONNECTRUNNING
```

Constant value: **5**

### **STREAMTIMEOUTREASON\_NORTSPSESSION**

```
public static final int STREAMTIMEOUTREASON_NORTSPSESSION
```

Constant value: **6**

(continued from last page)

## STREAMTIMEOUTREASON\_GOOD

```
public static final int STREAMTIMEOUTREASON_GOOD
```

Constant value: 100

## STREAMTIMEOUTREASON\_MISSING

```
public static final int STREAMTIMEOUTREASON_MISSING
```

Constant value: 101

## MEDIACASTERTYPE\_UNKNOWN

```
public static final int MEDIACASTERTYPE_UNKNOWN
```

Constant value: 0

## MEDIACASTERTYPE\_LIVEREPEATER

```
public static final int MEDIACASTERTYPE_LIVEREPEATER
```

Constant value: 1

## MEDIACASTERTYPE\_SHOUTCAST

```
public static final int MEDIACASTERTYPE_SHOUTCAST
```

Constant value: 2

## MEDIACASTERTYPE\_RTPLIVE

```
public static final int MEDIACASTERTYPE_RTPLIVE
```

Constant value: 3

## Methods

### init

```
public void init(MediaCasterStreamItem mediaCasterStreamItem,  
                 MediaCasterItem mediaCasterDef,  
                 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

**Returns:**

virtual host associated with this media caster

---

## getMediaCasterId

```
public String getMediaCasterId()
```

Get the media caster id

**Returns:**

media caster id

---

## setMediaCasterId

```
public void setMediaCasterId(String mediaCasterId)
```

Get the media caster id

**Parameters:**

mediaCasterId - media caster id

---

## getStream

```
public IMediaStream getStream()
```

Get the underlying stream being used by this media caster

**Returns:**

underlying stream being used by this media caster

---

## setStream

```
public void setStream(IMediaStream stream)
```

Set the underlying stream being used by this media caster

**Parameters:**

stream - underlying stream being used by this media caster

---

## getAppInstance

```
public IApplicationInstance getAppInstance()
```

Get the application instance this media caster is associated with

**Returns:**

application instance this media caster is associated with

---

## setAppInstance

```
public void setAppInstance(IApplicationInstance appInstance)
```

---

(continued from last page)

Set the application instance this media caster is associated with

**Parameters:**

appInstance - application instance this media caster is associated with

---

## getMediaCasterDef

```
public MediaCasterItem getMediaCasterDef()
```

Get the media caster definition

**Returns:**

media caster definition

---

## setMediaCasterDef

```
public void setMediaCasterDef(MediaCasterItem mediaCasterDef)
```

Set the media caster definition

**Parameters:**

mediaCasterDef - media caster definition

---

## shutdown

```
public void shutdown(boolean isAppInstanceShutdown)
```

Shutdown media caster

**Parameters:**

isAppInstanceShutdown - is server shutdown

---

## registerPlayer

```
public void registerPlayer(IMediaStreamPlay player)
```

Register a player with this media caster

**Parameters:**

player - player to register

---

## unregisterPlayer

```
public void unregisterPlayer(IMediaStreamPlay player)
```

Unregister a player with this media caster

**Parameters:**

player - player to unregister

---

## getIdleTimeout

```
public int getIdleTimeout()
```

Get the idle timeout for this media caster (milliseconds)

**Returns:**

idle timeout for this media caster (milliseconds)

---

## sessionOpened

```
public void sessionOpened(org.apache.mina.common.IoSession session)
```

sessionOpened callback

**Parameters:**

session - IO Session

---

## sessionClosed

```
public void sessionClosed(org.apache.mina.common.IoSession session)
```

sessionClosed callback

**Parameters:**

session - IO Session

---

## getMediaCasterStreamItem

```
public MediaCasterStreamItem getMediaCasterStreamItem()
```

Get the media caster item associated with this media caster

**Returns:**

media caster item associated with this media caster

---

## forceReset

```
public void forceReset()
```

Force a reset/reconnect of this media caster

---

## doWatchdog

```
public boolean doWatchdog()
```

Idle processor

**Returns:**

return true if media caster unloaded due to idle event

---

## getStreamTimeout

```
public int getStreamTimeout()
```

Get the watchdog stream timeout (milliseconds)

**Returns:**

stream timeout

---

## setStreamTimeout

```
public void setStreamTimeout(int streamTimeout)
```

Set the watchdog stream timeout (milliseconds)

(continued from last page)

**Parameters:**

streamTimeout - stream timeout

---

**getStreamMissingTime****public long getStreamMissingTime()**

Get the time in milliseconds the stream has been missing

**Returns:**

time in milliseconds the stream has been missing

---

**getStreamLastSeq****public long getStreamLastSeq()**

Get the AMFPacket sequence number of last watchdog processed packet

**Returns:**

AMFPacket sequence number

---

**getStreamTimeoutReason****public int getStreamTimeoutReason()**

Get the reason the stream is in timeout condition (debug)

**Returns:**

reason the stream is in timeout condition (debug)

---

**getStreamTimeoutLastTime****public long getStreamTimeoutLastTime()**

Get system time in milliseconds of last time stream was considered in missing state (debug)

**Returns:**

time in milliseconds of last time stream was considered in missing

---

**getStreamTimeoutLastReset****public long getStreamTimeoutLastReset()**

Get system time in milliseconds of last time stream was reset due to stream timeout (debug)

**Returns:**

time in milliseconds of last time stream was reset

---

**isSession****public boolean isSession()**

Is there current a session attached to this MediaCaster

**Returns:**

true is MediaCaster has session

(continued from last page)

## isStream

```
public boolean isStream()
```

Is there a stream associated with this MediaCaster

**Returns:**

true if stream associated with this MediaCaster

---

## getConnectLastAttempt

```
public long getConnectLastAttempt()
```

Get system time in milliseconds of last connection attempt

**Returns:**

system time in milliseconds of last connection attempt

---

## getConnectLastSuccess

```
public long getConnectLastSuccess()
```

Get system time in milliseconds of last connection success

**Returns:**

system time in milliseconds of last connection success

---

## getConnectLastForceReset

```
public long getConnectLastForceReset()
```

Get system time in milliseconds of last time forceReset was called

**Returns:**

system time in milliseconds of last time forceReset was called

---

## isStreamIsRunning

```
public boolean isStreamIsRunning()
```

Return true if stream is currently running

**Returns:**

true if stream is currently running

---

## getReconnectWaitTime

```
public int getReconnectWaitTime()
```

Get the minimum time between reconnect attempts (milliseconds)

**Returns:**

minimum time between reconnect attempts (milliseconds)

---

## setReconnectWaitTime

```
public void setReconnectWaitTime(int reconnectWaitTime)
```

---

(continued from last page)

Set the minimum time between reconnect attempts (milliseconds)

**Parameters:**

reconnectWaitTime - minimum time between reconnect attempts (milliseconds)

---

## getMediaCasterType

```
public int getMediaCasterType()
```

Get the media caster type. See IMediaCaster.MEDIACASTERTYPE\_\*

**Returns:**

media caster type

---

## setMediaCasterType

```
public void setMediaCasterType(int mediaCasterType)
```

Set the media caster type. See IMediaCaster.MEDIACASTERTYPE\_\*

**Parameters:**

mediaCasterType - media caster type

---

## getLock

```
public Object getLock()
```

Get the MediaCaster lock for an individual MediaCaster item.

---

## setStreamRunning

```
public void setStreamRunning(boolean streamRunning)
```

Called internally to track when streams are running or are stopped.

**Parameters:**

streamRunning - set to true if stream is running

---

## getStreamRunningIndex

```
public long getStreamRunningIndex()
```

Each time a MediaCaster stream switches from not running to running this value is incremented

**Returns:**

stream running index

---

## com.wowza.wms.medicaster Interface IMediaCasterDataReceiver

---

public interface **IMediaCasterDataReceiver**  
extends

IMediaCasterDataReceiver: For internal use only.

---

### Method Summary

void	<a href="#"><u>onData</u></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="#"><u>getNetConnection()</u></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(<i>IMediaCaster</i> mediaCaster)</a> Invoked when mediaCaster created
void	<a href="#">onMediaCasterDestroy(<i>IMediaCaster</i> mediaCaster)</a> Invoked when MediaCaster destroyed
void	<a href="#">onRegisterPlayer(<i>IMediaCaster</i> mediaCaster, <i>IMediaStreamPlay</i> player)</a> Invoked when a player is added to this mediaCaster
void	<a href="#">onSetSourceStream(<i>IMediaCaster</i> mediaCaster, <i>IMediaStream</i> stream)</a> Invoked when soure stream is set (can be called with stream of null)
void	<a href="#">onUnRegisterPlayer(<i>IMediaCaster</i> mediaCaster, <i>IMediaStreamPlay</i> player)</a> 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 soure 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.medicaster

### Class MediaCasterItem

```
java.lang.Object
+-com.wowza.wms.medicaster.MediaCasterItem
```

---

```
public class MediaCasterItem
extends Object
```

---

### Constructor Summary

public	<a href="#">MediaCasterItem(String name, String streamType, String baseClass)</a>
	Media caster item constructor

### Method Summary

void	<a href="#">clearProperty(String name)</a> Clear property
String	<a href="#">getBaseClass()</a> Get base class
int	<a href="#">getConnectionTimeout()</a> Get connection timeout (milliseconds)
String	<a href="#">getDescription()</a> Get description
static String	<a href="#">getIdString(String name, String liveStreamPacketizer, String liveStreamRepeater)</a> Get id string for this media caster item (not used - returns name unchanged)
int	<a href="#">getKeepAliveTime()</a> Get keep alive time (milliseconds)
String	<a href="#">getName()</a> Get name
<a href="#">WMSProperties</a>	<a href="#">getProperties()</a> Get properties collection
String	<a href="#">getProperty(String name)</a> Get property value
String	<a href="#">getStreamType()</a> Get the stream type
String	<a href="#">idStringToName(String idString)</a> Convert name to id string (not used - returns id string unchanged)
static MediaCasterStreamId	<a href="#">parseIdString(String idString)</a>

void	<a href="#"><u>setBaseClass</u></a> (String baseClass) Set base class
void	<a href="#"><u>setConnectionTimeout</u></a> (int connectionTimeout) Set connection timeout (milliseconds)
void	<a href="#"><u>setDescription</u></a> (String description) Set description
void	<a href="#"><u>setKeepAliveTime</u></a> (int keepAliveTime) Set keep alive time (milliseconds)
void	<a href="#"><u>setName</u></a> (String name) Set name
void	<a href="#"><u>setProperty</u></a> (String name, String value) Set a property
void	<a href="#"><u>setStreamType</u></a> (String streamType) Set stream type
String	<a href="#"><u>toString</u></a> () toString

**Methods inherited from class java.lang.Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, 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="#">MediaCasterList()</a>
	Constructor

### Method Summary

<a href="#">MediaCasterItem</a>	<a href="#">getMediaCasterDef(String name)</a> Get media caster definition by name
java.util.Map	<a href="#">getMediaCasterDefs()</a> Get map of media caster items
java.util.List	<a href="#">getMediaCasterNames()</a> Get list of media caster names

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, 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.medicaster Class MediaCasterSettings

```
java.lang.Object
+-com.wowza.wms.medicaster.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(HostPortConfig mediaCasterHostPortConfig)</a> Set media caster host port config
void	<a href="#">setMediaCasterProcessorCount(int mediaCasterProcessorCount)</a> 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="#">MediaCasterStreamItem(long uniqueId)</a>
	Media caster item constructor

### Method Summary

void	<a href="#">acquire()</a> Increment acquire lock count for this media caster item
void	<a href="#">acquireAndRelease()</a> Increment then decrement acquire lock count for this media caster item
boolean	<a href="#">doWatchdog()</a> Do idle processing
String	<a href="#">getLiveStreamPacketizer()</a> Get the live stream packetizer for this media caster stream item
String	<a href="#">getLiveStreamRepeater()</a> Get the live stream repeater for this media caster stream item
Object	<a href="#">getLock()</a>
int	<a href="#">getLockCount()</a> Get the current number of acquire locks on this media caster item
<a href="#">IMediaCaster</a>	<a href="#">getMediaCaster()</a> Get the underlying IMediaCaster interface for this MediaCaster
String	<a href="#">getMediaCasterId()</a> Get this media caster item id
int	<a href="#">getPlayerCount()</a> Get the current number of players associated with this media caster item
String	<a href="#">getStreamExt()</a>
long	<a href="#">getUniqueId()</a>

void	<a href="#"><u>init</u></a> (String mediaCasterId, String streamExt, <a href="#"><u>MediaCasterItem</u></a> mediaCasterDef, <a href="#"><u>MediaCasterStreamMap</u></a> parent, String liveStreamPacketizer, String liveStreamRepeater) Initialize the media caster item (internal use)
boolean	<a href="#"><u>isShutdownOnRelease</u></a> () On last release shutdown the stream even if clients are connected
boolean	<a href="#"><u>isValid</u></a> ()
void	<a href="#"><u>registerPlayer</u></a> ( <a href="#"><u>IMediaStreamPlay</u></a> player) Register a player with a media caster item (internal use)
void	<a href="#"><u>release</u></a> () Decrement acquire lock count for this media caster item
void	<a href="#"><u>reset</u></a> () Force a reconnect or reset for this media caster item
void	<a href="#"><u>setLiveStreamPacketizer</u></a> (String liveStreamPacketizer) Set the live stream packetizer for this media caster stream item
void	<a href="#"><u>setLiveStreamRepeater</u></a> (String liveStreamRepeater) Set the live stream repeater for this media caster stream item
void	<a href="#"><u>setShutdownOnRelease</u></a> (boolean shutdownOnRelease) On last release shutdown the stream even if clients are connected
void	<a href="#"><u>setStreamExt</u></a> (String streamExt)
void	<a href="#"><u>setValid</u></a> (boolean isValid)
void	<a href="#"><u>shutdown</u></a> (boolean isAppInstanceShutdown) Shutdown this media caster item
void	<a href="#"><u>unregisterPlayer</u></a> ( <a href="#"><u>IMediaStreamPlay</u></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

## 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	
--	--

### 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(IApplicationInstance appInstance)</a>
	Create a new mediacaster map

### Method Summary

<a href="#">MediaCasterStreamItem</a>	<a href="#">acquire(String streamName)</a> 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(String inStreamName, MediaCasterItem mediaCasterDef)</a> 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(String inStreamName, String streamType)</a> 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(String streamName, String liveStreamPacketizer, String liveStreamRepeater)</a> 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(String inStreamName, String liveStreamPacketizer, String liveStreamRepeater, MediaCasterItem mediaCasterDef)</a> 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(String inStreamName, String liveStreamPacketizer, String liveStreamRepeater, String streamType)</a> 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(String inStreamName, String liveStreamPacketizer, String liveStreamRepeater, MediaCasterItem mediaCasterDef)</a> 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(long streamSrc, String mediaCasterId)</a>
void	<a href="#">clearStreamSrcToMediaCaster(long streamSrc)</a>
void	<a href="#">doWatchdog()</a> Do periodic idle time processing

<a href="#">IApplicationInstance</a>	<a href="#"><code>getApplicationInstance()</code></a> Get the parent application instance for this map
<code>edu.emory.mathcs.backport.java.util.concurrent.locks.WMSReadWriteLock</code>	<a href="#"><code>getLock()</code></a> Get the read/write lock for this interface
<a href="#">MediaCasterStreamItem</a>	<a href="#"><code>getMediaCaster(String streamName)</code></a> Get media caster item based on given stream name
<a href="#">MediaCasterStreamItem</a>	<a href="#"><code>getMediaCaster(String streamName, String liveStreamPacketizer, String liveStreamRepeater)</code></a> Get media caster item based on given stream name
int	<a href="#"><code>getMediaCasterCount()</code></a> Get the number of mediacasters current running
<code>java.util.List</code>	<a href="#"><code>getMediaCasterNames()</code></a> Get a list of all the currently running media caster names
<code>java.util.List</code>	<a href="#"><code>getMediaCasterStreamItems()</code></a>
<a href="#">MediaCasterStreamManager</a>	<a href="#"><code>getStreamManager()</code></a> Get the stream manager interface for managing the starting and stopping of streams
void	<a href="#"><code>registerPlayer(IMediaStreamPlay player, MediaCasterItem mediaCasterDef)</code></a> Register a player to a media caster item (internal use)
void	<a href="#"><code>release(MediaCasterStreamItem mediaCasterStreamItem)</code></a> Decrement lock count on media caster item
void	<a href="#"><code>release(MediaCasterStreamItem mediaCasterStreamItem, boolean removeIfZero)</code></a>
void	<a href="#"><code>remove(MediaCasterStreamItem mediaCasterStreamItem)</code></a>
void	<a href="#"><code>shutdown(boolean isAppInstanceShutdown)</code></a> Shutdown this media caster and close all running media casters
String	<a href="#"><code>streamSrcToMediaCaster(long streamSrc)</code></a>
void	<a href="#"><code>unregisterPlayer(IMediaStreamPlay player, MediaCasterItem mediaCasterDef)</code></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`

## Constructors

### MediaCasterStreamMap

```
public MediaCasterStreamMap(IApplicationInstance appInstance)
```

(continued from last page)

Create a new medicaster map

**Parameters:**

appInstance - application instance

## Methods

### getLock

```
public edu.emory.mathcs.backport.java.util.concurrent.locks.WMSReadWriteLock getLock()
```

Get the read/write lock for this interface

**Returns:**

read/write lock

### getStreamManager

```
public MediaCasterStreamManager getStreamManager()
```

Get the stream manager interface for managing the starting and stopping of streams

**Returns:**

stream manager interface

### getApplicationInstance

```
public IApplicationInstance getApplicationInstance()
```

Get the parent application instance for this map

**Returns:**

parent application instance for this map

### getMediaCasterCount

```
public int getMediaCasterCount()
```

Get the number of mediacasters current running

**Returns:**

number of mediacasters current running

### getMediaCasterStreamItems

```
public java.util.List getMediaCasterStreamItems()
```

### doWatchdog

```
public void doWatchdog()
```

Do periodic idle time processing

(continued from last page)

## shutdown

```
public void shutdown(boolean isAppInstanceShutdown)
```

Shutdown this media caster and close all running media casters

**Parameters:**

isAppInstanceShutdown - is this due to application shutdown

---

## getMediaCasterNames

```
public java.util.List getMediaCasterNames()
```

Get a list of all the currently running media caster names

**Returns:**

list of all the currently running media caster names

---

## getMediaCaster

```
public MediaCasterStreamItem getMediaCaster(String streamName)
```

Get media caster item based on given stream name

**Parameters:**

streamName - stream name

**Returns:**

media caster item

---

## getMediaCaster

```
public MediaCasterStreamItem getMediaCaster(String streamName,
                                             String liveStreamPacketizer,
                                             String liveStreamRepeater)
```

Get media caster item based on given stream name

**Parameters:**

streamName - stream name

liveStreamPacketizer - live stream packetizer name

liveStreamRepeater - live stream repeater name

**Returns:**

media caster item

---

## acquire

```
public MediaCasterStreamItem acquire(String streamName,
                                       String liveStreamPacketizer,
                                       String liveStreamRepeater)
```

Increment the lock count for a media caster item (so that it remains loaded even if all listeners go away). Will load the media caster if not already loaded. Uses default application instance stream type.

**Parameters:**

streamName - stream name

liveStreamPacketizer - live stream packetizer name

liveStreamRepeater - live stream repeater name

(continued from last page)

**Returns:**

media caster item

**acquire**

```
public MediaCasterStreamItem acquire(String streamName)
```

Increment the lock count for a media caster item (so that it remains loaded even if all listeners go away). Will load the media caster if not already loaded. Uses default application instance stream type.

**Parameters:**

streamName - stream name

**Returns:**

media caster item

**acquire**

```
public MediaCasterStreamItem acquire(String inStreamName,  
        MediaCasterItem mediaCasterDef)
```

Increment the lock count for a media caster item (so that it remains loaded even if all listeners go away). Will load the media caster if not already loaded. This API allow loading of a MediaCaster into a non-MediaCaster stream type such as the "live" or "liverepeater-origin" stream type.

**Parameters:**

inStreamName - stream name

mediaCasterDef - MediaCaster definition

**Returns:**

media caster item

**acquire**

```
public MediaCasterStreamItem acquire(String inStreamName,  
        String liveStreamPacketizer,  
        String liveStreamRepeater,  
        MediaCasterItem mediaCasterDef)
```

Increment the lock count for a media caster item (so that it remains loaded even if all listeners go away). Will load the media caster if not already loaded. This API allow loading of a MediaCaster into a non-MediaCaster stream type such as the "live" or "liverepeater-origin" stream type.

**Parameters:**

inStreamName - stream name

liveStreamPacketizer - live stream packetizer name

liveStreamRepeater - live stream repeater name

mediaCasterDef - MediaCaster definition

**Returns:**

media caster item

**acquireAndRelease**

```
public MediaCasterStreamItem acquireAndRelease(String inStreamName,  
        String liveStreamPacketizer,  
        String liveStreamRepeater,  
        MediaCasterItem mediaCasterDef)
```

(continued from last page)

Increment and then decrement the lock count for a media caster item (so that it is loaded and will stay loaded for at least the KeepAliveTime). Will load the media caster if not already loaded. This API allow loading of a MediaCaster into a non-MediaCaster stream type such as the "live" or "liverepeater-origin" stream type.

**Parameters:**

inStreamName - stream name  
liveStreamPacketizer - live stream packetizer name  
liveStreamRepeater - live stream repeater name  
mediaCasterDef - MediaCaster definition

**Returns:**

media caster item

**acquire**

```
public MediaCasterStreamItem acquire(String inStreamName,  
String streamType)
```

Increment the lock count for a media caster item (so that it remains loaded even if all listeners go away). Will load the media caster if not already loaded.

**Parameters:**

inStreamName - stream name  
streamType - stream type to use (null will use default application streamType)

**Returns:**

media caster item

**acquire**

```
public MediaCasterStreamItem acquire(String inStreamName,  
String liveStreamPacketizer,  
String liveStreamRepeater,  
String streamType)
```

Increment the lock count for a media caster item (so that it remains loaded even if all listeners go away). Will load the media caster if not already loaded.

**Parameters:**

inStreamName - stream name  
liveStreamPacketizer - live stream packetizer name  
liveStreamRepeater - live stream repeater name  
streamType - stream type to use (null will use default application streamType)

**Returns:**

media caster item

**release**

```
public void release(MediaCasterStreamItem mediaCasterStreamItem)
```

Decrement lock count on media caster item

**Parameters:**

mediaCasterStreamItem - media caster item to decrement

**release**

```
public void release(MediaCasterStreamItem mediaCasterStreamItem,  
boolean removeIfZero)
```

(continued from last page)

---

**remove**

```
public void remove(MediaCasterStreamItem mediaCasterStreamItem)
```

---

**streamSrcToMediaCaster**

```
public String streamSrcToMediaCaster(long streamSrc)
```

---

**addStreamSrcToMediaCaster**

```
public void addStreamSrcToMediaCaster(long streamSrc,  
String mediaCasterId)
```

---

**clearStreamSrcToMediaCaster**

```
public void clearStreamSrcToMediaCaster(long streamSrc)
```

---

**registerPlayer**

```
public void registerPlayer(IMediaStreamPlay player,  
MediaCasterItem mediaCasterDef)
```

Register a player to a media caster item (internal use)

**Parameters:**

player - player to register  
mediaCasterDef - media caster definition

---

**unregisterPlayer**

```
public void unregisterPlayer(IMediaStreamPlay player,  
MediaCasterItem mediaCasterDef)
```

Unregister a player to a media caster item (internal use)

**Parameters:**

player - player to unregister  
mediaCasterDef - media caster definition

---

## com.wowza.wms.medicaster

### Class ModuleMediaCaster

```
java.lang.Object
+- com.wowza.wms.module.ModuleBase
  +- com.wowza.wms.medicaster.ModuleMediaCaster
```

#### 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(IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)</a> Increment the lock count of a media caster stream.
void	<a href="#">getLockCount(IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)</a> Get the current lock count for a stream
void	<a href="#">getPlayerCount(IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)</a> Get the numbers of players associated with a particular media caster
void	<a href="#">getStreamNames(IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)</a> Get an array of stream names that are media casters associate with this application instance (returned as AMFdataArray)
void	<a href="#">onAppStart(IApplicationInstance appInstance)</a> onAppStart

void	<code>onAppStop(IApplicationInstance appInstance)</code> onAppStop
void	<code>releaseMediaCaster(IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)</code> Decrement the lock count of a media caster stream
void	<code>resetStream(IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)</code> Reset a media caster stream
void	<code>shutdownStream(IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)</code> 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`,  
`sendStreamOnStatusError`

**Methods inherited from class** `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `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="#">doIdle(MediaCaster mediaCaster)</a>
RTPSessionDescription Data	<a href="#">getSessionDescriptionData(IApplicationInstance appInstance, String streamName, int retryCount, IRTPSessionDescriptionSessionHandler handler, IMediaCaster mediaCaster)</a>
void	<a href="#">init(MediaCaster mediaCaster)</a>
void	<a href="#">sessionStart(RTPSession rtpSession)</a>
void	<a href="#">sessionStop(RTPSession rtpSession)</a>

### 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="#"><u>onDisconnect()</u></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="#"><u>onCodecConfigAAC</u></a> (com.wowza.wms.media.aac.AACFrame frame, byte[] buffer, long offset)
void	<a href="#"><u>onFrameAAC</u></a> (com.wowza.wms.media.aac.AACFrame frame, byte[] buffer, long offset)
void	<a href="#"><u>onFrameMP3</u></a> (int frequency, int samplesPerFrame, int channels, byte[] syncHeader, byte[] packetHeader, byte[] frameData)
void	<a href="#"><u>onHeaderData</u></a> (java.util.Map headerMap)
void	<a href="#"><u>onMetaData</u></a> (java.util.Map metaMap)
void	<a href="#"><u>onTrim</u></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(<u>IShoutCastMetaDataNotify</u> listener)</a>
boolean	<a href="#">removeMetaDataListener(<u>IShoutCastMetaDataNotify</u> listener)</a>

### 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.medalist

### Class MediaList

```
java.lang.Object
+-com.wowza.wms.medalist.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>
protected	<a href="#">title</a>

#### Constructor Summary

public	<a href="#">MediaList()</a>
--------	-----------------------------

#### Method Summary

void	<a href="#">addSegment(int index, MediaListSegment mediaListSegment)</a>
void	<a href="#">addSegment(MediaListSegment mediaListSegment)</a>
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(boolean write)</a>
java.util.List	<a href="#">getSegment()</a>

<a href="#">MediaListSegment</a>	<a href="#">getSegment(int index)</a>
String	<a href="#">getTitle()</a>
<a href="#">MediaListSegment</a>	<a href="#">removeSegment(int index)</a>
void	<a href="#">removeSegment(MediaListSegment mediaListSegment)</a>
void	<a href="#">reset()</a>
void	<a href="#">setName(String name)</a>
void	<a href="#">setTitle(String title)</a>
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`

## Fields

**name**

```
protected java.lang.String name
```

**title**

```
protected java.lang.String title
```

**segments**

```
protected java.util.List segments
```

**properties**

```
protected com.wowza.wms.application.WMSProperties properties
```

(continued from last page)

**lock**

```
protected java.lang.Object lock
```

## Constructors

### MediaList

```
public MediaList()
```

## Methods

### toString

```
public String toString()
```

---

### toSMILString

```
public String toSMILString()
```

---

### reset

```
public void reset()
```

---

### getSegment

```
public java.util.List getSegment()
```

---

### addSegment

```
public void addSegment(MediaListSegment mediaListSegment)
```

---

### addSegment

```
public void addSegment(int index,  
                      MediaListSegment mediaListSegment)
```

---

### removeSegment

```
public void removeSegment(MediaListSegment mediaListSegment)
```

**removeSegment**

```
public MediaListSegment removeSegment(int index)
```

---

**clearSegments**

```
public void clearSegments()
```

---

**getFirstSegment**

```
public MediaListSegment getFirstSegment()
```

---

**getSegment**

```
public MediaListSegment getSegment(int index)
```

---

**getLock**

```
public Object getLock()
```

---

**getProperties**

```
public WMSProperties getProperties()
```

---

**getProperties**

```
public WMSProperties getProperties(boolean write)
```

---

**getName**

```
public String getName()
```

---

**setName**

```
public void setName(String name)
```

---

**getTitle**

```
public String getTitle()
```

---

(continued from last page)

---

### **setTitle**

```
public void setTitle(String title)
```

## com.wowza.wms.medialist Class MediaListRendition

```
java.lang.Object
+-com.wowza.wms.medialist.MediaListRendition
```

---

```
public class MediaListRendition
extends Object
```

---

### Field Summary

protected	<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="#">playSeek</a>
protected	<a href="#">playStart</a>
protected	<a href="#">properties</a>
protected	<a href="#">type</a>
protected	<a href="#">videoCodecId</a>
protected	<a href="#">vodTranscodeNGRP</a>
protected	<a href="#">width</a>
protected	<a href="#">wowzaAudioOnly</a>

protected	<a href="#">wowzaKeyFrameOnly</a>
protected	<a href="#">wowzaVideoOnly</a>

## Constructor Summary

public	<a href="#">MediaListRendition()</a>
--------	--------------------------------------

## Method Summary

String	<a href="#">getAudioCodecId()</a>
int	<a href="#">getBitrateAudio()</a>
int	<a href="#">getBitrateTotal()</a>
int	<a href="#">getBitrateVideo()</a>
String	<a href="#">getCodecId()</a>
int	<a href="#">getHeight()</a>
String	<a href="#">getLanguage()</a>
Object	<a href="#">getLock()</a>
<a href="#">MediaListSegment</a>	<a href="#">getMediaListSegment()</a>
String	<a href="#">getName()</a>
long	<a href="#">getPlayDuration()</a>
long	<a href="#">getPlaySeek()</a>
long	<a href="#">getPlayStart()</a>
<a href="#">WMSProperties</a>	<a href="#">getProperties()</a>
<a href="#">WMSProperties</a>	<a href="#">getProperties(boolean write)</a>
int[]	<a href="#">getSize()</a>
String	<a href="#">getTitle()</a>
int	<a href="#">getType()</a>
String	<a href="#">getTypeAsString()</a>

String	<a href="#">getVideoCodecId()</a>
String	<a href="#">getVODtranscodeNGRP()</a>
int	<a href="#">getWidth()</a>
boolean	<a href="#">isWowzaAudioOnly()</a>
boolean	<a href="#">isWowzaKeyFrameOnly()</a>
boolean	<a href="#">isWowzaVideoOnly()</a>
static java.util.ArrayList	<a href="#">parseCommaSeparatedLanguages(String langStr)</a>
void	<a href="#">setAudioCodecId(String audioCodecId)</a>
void	<a href="#">setBitrateAudio(int bitrateAudio)</a>
void	<a href="#">setBitrateTotal(int bitrateTotal)</a>
void	<a href="#">setBitrateVideo(int bitrateVideo)</a>
void	<a href="#">setHeight(int height)</a>
void	<a href="#">setLanguage(String language)</a>
void	<a href="#">setLock(Object lock)</a>
void	<a href="#">setMediaListSegment(MediaListSegment mediaListSegment)</a>
void	<a href="#">setName(String name)</a>
void	<a href="#">setPlayDuration(long playDuration)</a>
void	<a href="#">setPlaySeek(long playSeek)</a>
void	<a href="#">setPlayStart(long playStart)</a>
void	<a href="#">setSize(int width, int height)</a>
void	<a href="#">setTitle(String title)</a>
void	<a href="#">setType(int type)</a>
void	<a href="#">setVideoCodecId(String videoCodecId)</a>
void	<a href="#">setVODTranscodeNGRP(String vodTranscodeNGRP)</a>

void	<a href="#"><u>setWidth(int width)</u></a>
void	<a href="#"><u>setWowzaAudioOnly(boolean wowzaAudioOnly)</u></a>
void	<a href="#"><u>setWowzaKeyFrameOnly(boolean wowzaKeyFrameOnly)</u></a>
void	<a href="#"><u>setWowzaVideoOnly(boolean wowzaVideoOnly)</u></a>
String	<a href="#"><u>toSMILString()</u></a>
String	<a href="#"><u>toString()</u></a>

**Methods inherited from class java.lang.Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, 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**

(continued from last page)

---

**bitrateAudio**

```
protected int bitrateAudio
```

---

**bitrateVideo**

```
protected int bitrateVideo
```

---

**videoCodecId**

```
protected java.lang.String videoCodecId
```

---

**audioCodecId**

```
protected java.lang.String audioCodecId
```

---

**wowzaAudioOnly**

```
protected boolean wowzaAudioOnly
```

---

**wowzaVideoOnly**

```
protected boolean wowzaVideoOnly
```

---

**wowzaKeyFrameOnly**

```
protected boolean wowzaKeyFrameOnly
```

---

**width**

```
protected int width
```

---

**height**

```
protected int height
```

---

(continued from last page)

**playStart**

```
protected long playStart
```

---

**playDuration**

```
protected long playDuration
```

---

**playSeek**

```
protected long playSeek
```

---

**vodTranscodeNGRP**

```
protected java.lang.String vodTranscodeNGRP
```

**Constructors****MediaListRendition**

```
public MediaListRendition()
```

**Methods****toString**

```
public String toString()
```

---

**toSMILString**

```
public String toSMILString()
```

---

**getTypeAsString**

```
public String getTypeAsString()
```

---

**getLock**

```
public Object getLock()
```

**setLock**

```
public void setLock(Object lock)
```

---

**getMediaListSegment**

```
public MediaListSegment getMediaListSegment()
```

---

**setMediaListSegment**

```
public void setMediaListSegment(MediaListSegment mediaListSegment)
```

---

**getProperties**

```
public WMSProperties getProperties()
```

---

**getProperties**

```
public WMSProperties getProperties(boolean write)
```

---

**getBitrateTotal**

```
public int getBitrateTotal()
```

---

**setBitrateTotal**

```
public void setBitrateTotal(int bitrateTotal)
```

---

**getBitrateAudio**

```
public int getBitrateAudio()
```

---

**setBitrateAudio**

```
public void setBitrateAudio(int bitrateAudio)
```

---

**getBitrateVideo**

```
public int getBitrateVideo()
```

---

(continued from last page)

---

**setBitrateVideo**

```
public void setBitrateVideo(int bitrateVideo)
```

---

**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()
```

---

(continued from last page)

**setName**

```
public void setName(String name)
```

---

**getType**

```
public int getType()
```

---

**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()
```

(continued from last page)

---

**setPlayStart**

```
public void setPlayStart(long playStart)
```

---

**getPlayDuration**

```
public long getPlayDuration()
```

---

**setPlayDuration**

```
public void setPlayDuration(long playDuration)
```

---

**getPlaySeek**

```
public long getPlaySeek()
```

---

**setPlaySeek**

```
public void setPlaySeek(long playSeek)
```

---

**getVODTranscodeNGRP**

```
public String getVODTranscodeNGRP()
```

---

**setVODTranscodeNGRP**

```
public void setVODTranscodeNGRP(String vodTranscodeNGRP)
```

---

**setLanguage**

```
public void setLanguage(String language)
```

---

**getLanguage**

```
public String getLanguage()
```

---

(continued from last page)

**parseCommaSeparatedLanguages**

```
public static java.util.ArrayList parseCommaSeparatedLanguages(String langStr)
```

---

**isWowzaVideoOnly**

```
public boolean isWowzaVideoOnly()
```

---

**setWowzaVideoOnly**

```
public void setWowzaVideoOnly(boolean wowzaVideoOnly)
```

---

**isWowzaKeyFrameOnly**

```
public boolean isWowzaKeyFrameOnly()
```

---

**setWowzaKeyFrameOnly**

```
public void setWowzaKeyFrameOnly(boolean wowzaKeyFrameOnly)
```

---

**getTitle**

```
public String getTitle()
```

---

**setTitle**

```
public void setTitle(String title)
```

---

## com.wowza.wms.medialist Class MediaListSegment

```
java.lang.Object
+-com.wowza.wms.medialist.MediaListSegment
```

---

```
public class MediaListSegment
extends Object
```

### Field Summary

protected	<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(int index, MediaListRendition mediaListRendition)</a>
void	<a href="#">addRendition(MediaListRendition mediaListRendition)</a>
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(boolean write)</a>
<a href="#">MediaListRendition</a>	<a href="#">getRendition(int index)</a>
java.util.List	<a href="#">getRenditions()</a>

<a href="#">MediaListRendition</a>	<a href="#">removeRendition(int index)</a>
void	<a href="#">removeRendition(MediaListRendition mediaListRendition)</a>
void	<a href="#">setLock(Object lock)</a>
void	<a href="#">setMediaList(MediaList mediaList)</a>
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

## 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	<a href="#"><u>onResult</u></a> ( <a href="#"><u>IClient</u></a> client, com.wowza.wms.request.RequestFunction function, <a href="#"><u>AMFDataList</u></a> params)
	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  
IAplicationInstance.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 Subinterfaces:

[IModuleOnApp2](#)

All Known Implementing Classes:

[ModuleMediaCaster](#)

public interface **IModuleOnApp**

extends

IModuleCallResult: method interface examples for application level methods in a module.

Since module method must be implemented as static method a module cannot directly implements this interface. This interface only serves as an example of the method names and call signature needed to implement these application methods.

### Method Summary

void	<a href="#">onAppStart(<i>IApplicationInstance</i> appInstance)</a> Invoked when an application instance is started.
void	<a href="#">onAppStop(<i>IApplicationInstance</i> appInstance)</a> Invoked when an application instance is stopped (destroyed).

### Methods

#### onAppStart

public void [onAppStart\(\*IApplicationInstance\* appInstance\)](#)

Invoked when an application instance is started.

**Parameters:**

appInstance - application instance

#### onAppStop

public void [onAppStop\(\*IApplicationInstance\* appInstance\)](#)

Invoked when an application instance is stopped (destroyed).

**Parameters:**

appInstance - application instance

## com.wowza.wms.module Interface IModuleOnApp2

All Superinterfaces:

[IModuleOnApp](#)

---

public interface **IModuleOnApp2**

extends [IModuleOnApp](#)

### Method Summary

void	<a href="#">onAppCreate</a> ( <a href="#">IApplicationInstance</a> appInstance) Invoked when an application instance is created.
void	<a href="#">onAppDestroy</a> ( <a href="#">IApplicationInstance</a> appInstance) Invoked when an application instance is destroyed

Methods inherited from interface [com.wowza.wms.module.IModuleOnApp](#)

[onAppStart](#), [onAppStop](#)

### Methods

#### onAppCreate

public void **onAppCreate**([IApplicationInstance](#) appInstance)

Invoked when an application instance is created.

**Parameters:**

appInstance - application instance

---

#### onAppDestroy

public void **onAppDestroy**([IApplicationInstance](#) appInstance)

Invoked when an application instance is destroyed

**Parameters:**

appInstance - application instance

## com.wowza.wms.module Interface IModuleOnCall

public interface IModuleOnCall  
extends

IModuleOnCall: method interface example for the catch-all method handler onCall.

Since module method must be implemented as static method a module cannot directly implements this interface. This interface only serves as an example of the method name and call signature needed to implement this method. The onCall method, when defined in a module, is invoked for all handlers that are undefined in a given module. The onCall handler can also be used to catch calls to server side component calls.

### Method Summary

void	<a href="#"><b>onCall</b></a> (String handlerName, <a href="#">IClient</a> client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> 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(IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)</code></a>
	Invoked when a client connection is initiated.
void	<a href="#"><code>onConnectAccept(IClient client)</code></a>
	Invoked when a client connection is accepted.
void	<a href="#"><code>onConnectReject(IClient client)</code></a>
	Invoked when a client connection is rejected.
void	<a href="#"><code>onDisconnect(IClient client)</code></a>
	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="#">KEYDATA_MODE_ENCRYPT</a>
	Value: 3
public static final	<a href="#">KEYDATA_MODE_INIT</a>
	Value: 1
public static final	<a href="#">KEYDATA_MODE_KEYFRAME_ENCRYPT</a>
	Value: 6
public static final	<a href="#">KEYDATA_MODE_KEYFRAME_INIT</a>
	Value: 4
public static final	<a href="#">KEYDATA_MODE_KEYFRAME_PLAYLIST</a>
	Value: 5
public static final	<a href="#">KEYDATA_MODE_PLAYLIST</a>
	Value: 2

### Method Summary

void	<a href="#">onHTTPCupertinoEncryptionKeyCreateLive</a> ( <a href="#">IApplicationInstance</a> appInstance, String streamName, byte[] encKey) Called when live stream key is requested (per-published stream)
void	<a href="#">onHTTPCupertinoEncryptionKeyCreateVOD</a> ( <a href="#">HTTPStreamerSessionCupertino</a> httpSession, byte[] encKey) Called when video on demand key is requested (per-session)
void	<a href="#">onHTTPCupertinoEncryptionKeyData</a> ( <a href="#">HTTPStreamerSessionCupertino</a> httpSession, <a href="#">IHTTPRequest</a> req, <a href="#">IHTTPResponse</a> resp, byte[] encKeyData) Called when a key data is requested.
void	<a href="#">onHTTPCupertinoEncryptionKeyLiveChunk</a> ( <a href="#">ILiveStreamPacketizer</a> LiveStreamPacketizer, String streamName, CupertinoEncInfo encInfo, long chunkId, int mode) Called when live stream key is requested (per-published stream, per-chunk - for rotating keys)
void	<a href="#">onHTTPCupertinoEncryptionKeyRequest</a> ( <a href="#">HTTPStreamerSessionCupertino</a> httpSession, <a href="#">IHTTPRequest</a> req, <a href="#">IHTTPResponse</a> resp) Called when a key is requested.

void	<code>onHTTPCupertinoEncryptionKeyVODChunk( <a href="#">HTTPStreamerSessionCupertino</a> httpSession, <a href="#">IHTTPStreamerCupertinoIndex</a> index, <a href="#">CupertinoEncInfo</a> encInfo, long chunkId, int mode)</code>
------	---

Called when video on demand key is requested (per-session).

## Fields

### KEYDATA\_MODE\_INIT

```
public static final int KEYDATA_MODE_INIT
```

Constant value: 1

### KEYDATA\_MODE\_PLAYLIST

```
public static final int KEYDATA_MODE_PLAYLIST
```

Constant value: 2

### KEYDATA\_MODE\_ENCRYPT

```
public static final int KEYDATA_MODE_ENCRYPT
```

Constant value: 3

### KEYDATA\_MODE\_KEYFRAME\_INIT

```
public static final int KEYDATA_MODE_KEYFRAME_INIT
```

Constant value: 4

### KEYDATA\_MODE\_KEYFRAME\_PLAYLIST

```
public static final int KEYDATA_MODE_KEYFRAME_PLAYLIST
```

Constant value: 5

### KEYDATA\_MODE\_KEYFRAME\_ENCRYPT

```
public static final int KEYDATA_MODE_KEYFRAME_ENCRYPT
```

Constant value: 6

## Methods

### onHTTPCupertinoEncryptionKeyRequest

```
public void onHTTPCupertinoEncryptionKeyRequest( HTTPStreamerSessionCupertino  
httpSession,  
      IHTTPRequest req,  
      IHTTPResponse resp)
```

(continued from last page)

Called when a key is requested. Call httpSession.rejectSession to reject the streaming session and stop delivery of the encryption key.

**Parameters:**

- httpSession - HTTP session
- req - HTTP request
- resp - HTTP response

**onHTTPCupertinoEncryptionKeyData**

```
public void onHTTPCupertinoEncryptionKeyData(HTTPStreamerSessionCupertino httpSession,
    IHTTPRequest req,
    IHTTPResponse resp,
    byte[] encKeyData)
```

Called when a key data is requested. Set encKeyData to the key data to be sent to the client.

**Parameters:**

- httpSession - HTTP session
- req - request
- resp - response
- encKeyData - key data

**onHTTPCupertinoEncryptionKeyCreateVOD**

```
public void onHTTPCupertinoEncryptionKeyCreateVOD(HTTPStreamerSessionCupertino
    httpSession,
    byte[] encKey)
```

Called when video on demand key is requested (per-session)

**Parameters:**

- httpSession - HTTP session
- encKey - encryption key

**onHTTPCupertinoEncryptionKeyVODChunk**

```
public void onHTTPCupertinoEncryptionKeyVODChunk(HTTPStreamerSessionCupertino
    httpSession,
    IHTTPStreamerCupertinoIndex index,
    CupertinoEncInfo encInfo,
    long chunkId,
    int mode)
```

Called when video on demand key is requested (per-session). Allows setting of encryption key and URL.

**Parameters:**

- httpSession - HTTP session
- index - file index
- encInfo - encryption key
- mode - KEYDATA\_MODE\_\*

**onHTTPCupertinoEncryptionKeyCreateLive**

```
public void onHTTPCupertinoEncryptionKeyCreateLive(IApplicationInstance appInstance,
    String streamName,
    byte[] encKey)
```

Called when live stream key is requested (per-published stream)

(continued from last page)

**Parameters:**

appInstance - application instance  
streamName - stream name  
encKey - encryption key

---

## onHTTPCupertinoEncryptionKeyLiveChunk

```
public void onHTTPCupertinoEncryptionKeyLiveChunk(ILiveStreamPacketizer  
liveStreamPacketizer,  
String streamName,  
CupertinoEncInfo encInfo,  
long chunkId,  
int mode)
```

Called when live stream key is requested (per-published stream, per-chunk - for rotating keys)

**Parameters:**

liveStreamPacketizer - live stream packetizer  
streamName - stream name  
encInfo - encryption info  
chunkId - chunk ID, -1 for stream creation  
mode - KEYDATA\_MODE\_\*

## com.wowza.wms.module Interface IModuleOnHTTPCupertinoStreamingSession

public interface IModuleOnHTTPCupertinoStreamingSession  
extends

IModuleOnHTTPSession: method interface for HTTP Cupertino Streaming session create/destroy.

### Method Summary

void	<a href="#">onHTTPCupertinoStreamingSessionCreate</a> ( <a href="#">HTTPStreamerSessionCupertino</a> <a href="#">httpCupertinoStreamingSession</a> )
	Invoked when an HTTP Cupertino Streaming session is created.
void	<a href="#">onHTTPCupertinoStreamingSessionDestroy</a> ( <a href="#">HTTPStreamerSessionCupertino</a> <a href="#">httpCupertinoStreamingSession</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 IModuleOnHTTPMPMEGDashEncryption

public interface IModuleOnHTTPMPMEGDashEncryption  
extends

IModuleOnHTTPMPMEGDashEncryption: listener interface for listening to MPEG-DASH encryption events.

### Method Summary

void	<a href="#">onHTTPMPMEGDashEncryptionKeyLiveChunk</a> ( <a href="#">ILiveStreamPacketizer</a> LiveStreamPacketizer, String streamName, com.wowza.wms.drm.cenc.CencInfo cencInfo, long chunkId) Called when live stream key is requested (per-published stream)
void	<a href="#">onHTTPMPMEGDashEncryptionKeyVODChunk</a> ( <a href="#">HTTPStreamerSessionMPEGDash</a> httpSession, <a href="#">IHTTPStreamerMPEGDashIndex</a> index, com.wowza.wms.drm.cenc.CencInfo cencInfo, long chunkId) Called when video on demand key is requested (per-session).

### Methods

#### onHTTPMPMEGDashEncryptionKeyVODChunk

```
public void onHTTPMPMEGDashEncryptionKeyVODChunk(HTTPStreamerSessionMPEGDash  
httpSession,  
    IHTTPStreamerMPEGDashIndex index,  
    com.wowza.wms.drm.cenc.CencInfo cencInfo,  
    long chunkId)
```

Called when video on demand key is requested (per-session). Allows setting of encryption key and URL.

**Parameters:**

- httpSession - HTTP session
- index - file index
- cencInfo - Common Encryption information
- chunkId - chunk identifier

#### onHTTPMPMEGDashEncryptionKeyLiveChunk

```
public void onHTTPMPMEGDashEncryptionKeyLiveChunk(ILiveStreamPacketizer  
liveStreamPacketizer,  
    String streamName,  
    com.wowza.wms.drm.cenc.CencInfo cencInfo,  
    long chunkId)
```

Called when live stream key is requested (per-published stream)

**Parameters:**

- liveStreamPacketizer - live stream packetizer
- streamName - stream name
- cencInfo - Common Encryption information
- chunkId - chunk identifier

## com.wowza.wms.module Interface IModuleOnHTTPMPMEGDashStreamingSession

public interface IModuleOnHTTPMPMEGDashStreamingSession  
extends

IModuleOnHTTPMPMEGDashStreamingSession: method interface for HTTP MPEGDash Streaming session create/destroy.

### Method Summary

void	<a href="#">onHTTPMPMEGDashStreamingSessionCreate</a> ( <a href="#">HTTPStreamerSessionMPEGDash</a> <a href="#">httpMPMEGDashStreamingSession</a> )
void	<a href="#">onHTTPMPMEGDashStreamingSessionDestroy</a> ( <a href="#">HTTPStreamerSessionMPEGDash</a> <a href="#">httpMPMEGDashStreamingSession</a> )

### Methods

#### onHTTPMPMEGDashStreamingSessionCreate

public void [onHTTPMPMEGDashStreamingSessionCreate](#)([HTTPStreamerSessionMPEGDash](#)  
[httpMPMEGDashStreamingSession](#))

Invoked when an HTTP MPEGDash Streaming session is created.

**Parameters:**

[httpMPMEGDashStreamingSession](#) - httpMPMEGDashStreamingSession

#### onHTTPMPMEGDashStreamingSessionDestroy

public void [onHTTPMPMEGDashStreamingSessionDestroy](#)([HTTPStreamerSessionMPEGDash](#)  
[httpMPMEGDashStreamingSession](#))

Invoked when an HTTP MPEGDash Streaming session is destroyed.

**Parameters:**

[httpMPMEGDashStreamingSession](#) - httpMPMEGDashStreamingSession

## com.wowza.wms.module Interface IModuleOnHTTPSSanJoseStreamingSession

public interface IModuleOnHTTPSSanJoseStreamingSession  
extends

IModuleOnHTTPSSanJoseStreamingSession: method interface for HTTP SanJose Streaming session create/destroy.

### Method Summary

void	<a href="#">onHTTPSSanJoseStreamingSessionCreate</a> ( <a href="#">HTTPStreamerSessionSanJose</a> <a href="#">httpSanJoseStreamingSession</a> )
void	<a href="#">onHTTPSSanJoseStreamingSessionDestroy</a> ( <a href="#">HTTPStreamerSessionSanJose</a> <a href="#">httpSanJoseStreamingSession</a> )

Invoked when an HTTP SanJose Streaming session is created.

Invoked when an HTTP SanJose Streaming session is destroyed.

### Methods

#### onHTTPSSanJoseStreamingSessionCreate

public void [onHTTPSSanJoseStreamingSessionCreate](#)([HTTPStreamerSessionSanJose](#)  
[httpSanJoseStreamingSession](#))

Invoked when an HTTP SanJose Streaming session is created.

**Parameters:**

[httpSanJoseStreamingSession](#) - httpSanJoseStreamingSession

#### onHTTPSSanJoseStreamingSessionDestroy

public void [onHTTPSSanJoseStreamingSessionDestroy](#)([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(IHTTPStreamerSession httpSession)</a> Invoked when an HTTP Session is created (both Smooth and Cupertino sessions).
void	<a href="#">onHTTPSessionDestroy(IHTTPStreamerSession httpSession)</a> 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</a> ( <a href="#">HTTPStreamerSessionSmoothStreamer</a> httpSmoothStreamingSession)
void	<a href="#">onHTTPSmoothStreamingSessionDestroy</a> ( <a href="#">HTTPStreamerSessionSmoothStreamer</a> httpSmoothStreamingSession)

### Methods

#### onHTTPSmoothStreamingSessionCreate

public void **onHTTPSmoothStreamingSessionCreate**([HTTPStreamerSessionSmoothStreamer](#)  
httpSmoothStreamingSession)

Invoked when an HTTP Smooth Streaming session is created.

**Parameters:**

httpSmoothStreamingSession - httpSmoothStreamingSession

#### onHTTPSmoothStreamingSessionDestroy

public void **onHTTPSmoothStreamingSessionDestroy**([HTTPStreamerSessionSmoothStreamer](#)  
httpSmoothStreamingSession)

Invoked when an HTTP Smooth Streaming session is destroyed.

**Parameters:**

httpSmoothStreamingSession - httpSmoothStreamingSession

## com.wowza.wms.module Interface IModuleOnPushPublish

public interface IModuleOnPushPublish  
extends

### Method Summary

void	<a href="#">onPushPublishSessionCreate</a> (String streamName, <a href="#">IMediaStream</a> stream, <a href="#">IPushPublishSession</a> pushPublishSession)
	Invoked when push publish session is created
void	<a href="#">onPushPublishSessionDestroy</a> (String streamName, <a href="#">IMediaStream</a> stream, <a href="#">IPushPublishSession</a> pushPublishSession)
	Invoked when push publish session is destroyed

### Methods

#### onPushPublishSessionCreate

```
public void onPushPublishSessionCreate(String streamName,
IMediaStream stream,
IPushPublishSession pushPublishSession)
```

Invoked when push publish session is created

**Parameters:**

- streamName - stream name
- stream - stream
- pushPublishSession - push publish session

#### onPushPublishSessionDestroy

```
public void onPushPublishSessionDestroy(String streamName,
IMediaStream stream,
IPushPublishSession pushPublishSession)
```

Invoked when push publish session is destroyed

**Parameters:**

- streamName - stream name
- stream - stream
- pushPublishSession - push publish session

## com.wowza.wms.module Interface IModuleOnRTPSession

public interface IModuleOnRTPSession  
extends

IModuleOnRTPSession: method interface for RTP session create/destroy.

### Method Summary

void	<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="#"><u>onResult</u></a> ( <a href="#">IClient</a> client, long pingTime, int pingId, boolean result) Triggered on client side result from call to IClient.ping.
------	--

### Methods

#### onResult

```
public void onResult(IClient client,  
    long pingTime,  
    int pingId,  
    boolean result)
```

Triggered on client side result from call to IClient.ping. This method will be called on a success or failure (ping timeout) of a ping.

##### Parameters:

client - client  
pingTime - time in server time (milliseconds) when the ping was initiated  
pingId - internal server id for the ping request  
result - true if 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	<u>CALLBACK_PARAM8</u> Callback param: param8 Value: <b>8</b>
public static final	<u>CALLBACK_PARAM9</u> Callback param: param9 Value: <b>9</b>
public static final	<u>PARAM1</u> Method param: param1 Value: <b>3</b>
public static final	<u>PARAM10</u> Method param: param10 Value: <b>12</b>
public static final	<u>PARAM2</u> Method param: param2 Value: <b>4</b>
public static final	<u>PARAM3</u> Method param: param3 Value: <b>5</b>
public static final	<u>PARAM4</u> Method param: param4 Value: <b>6</b>
public static final	<u>PARAM5</u> Method param: param5 Value: <b>7</b>
public static final	<u>PARAM6</u> Method param: param6 Value: <b>8</b>
public static final	<u>PARAM7</u> Method param: param7 Value: <b>9</b>
public static final	<u>PARAM8</u> Method param: param8 Value: <b>10</b>
public static final	<u>PARAM9</u> Method param: param9 Value: <b>11</b>
public static final	<u>PARAMMETHODNAME</u> Method: method name Value: <b>0</b>
public static final	<u>PLAYTRANSITION_APPEND</u> Value: <b>0</b>

public static final	<a href="#">PLAYTRANSITION_APPEND_IMMEDIATE</a> Value: <b>2</b>
public static final	<a href="#">PLAYTRANSITION_RESET</a> Value: <b>1</b>
public static final	<a href="#">PLAYTRANSITION_RESET_IMMEDIATE</a> Value: <b>3</b>
public static final	<a href="#">PLAYTRANSITION_STOP</a> Value: <b>10</b>
public static final	<a href="#">PLAYTRANSITION_SWAP</a> Value: <b>12</b>
public static final	<a href="#">PLAYTRANSITION_SWITCH</a> Value: <b>13</b>
public static final	<a href="#">PLAYTRANSITION_UNKNOWN</a> Value: <b>14</b>
public static final	<a href="#">PLAYTRANSITIONSTR_APPEND</a> Play2 transition: APPEND Value: <b>append</b>
public static final	<a href="#">PLAYTRANSITIONSTR_RESET</a> Play2 transition: RESET Value: <b>reset</b>
public static final	<a href="#">PLAYTRANSITIONSTR_STOP</a> Play2 transition: STOP Value: <b>stop</b>
public static final	<a href="#">PLAYTRANSITIONSTR_SWAP</a> Play2 transition: SWAP Value: <b>swap</b>
public static final	<a href="#">PLAYTRANSITIONSTR_SWITCH</a> Play2 transition: SWITCH Value: <b>switch</b>
public static final	<a href="#">PLAYTRANSITIONSTR_UNKNOWN</a> Play2 transition: UNKNOWN Value: <b>unknown</b>

## Constructor Summary

public [ModuleBase\(\)](#)

## Method Summary

static <a href="#">IAplicationInstance</a>	<a href="#">getAppInstance(IClient client)</a> Get applicationInstace of a client.
static <a href="#">IAplication</a>	<a href="#">getApplication(IClient client)</a> Get application of a client.
static int	<a href="#">getCallbackParamCount(AMFDataList params)</a> 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(AMFDataList params, int index)</a> Get parameter by index.
static boolean	<a href="#">getParamBoolean(AMFDataList params, int index)</a> Get parameter by index, Return as boolean.
static boolean	<a href="#">getParamBoolean(AMFDataList params, int index, boolean defaultValue)</a> Get parameter by index, Return as boolean.
static int	<a href="#">getParamCount(AMFDataList params)</a> Get the total number of parameters passed to method.
static java.util.Date	<a href="#">getParamDate(AMFDataList params, int index)</a> Get parameter by index, Return as Date.
static double	<a href="#">getParamDouble(AMFDataList params, int index)</a> Get parameter by index, Return as double.
static double	<a href="#">getParamDouble(AMFDataList params, int index, double defaultValue)</a> Get parameter by index, Return as double.
static int	<a href="#">getParamInt(AMFDataList params, int index)</a> Get parameter by index, Return as int.
static int	<a href="#">getParamInt(AMFDataList params, int index, int defaultValue)</a> Get parameter by index, Return as int.
static long	<a href="#">getParamLong(AMFDataList params, int index)</a> Get parameter by index, Return as long.
static long	<a href="#">getParamLong(AMFDataList params, int index, long defaultValue)</a> Get parameter by index, Return as long.
static <a href="#">AMFDataMixedArray</a>	<a href="#">getParamMixedArray(AMFDataList params, int index)</a> Get parameter by index, Return as AMFDataMixedArray.
static <a href="#">AMFDataObj</a>	<a href="#">getParamObj(AMFDataList params, int index)</a> Get parameter by index, Return as Object.
static String	<a href="#">getParamString(AMFDataList params, int index)</a> Get parameter by index, Return as String.
static String	<a href="#">getParamString(AMFDataList params, int index, String defaultValue)</a> Get parameter by index, Return as String.
static int	<a href="#">getParamType(AMFDataList params, int index)</a> 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,

**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\_PARAMS5

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  
defaultValue - default value

**Returns:**

parameter value as String object, defaultValue 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 defaultValue)
```

Get parameter by index, Return as int.

**Parameters:**

params - parameters  
index - parameter index  
defaultValue - default value

**Returns:**

parameter value as int, defaultValue 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 defaultValue)
```

Get parameter by index, Return as double.

**Parameters:**

params - parameters  
index - parameter index  
defaultValue - 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 defaultValue)
```

Get parameter by index, Return as long.

**Parameters:**

params - parameters  
index - parameter index  
defaultValue - default value

**Returns:**

parameter value as long, defaultValue 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 applicationInstace 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
+- com.wowza.wms.module.ModuleBase
  +- com.wowza.wms.module.ModuleClientLogging
```

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(IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)</a> Send a debug message to the logging system NetConnection.call("logDebug", null, message);
static void	<a href="#">logError(IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)</a> Send a error message to the logging system NetConnection.call("logError", null, message);
static void	<a href="#">logInfo(IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)</a> Send a info message to the logging system NetConnection.call("logInfo", null, message);
static void	<a href="#">logWarn(IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)</a> 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`

**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(IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)</a> Server side implementation of NetStream.close();
static void	<a href="#">createStream(IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)</a> Create new server side NetStream object (internal to Flash workings).
static void	<a href="#">deleteStream(IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)</a> Delete server side stream object (internal to Flash workings).
static void	<a href="#">FCPublish(IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)</a> FCPublish method called by FME 2.5
static void	<a href="#">FCSubscribe(IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)</a> FCSubscribe subscribes to a live stream (if origin edge will start the stream from the origin to the edge)
static void	<a href="#">FCUnpublish(IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)</a>
static void	<a href="#">FCUnPublish(IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)</a> FCUnpublish method called by FME 2.5
static void	<a href="#">FCUnSubscribe(IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)</a> FCUnSubscribe to a live stream
static void	<a href="#">FCUnSubscribe(IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)</a>
static void	<a href="#">getClientID(IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)</a> Get the clientId for a client connection NetConnection.call("getClientID", resultObj);
static void	<a href="#">getLastStreamId(IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)</a> Get the id for the last created stream NetConnection.call("getLastStreamId", resultObj);
static void	<a href="#">getLiveStreamPacketizer(IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)</a> Get the live stream packetizer list for a client connection NetConnection.call("getLiveStreamPacketizer", resultObj);
static void	<a href="#">getPageUrl(IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)</a> getPageUrl returns the pageUrl from the onConnect metadata
static void	<a href="#">getReferrer(IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)</a> getReferrer returns the referrer from the onConnect metadata

static void	<code>getRepeaterOriginUrl(IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)</code> Get the Repeater Origin URL for this client
void	<code>getStreamBitrate(IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)</code> Get stream bitrate in bits-per-second for a video on demand stream (will not work for live).
static void	<code>getStreamLength(IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)</code> 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	<code>getStreamType(IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)</code> Get the default stream type for a client connection NetConnection.call("getStreamType", resultObj);
static void	<code>getVersion(IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)</code> Get the Wowza Pro server version and build number NetConnection.call("getVersion", resultObj);
static void	<code>initLiveStreamRepeating(IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)</code> Initialize a stream for live stream repeating
static void	<code>initStream(IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)</code> Create new server side NetStream object (internal to Flash workings).
static void	<code>pause(IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)</code> Server side implementation of NetStream.pause([ flag : Boolean]);
static void	<code>pauseRaw(IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)</code> pauseRaw method introduced in Flash player 10
static void	<code>play(IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)</code> Server side implementation of NetStream.play(name : Object [,start : Number[, len : Number[, reset : Object]]]);
static void	<code>play2(IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)</code> Server side implementation of NetStream.play(playOptions : NetStreamPlayOptions);
static void	<code>publish(IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)</code> Server side implementation of NetStream.publish(name : String [, howToPublish : String]);
static void	<code>receiveAudio(IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)</code> Server side implementation of NetStream.receiveAudio(receive : Boolean);
static void	<code>receiveVideo(IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)</code> Server side implementation of NetStream.receiveVideo(receive : Boolean   FPS : Number); FPS does not work the same as FMS.
static void	<code>releaseStream(IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)</code>

static void	<code>seek(IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)</code> Server side implementation of NetStream.seek(offset : Number);
static void	<code>setBandwidthLimit(IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)</code>
static void	<code>setBufferTime(IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)</code> Server side implementation of NetStream.setBufferTime(bufferTime : Number);
static void	<code>setLiveStreamPacketizer(IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)</code> Set the live stream packetizer for a stream
static void	<code>setRepeaterOriginUrl(IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)</code> Set the Repeater Origin URL for this client
static void	<code>setStreamType(IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)</code> Set the default stream type for a client connection NetConnection.call("setStreamType", null, streamType);
static void	<code>startTransmit(IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)</code>
static void	<code>stopTransmit(IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)</code>

**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`

## Constructors

### ModuleCore

```
public ModuleCore()
```

## Methods

(continued from last page)

## createStream

```
public static void createStream(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Create new server side NetStream object (internal to Flash workings).

**Parameters:**

- client - client
- function - function
- params - parameters (no params)

---

## initStream

```
public static void initStream(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Create new server side NetStream object (internal to Flash workings).

**Parameters:**

- client - client
- function - function
- params - parameters (streamIndex, boolean)

---

## releaseStream

```
public static void releaseStream(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

---

## getLastStreamId

```
public static void getLastStreamId(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Get the id for the last created stream NetConnection.call("getLastStreamId", resultObj);

**Parameters:**

- client - client
- function - client
- params - params (no params)

---

## deleteStream

```
public static void deleteStream(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Delete server side stream object (internal to Flash workings).

**Parameters:**

- client - client
- function - function
- params - params

## publish

```
public static void publish(IClient client,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

Server side implementation of NetStream.publish(name : String [, howToPublish : String]);

**Parameters:**

client - client  
function - function  
params - params (name [if name = "false" or "null" stop publishing], howToPublish ["record", "live", "append"])

---

## setBandwidthLimit

```
public static void setBandwidthLimit(IClient client,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

---

## play2

```
public static void play2(IClient client,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

Server side implementation of NetStream.play(playOptions : NetStreamPlayOptions);

**Parameters:**

client  
function  
params

---

## play

```
public static void play(IClient client,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

Server side implementation of NetStream.play(name : Object [,start : Number[, len : Number[, reset : Object]]]);

**Parameters:**

client - client  
function - function  
params - params (name, start, len, reset)

---

## closeStream

```
public static void closeStream(IClient client,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

Server side implementation of NetStream.close();

**Parameters:**

client - client  
function - function  
params - params (no params)

---

## seek

```
public static void seek(IClient client,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

Server side implementation of NetStream.seek(offset : Number);

### Parameters:

- client - client
- function - function
- params - params (offset)

---

## pause

```
public static void pause(IClient client,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

Server side implementation of NetStream.pause([ flag : Boolean]);

### Parameters:

- client - client
- function - function
- params - params (flag)

---

## setBufferTime

```
public static void setBufferTime(IClient client,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

Server side implementation of NetStream.setBufferTime(bufferTime : Number);

### Parameters:

- client - client
- function - function
- params - params (bufferTime)

---

## getClientID

```
public static void getClientID(IClient client,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

Get the clientId for a client connection NetConnection.call("getClientID", resultObj);

### Parameters:

- client - client
- function - function
- params - params (no params)

---

## getVersion

```
public static void getVersion(IClient client,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

Get the Wowza Pro server version and build number NetConnection.call("getVersion", resultObj);

(continued from last page)

**Parameters:**

client - client  
function - function  
params - params (no params)

## setLiveStreamPacketizer

```
public static void setLiveStreamPacketizer(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Set the live stream packetizer for a stream

**Parameters:**

client - client  
function - function  
params - params (liveStreamPacketizer)

## initLiveStreamRepeating

```
public static void initLiveStreamRepeating(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Initialize a stream for live stream repeating

**Parameters:**

client - client  
function - function  
params - params (liveStreamPacketizer, liveStreamRepeater)

## getLiveStreamPacketizer

```
public static void getLiveStreamPacketizer(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Get the live stream packetizer list for a client connection NetConnection.call("getLiveStreamPacketizer", resultObj);

**Parameters:**

client - client  
function - function  
params - params (no params)

## setStreamType

```
public static void setStreamType(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Set the default stream type for a client connection NetConnection.call("setStreamType", null, streamType);

**Parameters:**

client - client  
function - function  
params - params (streamType)

(continued from last page)

## getStreamType

```
public static void getStreamType(IClient client,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

Get the default stream type for a client connection NetConnection.call("getStreamType", resultObj);

**Parameters:**

- client - client
- function - function
- params - params (no params)

## receiveAudio

```
public static void receiveAudio(IClient client,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

Server side implementation of NetStream.receiveAudio(receive : Boolean);

**Parameters:**

- client - client
- function - function
- params - params (receive)

## receiveVideo

```
public static void receiveVideo(IClient client,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

Server side implementation of NetStream.receiveVideo(receive : Boolean | FPS : Number); FPS does not work the same as FMS. Wowza Pro accepts the following values:

- true: Send all video frames
- false: Send no video
- -1: Send all video frames
- -2: Send half the frames (remove B frames) (Note: only works with Sorenson Spark video with B-frames)
- -3: Send only key frames

**Parameters:**

- client - client
- function - function
- params - params (receive | FPS)

## getStreamBitrate

```
public void getStreamBitrate(IClient client,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

Get stream bitrate in bits-per-second for a video on demand stream (will not work for live). To call, NetConnection.call("getStreamBitrate", resultObj, streamName);

**Parameters:**

- client - client
- function - function
- params - params (streamName:String)

## getStreamLength

```
public static void getStreamLength(IClient client,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

Get stream length in seconds NetConnection.call("getStreamLength", resultObj, streamName); If you pass in an array of streamNames it will return an array of durations.

### Parameters:

- client - client
- function - function
- params - params (streamName:String or streamNames:Array)

---

## getRepeaterOriginUrl

```
public static void getRepeaterOriginUrl(IClient client,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

Get the Repeater Origin URL for this client

### Parameters:

- client - client
- function - function
- params - params

---

## setRepeaterOriginUrl

```
public static void setRepeaterOriginUrl(IClient client,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

Set the Repeater Origin URL for this client

### Parameters:

- client - client
- function - function
- params - params (repeaterOriginUrl)

---

## FCPublish

```
public static void FCPublish(IClient client,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

FCPublish method called by FME 2.5

### Parameters:

- client - client
- function - function
- params - params

---

## FCUnPublish

```
public static void FCUnPublish(IClient client,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

(continued from last page)

FCUnpublish method called by FME 2.5

**Parameters:**

- client - client
- function - function
- params - params

**FCUnpublish**

```
public static void FCUnpublish(IClient client,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

**pauseRaw**

```
public static void pauseRaw(IClient client,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

pauseRaw method introduced in Flash player 10

**Parameters:**

- client - client
- function - function
- params - params

**getReferrer**

```
public static void getReferrer(IClient client,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

getReferrer returns the referrer from the onConnect metadata

**Parameters:**

- client - client
- function - function
- params - params

**getPageUrl**

```
public static void getPageUrl(IClient client,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

getPageUrl returns the pageUrl from the onConnect metadata

**Parameters:**

- client - client
- function - function
- params - params

**FCSubscribe**

```
public static void FCSubscribe(IClient client,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

(continued from last page)

FCSubscribe subscribes to a live stream (if origin edge will start the stream from the origin to the edge)

**Parameters:**

client - client  
function - function  
params - params (streamName:String, [mediaCasterType:String])

---

**FCUnsubscribe**

```
public static void FCUnsubscribe(IClient client,  
                                com.wowza.wms.request.RequestFunction function,  
                                AMFDataList params)
```

FCUnSubscribe to a live stream

**Parameters:**

client - client  
function - function  
params - params (streamName:String)

---

**FCUnSubscribe**

```
public static void FCUnSubscribe(IClient client,  
                                com.wowza.wms.request.RequestFunction function,  
                                AMFDataList params)
```

---

**startTransmit**

```
public static void startTransmit(IClient client,  
                                com.wowza.wms.request.RequestFunction function,  
                                AMFDataList params)
```

---

**stopTransmit**

```
public static void stopTransmit(IClient client,  
                                com.wowza.wms.request.RequestFunction function,  
                                AMFDataList params)
```

## com.wowza.wms.module Class ModuleFastPlay

```
java.lang.Object
+- com.wowza.wms.module.ModuleBase
  +- com.wowza.wms.module.ModuleFastPlay
```

public class **ModuleFastPlay**  
 extends [ModuleBase](#)

### Fields inherited from class [com.wowza.wms.module.ModuleBase](#)

```
CALLBACK_PARAM1, CALLBACK_PARAM10, CALLBACK_PARAM2, CALLBACK_PARAM3, CALLBACK_PARAM4,
CALLBACK_PARAM5, CALLBACK_PARAM6, CALLBACK_PARAM7, CALLBACK_PARAM8, CALLBACK_PARAM9, PARAM1,
PARAM10, PARAM2, PARAM3, PARAM4, PARAM5, PARAM6, PARAM7, PARAM8, PARAM9, PARAMMETHODNAME,
PLAYTRANSITION_APPEND, PLAYTRANSITION_APPEND_IMMEDIATE, PLAYTRANSITION_RESET,
PLAYTRANSITION_RESET_IMMEDIATE, PLAYTRANSITION_STOP, PLAYTRANSITION_SWAP,
PLAYTRANSITION_SWITCH, PLAYTRANSITION_UNKNOWN, PLAYTRANSITIONSTR_APPEND,
PLAYTRANSITIONSTR_RESET, PLAYTRANSITIONSTR_STOP, PLAYTRANSITIONSTR_SWAP,
PLAYTRANSITIONSTR_SWITCH, PLAYTRANSITIONSTR_UNKNOWN
```

## Constructor Summary

public	<a href="#">ModuleFastPlay()</a>
--------	----------------------------------

## Method Summary

static void	<a href="#">setFastPlay(IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)</a>
-------------	---

Turn on fast play NetStream.call("setFastPlay", null, multiplier, frames-per-second, direction);

### Methods inherited from class [com.wowza.wms.module.ModuleBase](#)

```
getAppInstance, getApplication, getCallbackParamCount, getLogger, getParam,
getParamBoolean, getParamBoolean, getParamCount, getParamDate, getParamDouble,
getParamDouble, getParamInt, getParamInt, getParamLong, getParamLong,
getParamMixedArray, getParamObj, getParamString, getParamString, getParamType,
getStream, getVHost, invokePrevious, invokePrevious, isSendResult,
sendClientOnStatusError, sendResult, sendResult, sendResult, sendResult, sendResult,
sendStreamOnStatusError
```

### Methods inherited from class [java.lang.Object](#)

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait,
```

## Constructors

(continued from last page)

## ModuleFastPlay

```
public ModuleFastPlay()
```

### Methods

#### setFastPlay

```
public static void setFastPlay(IClient client,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Turn on fast play NetStream.call("setFastPlay", null, multiplier, frames-per-second, direction);

**Parameters:**

- client - client
- function - function
- params - params (multiplier, frames-per-second, direction)

## com.wowza.wms.module Class ModuleMediaCasterStreamMonitorAdvanced

```
java.lang.Object
+- com.wowza.wms.module.ModuleBase
    +- com.wowza.wms.module.ModuleMediaCasterStreamMonitorAdvanced
```

### All Implemented Interfaces:

[IMediaCasterValidateMediaCaster](#)

---

```
public class ModuleMediaCasterStreamMonitorAdvanced
```

```
extends ModuleBase
```

```
implements IMediaCasterValidateMediaCaster
```

ModuleMediaCasterStreamMonitorAdvanced: Advanced stream monitoring module.

Add this <Module> definition to the end of the <Modules> list in [install-dir]/conf/[application]/Application.xml:

```
<Module>
    <Name>ModuleMediaCasterStreamMonitorAdvanced</Name>
    <Description>ModuleMediaCasterStreamMonitorAdvanced</Description>
    <Class>com.wowza.wms.module.ModuleMediaCasterStreamMonitorAdvanced</Class>
</Module>
```

Add these properties to the application level <Properties> container at the bottom of [install-dir]/conf/[application]/Application.xml (be sure to get the correct <Properties> container - there are several in the Application.xml file). Carefully read the instructions above each set of properties and set accordingly:

```

<!--
Monitor incoming packets (stream, audio, video) to be sure packets continue to flow from
encoder to stream. The
streamMonitor[type]StartTimeout controls the timeout (milliseconds) for the first packet. The
streamMonitor[type]Timeout
controls the timeout (milliseconds) for packets after the first packet. The stream type
refers to a catch all of any packet of
any type (audio, video, data). If any of these values are set to zero, the test is turned
off.
-->
<Property>
    <Name>streamMonitorStreamStartTimeout</Name>
    <Value>20000</Value>
    <Type>Integer</Type>
</Property>
<Property>
    <Name>streamMonitorStreamTimeout</Name>
    <Value>12000</Value>
    <Type>Integer</Type>
</Property>
<Property>
    <Name>streamMonitorVideoStartTimeout</Name>
    <Value>0</Value>
    <Type>Integer</Type>
</Property>
<Property>
    <Name>streamMonitorVideoTimeout</Name>
    <Value>0</Value>
    <Type>Integer</Type>
</Property>
<Property>
    <Name>streamMonitorAudioStartTimeout</Name>
    <Value>0</Value>
    <Type>Integer</Type>
</Property>
<Property>
    <Name>streamMonitorAudioTimeout</Name>
    <Value>0</Value>
    <Type>Integer</Type>
</Property>

<!--
Monitor the incoming packet timecodes (audio, video or data) to be sure packets do not arrive
out of order or late. The
streamMonitor[type]TCPosTolerance and streamMonitor[type]TCNegTolerance (milliseconds) values
define a sliding window
based on the timecode of the previous packet. For example if these values are set to -500 and
3000 respectively then the
timecode difference between the current arriving packet and the previous packet of that type
must fall within
-500 and 3000 milliseconds. If not and streamMonitor[type]TCToleranceEnable is set to true

```

```
then stream will be considered
unhealthy and will be reset.

-->
<Property>
    <Name>streamMonitorVideoTCToleranceEnable</Name>
    <Value>false</Value>
    <Type>Boolean</Type>
</Property>
<Property>
    <Name>streamMonitorVideoTCPoS Tolerance</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>streamMonitorAudioTCPoS Tolerance</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>streamMonitorDataTCPoS Tolerance</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 belongs 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="#">streamMonitorAudioTCPoSolerance</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="#">streamMonitorDataTCPoSolerance</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="#">streamMonitorVideoTCPoSolerance</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(IApplicationInstance appInstance)</a>
void	<a href="#">onAppStop(IApplicationInstance appInstance)</a>
boolean	<a href="#">onResetMediaCaster(IApplicationInstance appInstance, IMediaCaster mediaCaster)</a>
void	<a href="#">onStreamCreate(IMediaStream stream)</a>
void	<a href="#">onStreamDestroy(IMediaStream stream)</a>
boolean	<a href="#">onValidateMediaCaster(IApplicationInstance appInstance, IMediaCaster mediaCaster)</a>
void	<a href="#">onValidateMediaCastersStart(IApplicationInstance appInstance)</a>
void	<a href="#">onValidateMediaCastersStop(IApplicationInstance appInstance)</a>

### 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](#), [sendStreamOnStatusError](#)

### Methods inherited from class java.lang.Object

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [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
```

---

**streamMonitorVideoTCPoS Tolerance**

```
protected int streamMonitorVideoTCPoS Tolerance
```

---

**streamMonitorVideoTCNegTolerance**

```
protected int streamMonitorVideoTCNegTolerance
```

---

**streamMonitorAudioTCToleranceEnable**

```
protected boolean streamMonitorAudioTCToleranceEnable
```

---

**streamMonitorAudioTCPoS Tolerance**

```
protected int streamMonitorAudioTCPoS Tolerance
```

---

**streamMonitorAudioTCNegTolerance**

```
protected int streamMonitorAudioTCNegTolerance
```

---

**streamMonitorDataTCToleranceEnable**

```
protected boolean streamMonitorDataTCToleranceEnable
```

---

**streamMonitorDataTCPoS Tolerance**

```
protected int streamMonitorDataTCPoS Tolerance
```

---

**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(IAplicationInstance appInstance)
```

---

### onAppStop

```
public void onAppStop(IAplicationInstance appInstance)
```

---

### onStreamCreate

```
public void onStreamCreate(IMediaStream stream)
```

---

### onStreamDestroy

```
public void onStreamDestroy(IMediaStream stream)
```

---

### onValidateMediaCastersStart

```
public void onValidateMediaCastersStart(IAplicationInstance appInstance)
```

---

### onValidateMediaCaster

```
public boolean onValidateMediaCaster(IAplicationInstance appInstance,  
        IMediaCaster mediaCaster)
```

---

## onValidateMediaCastersStop

```
public void onValidateMediaCastersStop(IAplicationInstance appInstance)
```

---

## onResetMediaCaster

```
public boolean onResetMediaCaster(IAplicationInstance appInstance,  
                                IMediaCaster mediaCaster)
```

## com.wowza.wms.module Class ModuleProperties

```
java.lang.Object
+- com.wowza.wms.module.ModuleBase
  +- com.wowza.wms.module.ModuleProperties
```

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> (IClient 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> (IClient 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> (IClient 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> (IClient 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> (IClient 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> (IClient client, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params) Set application level property NetConnection.call("setApplicationProperty", null, name, value);

static void	<code>setClientProperty(IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)</code> Set client level property NetConnection.call("setClientProperty", null, name, value);
static void	<code>setStreamProperty(IClient client, com.wowza.wms.request.RequestFunction function, AMFDataList params)</code> 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](#)

## Constructors

### ModuleProperties

```
public ModuleProperties()
```

## Methods

### setApplicationProperty

```
public static void setApplicationProperty(IClient client,
                                         com.wowza.wms.request.RequestFunction function,
                                         AMFDataList params)
```

Set application level property NetConnection.call("setApplicationProperty", null, name, value);

**Parameters:**

- client - client
- function - function
- params - params (name, value)

### getApplicationProperty

```
public static void getApplicationProperty(IClient client,
                                         com.wowza.wms.request.RequestFunction function,
                                         AMFDataList params)
```

Get application level property value NetConnection.call("getApplicationProperty", resultObj, name);

**Parameters:**

- client - client
- function - function

(continued from last page)

params - params (name)

---

## setAppInstanceProperty

```
public static void setAppInstanceProperty(IClient client,  
com.wowza.wms.request.RequestFunction function,  
AMFDataList params)
```

Set application instance level property NetConnection.call("setAppInstanceProperty", null, name, value);

**Parameters:**

client - client  
function - function  
params - params (name, value)

---

## getAppInstanceProperty

```
public static void getAppInstanceProperty(IClient client,  
com.wowza.wms.request.RequestFunction function,  
AMFDataList params)
```

Get application instance level property value NetConnection.call("getAppInstanceProperty", resultObj, name);

**Parameters:**

client - client  
function - function  
params - params (name)

---

## setClientProperty

```
public static void setClientProperty(IClient client,  
com.wowza.wms.request.RequestFunction function,  
AMFDataList params)
```

Set client level property NetConnection.call("setClientProperty", null, name, value);

**Parameters:**

client - client  
function - function  
params - params (name, value)

---

## getClientProperty

```
public static void getClientProperty(IClient client,  
com.wowza.wms.request.RequestFunction function,  
AMFDataList params)
```

Get client level property value NetConnection.call("getClientProperty", resultObj, name);

**Parameters:**

client - client  
function - function  
params - params (name)

---

## setStreamProperty

```
public static void setStreamProperty(IClient client,  
com.wowza.wms.request.RequestFunction function,  
AMFDataList params)
```

(continued from last page)

Set stream level property NetConnection.call("setStreamProperty", null, streamId, name, value);

**Parameters:**

client - client  
function - function  
params - params (streamId, name, value)

---

## getStreamProperty

```
public static void getStreamProperty(IClient client,  
        com.wowza.wms.request.RequestFunction function,  
        AMFDataList params)
```

Get stream level property value NetConnection.call("getStreamProperty", resultObj, streamId, name);

**Parameters:**

client - client  
function - function  
params - params (streamId, name)

---

Package

**com.wowza.wms.pushpublish.model**

## com.wowza.wms.pushpublish.model Interface IPushPublish

All Known Implementing Classes:  
[PushPublishBase](#)

public interface **IPushPublish**  
 extends

The Interface IPushPublisher.

### Field Summary

public static final	<a href="#">PUSHPUBLISH_TYPE_RTMP</a>
	Value: <b>1</b>
public static final	<a href="#">PUSHPUBLISH_TYPE_RTP</a>
	Value: <b>2</b>
public static final	<a href="#">PUSHPUBLISH_TYPE_UNKNOWN</a>
	Value: <b>0</b>

### Method Summary

void	<a href="#">connect()</a>
<a href="#">IPushPublishSession</a>	<a href="#">createPushPublishSession()</a>
void	<a href="#">disconnect()</a>
String	<a href="#">getContextStr()</a>
String	<a href="#">getDstStreamName()</a>
String	<a href="#">getHostname()</a>
int	<a href="#">getMapFileIndex()</a>
int	<a href="#">getPort()</a>
<a href="#">WMSProperties</a>	<a href="#">getProperties()</a>
int	<a href="#">getPushPublishType()</a>
int	<a href="#">getReconnectWaitTime()</a>
<a href="#">IMediaStream</a>	<a href="#">getSrcStream()</a>

String	<a href="#">getSrcStreamName()</a>
void	<a href="#">init(IApplicationInstance appInstance2, String streamName, IMediaStream stream, java.util.HashMap profileData, java.util.HashMap maps, boolean debug)</a>
void	<a href="#">resetConnection()</a>
void	<a href="#">setDebugLog(boolean debugLog)</a>
void	<a href="#">setDstStreamName(String dstStreamNameIn)</a>
void	<a href="#">setHost(String hostname)</a>
void	<a href="#">setHostname(String hostname)</a>
void	<a href="#">setMapFileIndex(String index)</a>
void	<a href="#">setPort(int port)</a>
void	<a href="#">setPort(String port)</a>
void	<a href="#">setReconnectWaitTime(int reconnectWaitTime)</a>
void	<a href="#">setSrcStream(IMediaStream srcStream)</a>
void	<a href="#">setSrcStreamName(String srcStreamName)</a>
void	<a href="#">setStreamName(String streamName)</a>

## Fields

### PUSHPUBLISH\_TYPE\_UNKNOWN

```
public static final int PUSHPUBLISH_TYPE_UNKNOWN
```

Constant value: 0

### PUSHPUBLISH\_TYPE\_RTMP

```
public static final int PUSHPUBLISH_TYPE_RTMP
```

Constant value: 1

### PUSHPUBLISH\_TYPE\_RTP

```
public static final int PUSHPUBLISH_TYPE_RTP
```

(continued from last page)

Constant value: 2

## Methods

### connect

```
public void connect()
```

---

### disconnect

```
public void disconnect()
```

---

### resetConnection

```
public void resetConnection()
```

---

### getProperties

```
public WMSProperties getProperties()
```

---

### getMapFileIndex

```
public int getMapFileIndex()
```

---

### setMapFileIndex

```
public void setMapFileIndex(String index)
```

---

### getSrcStreamName

```
public String getSrcStreamName()
```

---

### setSrcStreamName

```
public void setSrcStreamName(String srcStreamName)
```

---

### getDstStreamName

```
public String getDstStreamName()
```

---

### **setDstStreamName**

```
public void setDstStreamName(String dstStreamNameIn)
```

---

### **setStreamName**

```
public void setStreamName(String streamName)
```

---

### **getHostname**

```
public String getHostname()
```

---

### **setHost**

```
public void setHost(String hostname)
```

---

### **setHostname**

```
public void setHostname(String hostname)
```

---

### **setDebugLog**

```
public void setDebugLog(boolean debugLog)
```

---

### **getPort**

```
public int getPort()
```

---

### **setPort**

```
public void setPort(int port)
```

---

### **setPort**

```
public void setPort(String port)
```

---

### **getReconnectWaitTime**

```
public int getReconnectWaitTime()
```

---

(continued from last page)

---

**setReconnectWaitTime**

```
public void setReconnectWaitTime(int reconnectWaitTime)
```

---

**init**

```
public void init(IApplicationInstance appInstance2,  
                 String streamName,  
                 IMediaStream stream,  
                 java.util.HashMap profileData,  
                 java.util.HashMap maps,  
                 boolean debug)
```

---

**getPushPublishType**

```
public int getPushPublishType()
```

---

**createPushPublishSession**

```
public IPushPublishSession createPushPublishSession()
```

---

**getSrcStream**

```
public IMediaStream getSrcStream()
```

---

**setSrcStream**

```
public void setSrcStream(IMediaStream srcStream)
```

---

**getContextStr**

```
public String getContextStr()
```

---

## com.wowza.wms.pushpublish.model Interface IPushPublishCrossSession

---

public interface IPushPublishCrossSession  
extends

---

### Method Summary

boolean	<a href="#"><u>isSessionValid()</u></a>
---------	---

### Methods

#### **isSessionValid**

public boolean **isSessionValid()**

## com.wowza.wms.pushpublish.model Interface IPushPublishSession

All Known Implementing Classes:  
[PushPublishSessionBase](#)

---

public interface **IPushPublishSession**  
 extends

The Interface IPushPublishSession.

### Field Summary

public static final	<a href="#">PUSHPUBLISHSESSION_TYPE_RTMP</a>
	Value: <b>1</b>
public static final	<a href="#">PUSHPUBLISHSESSION_TYPE_RTP</a>
	Value: <b>2</b>
public static final	<a href="#">PUSHPUBLISHSESSION_TYPE_UNKNOWN</a>
	Value: <b>0</b>

### Method Summary

String	<a href="#">getContextStr()</a>
String	<a href="#">getProfile()</a>
<a href="#">IPushPublish</a>	<a href="#">getPushPublishImpl()</a>
<a href="#">IMediaStream</a>	<a href="#">getStream()</a>
int	<a href="#">getType()</a>
void	<a href="#">setProfile(String profile)</a>
void	<a href="#">setPushPublishImpl(<a href="#">IPushPublish</a> pushPublishImpl)</a>
void	<a href="#">setStream(<a href="#">IMediaStream</a> stream)</a>

### Fields

#### PUSHPUBLISHSESSION\_TYPE\_UNKNOWN

public static final int **PUSHPUBLISHSESSION\_TYPE\_UNKNOWN**

(continued from last page)

Constant value: 0

**PUSHPUBLISHSESSION\_TYPE\_RTMP**

public static final int PUSHPUBLISHSESSION\_TYPE\_RTMP

Constant value: 1

**PUSHPUBLISHSESSION\_TYPE\_RTP**

public static final int PUSHPUBLISHSESSION\_TYPE\_RTP

Constant value: 2

**Methods****getType**

public int getType()

**getPushPublishImpl**public [IPushPublish](#) getPushPublishImpl()**setPushPublishImpl**public void setPushPublishImpl([IPushPublish](#) pushPublishImpl)**getStream**public [IMediaStream](#) getStream()**setStream**public void setStream([IMediaStream](#) stream)**getProfile**

public String getProfile()

**setProfile**

public void setProfile(String profile)

(continued from last page)

---

## **getContextStr**

```
public String getContextStr()
```

## com.wowza.wms.pushpublish.model Interface IPushPublishSessionNotify

All Known Implementing Classes:

[PushPublishSessionNotifyBase](#)

public interface **IPushPublishSessionNotify**

extends

### Method Summary

void	<a href="#"><b>onPushPublishSessionCreate</b></a> (String streamName, <a href="#">IMediaStream</a> stream, <a href="#">IPushPublishSession</a> pushPublishSession) Invoked when push publish session is created
void	<a href="#"><b>onPushPublishSessionDestroy</b></a> (String streamName, <a href="#">IMediaStream</a> stream, <a href="#">IPushPublishSession</a> pushPublishSession) Invoked when push publish session is destroyed

### Methods

#### **onPushPublishSessionCreate**

```
public void onPushPublishSessionCreate(String streamName,
    IMediaStream stream,
    IPushPublishSession pushPublishSession)
```

Invoked when push publish session is created

**Parameters:**

- streamName - stream name
- stream - stream
- pushPublishSession - push publish session

#### **onPushPublishSessionDestroy**

```
public void onPushPublishSessionDestroy(String streamName,
    IMediaStream stream,
    IPushPublishSession pushPublishSession)
```

Invoked when push publish session is destroyed

**Parameters:**

- streamName - stream name
- stream - stream
- pushPublishSession - push publish session

## com.wowza.wms.pushpublish.model Class PushPublishBase

```
java.lang.Object
+-com.wowza.wms.pushpublish.model.PushPublishBase
```

All Implemented Interfaces:

[IPushPublish](#)

Direct Known Subclasses:

[PushPublishRTP](#), [PushPublishRTMP](#), [PushPublishHTTPBase](#)

public abstract class **PushPublishBase**

extends Object

implements [IPushPublish](#)

The Class PushPublisherBase.

### Field Summary

protected	<a href="#">appInstance</a> The app instance.
public static final	<a href="#">CLASS</a>
public static final	<a href="#">CLASSNAME</a> Value: <b>PushPublishBase</b>
protected	<a href="#">dstStreamName</a> The dst stream name.
protected	<a href="#">hostname</a> The m host.
protected	<a href="#">mapFileIndex</a>
protected	<a href="#">mImplementation</a> The m implementation.
protected	<a href="#">mType</a> The m type.
protected	<a href="#">originalTimecodeThreshold</a>
protected	<a href="#">password</a> The password.
protected	<a href="#">port</a> The m port.
protected	<a href="#">properties</a> The properties.

protected	<a href="#">pushPublishStreamDebug</a>
protected	<a href="#">pushPublishType</a>
protected	<a href="#">reconnectWaitTime</a>
protected	<a href="#">sendingPerfomance</a>
protected	<a href="#">sendOriginalTimecodes</a> The my params.
protected	<a href="#">srcStream</a>
protected	<a href="#">srcStreamName</a> The src stream name.
protected	<a href="#">userName</a> The user name.

Fields inherited from interface [com.wowza.wms.pushpublish.model.IPushPublish](#)

[PUSHPUBLISH\\_TYPE\\_RTMP](#), [PUSHPUBLISH\\_TYPE\\_RTP](#), [PUSHPUBLISH\\_TYPE\\_UNKNOWN](#)

## Constructor Summary

public	<a href="#">PushPublishBase()</a> Instantiates a new push publisher base.
--------	--

## Method Summary

<a href="#">IPushPublishSession</a>	<a href="#">createPushPublishSession()</a>
<a href="#">IAplicationInstance</a>	<a href="#">getAppInstance()</a> Get appInstance interface.
double	<a href="#">getBandwidth()</a>
boolean	<a href="#">getConfigDataOrDefault</a> (org.w3c.dom.Element config, String tag, boolean deflt) Gets the config data or default.
int	<a href="#">getConfigDataOrDefault</a> (org.w3c.dom.Element config, String tag, int deflt) Gets the config data or default.
String	<a href="#">getConfigDataOrDefault</a> (org.w3c.dom.Element config, String tag, String deflt) Gets the config data or default.
String	<a href="#">getContextStr()</a>
String	<a href="#">getDstStreamName()</a> Get destination stream name.

String	<a href="#"><u>getHostname()</u></a> Get destination server host name.
String	<a href="#"><u>getImplementation()</u></a> Gets the implementation.
int	<a href="#"><u>getMapFileIndex()</u></a>
static int	<a href="#"><u>getNumberFromString(String in)</u></a>
long	<a href="#"><u>getOriginalTimecodeThreshold()</u></a> Get the original timecode threshold.
String	<a href="#"><u>getPassword()</u></a> Get Akamai password.
int	<a href="#"><u>getPort()</u></a> Get destination server port.
<a href="#"><u>WMSProperties</u></a>	<a href="#"><u>getProperties()</u></a>
int	<a href="#"><u>getPushPublishType()</u></a>
static String	<a href="#"><u>getPushPublishVersionStr()</u></a> Gets the push publish version str.
int	<a href="#"><u>getReconnectWaitTime()</u></a> Get the reconnect wait time (milliseconds).
<a href="#"><u>IMediaStream</u></a>	<a href="#"><u>getSrcStream()</u></a>
String	<a href="#"><u>getSrcStreamName()</u></a> Get source stream name.
String	<a href="#"><u>getUserName()</u></a> Gets the user name.
void	<a href="#"><u>init(IApplicationInstance appInstance2, String streamName, IMediaStream stream, java.util.HashMap profileData, java.util.HashMap maps, boolean streamDebug)</u></a> Inits the instance.
boolean	<a href="#"><u>isAdaptiveStreaming()</u></a> Checks if is adaptive streaming.
boolean	<a href="#"><u>isSendOriginalTimecodes()</u></a> true if sending the original timecodes of source stream.
void	<a href="#"><u>load(java.util.HashMap dataMap)</u></a>
void	<a href="#"><u>setAdaptiveStreaming(boolean isAdaptiveStreaming)</u></a> Sets the adaptive streaming.
void	<a href="#"><u>setAppInstance(IApplicationInstance appInstance)</u></a> Set appInstance interface.

void	<a href="#"><u>setDebugLog</u></a> (boolean debugLog) Set debug logging.
void	<a href="#"><u>setDstStreamName</u></a> (String dstStreamNameIn) Set destination stream name.
void	<a href="#"><u>setHost</u></a> (String hostname) Set destination server host name.
void	<a href="#"><u>setHostname</u></a> (String hostname)
void	<a href="#"><u>setImplementation</u></a> (String implementation) Sets the implementation.
void	<a href="#"><u>setMapFileIndex</u></a> (String index)
void	<a href="#"><u>setOriginalTimecodeThreshold</u></a> (long originalTimecodeThreshold) Set the original timecode threshold.
void	<a href="#"><u>setOriginalTimecodeThreshold</u></a> (String i)
void	<a href="#"><u>setPassword</u></a> (String password) Set Akamai password.
void	<a href="#"><u>setPort</u></a> (int port) Set destination server port.
void	<a href="#"><u>setPort</u></a> (String port)
void	<a href="#"><u>setReconnectWaitTime</u></a> (int reconnectWaitTime) Set the reconnect wait time (milliseconds).
void	<a href="#"><u>setSendOriginalTimecodes</u></a> (boolean sendOriginalTimecodes) true if sending the original timecodes of source stream.
void	<a href="#"><u>setSrcStream</u></a> ( <a href="#">IMediaStream</a> srcStream)
void	<a href="#"><u>setSrcStreamName</u></a> (String srcStreamName) Set source stream name.
void	<a href="#"><u>setStreamName</u></a> (String streamName) Sets the stream name.
void	<a href="#"><u>setUserName</u></a> (String userName) Set Akamai user name.

**Methods inherited from class java.lang.Object**

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait`

**Methods inherited from interface [com.wowza.wms.pushpublish.model.IPushPublish](#)**

```
connect, createPushPublishSession, disconnect, getContextStr, getDstStreamName,  
getHostname, getMapFileIndex, getPort, getProperties, getPushPublishType,  
getReconnectWaitTime, getSrcStream, getSrcStreamName, init, resetConnection,  
setDebugLog, setDstStreamName, setHost, setHostname, setMapFileIndex, setPort,  
setPort, setReconnectWaitTime, setSrcStream, setSrcStreamName, setStreamName
```

## Fields

### CLASS

```
public static final java.lang.Class CLASS
```

### CLASSNAME

```
public static final java.lang.String CLASSNAME
```

Constant value: **PushPublishBase**

### pushPublishType

```
protected int pushPublishType
```

### sendingPerfomance

```
protected com.wowza.util.IOPerformanceCounter sendingPerfomance
```

### pushPublishStreamDebug

```
protected boolean pushPublishStreamDebug
```

### reconnectWaitTime

```
protected int reconnectWaitTime
```

### mapFileIndex

```
protected int mapFileIndex
```

### properties

```
protected com.wowza.wms.application.WMSProperties properties
```

(continued from last page)

The properties.

---

**userName**

```
protected java.lang.String userName
```

The user name.

---

**password**

```
protected java.lang.String password
```

The password.

---

**appInstance**

```
protected com.wowza.wms.application.IApplicationInstance appInstance
```

The app instance.

---

**srcStreamName**

```
protected java.lang.String srcStreamName
```

The src stream name.

---

**srcStream**

```
protected com.wowza.wms.stream.IMediaStream srcStream
```

---

**dstStreamName**

```
protected java.lang.String dstStreamName
```

The dst stream name.

---

**hostname**

```
protected java.lang.String hostname
```

The m host.

---

**port**

```
protected int port
```

The m port.

---

**mImplementation**

```
protected java.lang.String mImplementation
```

The m implementation.

---

(continued from last page)

## mType

```
protected java.lang.String mType
```

The m type.

## sendOriginalTimecodes

```
protected boolean sendOriginalTimecodes
```

The my params.

## originalTimecodeThreshold

```
protected long originalTimecodeThreshold
```

## Constructors

### PushPublishBase

```
public PushPublishBase()
```

Instantiates a new push publisher base.

## Methods

### init

```
public void init(IAplicationInstance appInstance2,  
String streamName,  
IMediaStream stream,  
java.util.HashMap profileData,  
java.util.HashMap maps,  
boolean streamDebug)
```

Inits the instance.

#### Parameters:

appInstance2 - the app instance2  
streamName  
maps  
profileData

### getImplementation

```
public String getImplementation()
```

Gets the implementation.

#### Returns:

the implementation

### setImplementation

```
public void setImplementation(String implementation)
```

Sets the implementation.

(continued from last page)

**Parameters:**

implementation - the new implementation

---

**getProperties**

```
public WMSProperties getProperties()
```

---

**getConfigDataOrDefault**

```
protected String getConfigDataOrDefault(org.w3c.dom.Element config,
    String tag,
    String deflt)
```

Gets the config data or default.

**Parameters:**

config - the config  
tag - the tag  
deflt - the deflt

**Returns:**

the config data or default

---

**getConfigDataOrDefault**

```
protected int getConfigDataOrDefault(org.w3c.dom.Element config,
    String tag,
    int deflt)
```

Gets the config data or default.

**Parameters:**

config - the config  
tag - the tag  
deflt - the deflt

**Returns:**

the config data or default

---

**getConfigDataOrDefault**

```
protected boolean getConfigDataOrDefault(org.w3c.dom.Element config,
    String tag,
    boolean deflt)
```

Gets the config data or default.

**Parameters:**

config - the config  
tag - the tag  
deflt - the deflt

**Returns:**

the config data or default

---

(continued from last page)

## isAdaptiveStreaming

```
public boolean isAdaptiveStreaming()
```

Checks if is adaptive streaming.

**Returns:**

true, if is adaptive streaming

---

## setAdaptiveStreaming

```
public void setAdaptiveStreaming(boolean isAdaptiveStreaming)
```

Sets the adaptive streaming.

**Parameters:**

isAdaptiveStreaming - the new adaptive streaming

---

## isSendOriginalTimecodes

```
public boolean isSendOriginalTimecodes()
```

true if sending the original timecodes of source stream. false if starting at zero.

**Returns:**

true if sending the original timecodes of source stream

---

## setSendOriginalTimecodes

```
public void setSendOriginalTimecodes(boolean sendOriginalTimecodes)
```

true if sending the original timecodes of source stream. false if starting at zero.

**Parameters:**

sendOriginalTimecodes - true if sending the original timecodes of source stream

---

## getOriginalTimecodeThreshold

```
public long getOriginalTimecodeThreshold()
```

Get the original timecode threshold. This value will attempt to reduce the magnitude of absolute timecodes while keeping multiple-bitrate alignment. It is probably best to set this value quite high to a value such as 0x100000. The value is in milliseconds. The following formula is used to calculate a timecode offset timecodeOffset = timecode - (timecode%originalTimecodeThreshold);

**Returns:**

original timecode threshold

---

## setOriginalTimecodeThreshold

```
public void setOriginalTimecodeThreshold(long originalTimecodeThreshold)
```

Set the original timecode threshold. This value will attempt to reduce the magnitude of absolute timecodes while keeping multiple-bitrate alignment. It is probably best to set this value quite high to a value such as 0x100000. The value is in milliseconds. The following formula is used to calculate a timecode offset timecodeOffset = timecode - (timecode%originalTimecodeThreshold);

**Parameters:**

originalTimecodeThreshold - original timecode threshold

## setOriginalTimecodeThreshold

```
public void setOriginalTimecodeThreshold(String i)
```

---

## getUserName

```
public String getUserName()
```

Gets the user name.

**Returns:**

the user name

---

## setUserName

```
public void setUserName(String userName)
```

Set Akamai user name.

**Parameters:**

userName - Akamai user name

---

## getPassword

```
public String getPassword()
```

Get Akamai password.

**Returns:**

Akamai password

---

## setPassword

```
public void setPassword(String password)
```

Set Akamai password.

**Parameters:**

password - Akamai password

---

## getAppInstance

```
public IApplicationInstance getAppInstance()
```

Get appInstance interface.

**Returns:**

appInstance interface

---

## setAppInstance

```
public void setAppInstance(IApplicationInstance appInstance)
```

Set appInstance interface.

(continued from last page)

**Parameters:**

appInstance - appInstance interface

**getSrcStreamName****public String getSrcStreamName( )**

Get source stream name.

**Returns:**

source stream name

**setSrcStreamName****public void setSrcStreamName(String srcStreamName)**

Set source stream name.

**Parameters:**

srcStreamName - source stream name

**getDstStreamName****public String getDstStreamName( )**

Get destination stream name.

**Returns:**

destination stream name

**setDstStreamName****public void setDstStreamName(String dstStreamNameIn)**

Set destination stream name.

**Parameters:**

dstStreamNameIn - destination stream name

**setStreamName****public void setStreamName(String streamName)**

Sets the stream name.

**Parameters:**

streamName - the new stream name

**getHostname****public String getHostname()**

Get destination server host name.

**Returns:**

destination server host name

(continued from last page)

## setHost

```
public void setHost(String hostname)
```

Set destination server host name.

**Parameters:**

hostname - destination server host name

---

## setHostname

```
public void setHostname(String hostname)
```

---

## setDebugLog

```
public void setDebugLog(boolean debugLog)
```

Set debug logging.

**Parameters:**

debugLog - debug logging

---

## getPort

```
public int getPort()
```

Get destination server port.

**Returns:**

destination server port

---

## setPort

```
public void setPort(int port)
```

Set destination server port.

**Parameters:**

port - destination server port

---

## setPort

```
public void setPort(String port)
```

---

## getPushPublishVersionStr

```
public static String getPushPublishVersionStr()
```

Gets the push publish version str.

**Returns:**

the push publish version str

(continued from last page)

**load**

```
public void load(java.util.HashMap dataMap)
```

---

**getNumberFromString**

```
protected static int getNumberFromString(String in)
```

---

**getMapFileIndex**

```
public int getMapFileIndex()
```

---

**setMapFileIndex**

```
public void setMapFileIndex(String index)
```

---

**getReconnectWaitTime**

```
public int getReconnectWaitTime()
```

Get the reconnect wait time (milliseconds). Shortest time between connection attempts.

**Returns:**

reconnect wait time (milliseconds)

---

**setReconnectWaitTime**

```
public void setReconnectWaitTime(int reconnectWaitTime)
```

Set the reconnect wait time (milliseconds). Shortest time between connection attempts.

**Parameters:**

reconnectWaitTime - reconnect wait time (milliseconds)

---

**getBandwidth**

```
public double getBandwidth()
```

---

**getPushPublishType**

```
public int getPushPublishType()
```

---

**createPushPublishSession**

```
public IPushPublishSession createPushPublishSession()
```

(continued from last page)

---

**getSrcStream**

```
public IMediaStream getSrcStream( )
```

---

**setSrcStream**

```
public void setSrcStream(IMediaStream srcStream)
```

---

**getContextStr**

```
public String getContextStr( )
```

## com.wowza.wms.pushpublish.model Class PushPublishContext

```
java.lang.Object
+-com.wowza.wms.pushpublish.model.PushPublishContext
```

---

```
public class PushPublishContext
extends Object
```

### Field Summary

public static final	<a href="#">CLASS</a>
public static final	<a href="#">CLASSNAME</a> Value: <b>PushPublishContext</b>

### Constructor Summary

public	<a href="#">PushPublishContext(IVHost vhost)</a>
--------	--

### Method Summary

void	<a href="#">doWatchdog()</a>
<a href="#">IPushPublishCrossSession</a>	<a href="#">getCrossSession(String contextStr)</a>
Object	<a href="#">getCrossSessionLock()</a>
int	<a href="#">getNextMapIndex()</a>
java.util.HashMap	<a href="#">getProfileData(String profileName)</a>
void	<a href="#">putCrossSession(String contextStr, <a href="#">IPushPublishCrossSession</a> crossSession)</a>
void	<a href="#">putProfileData(String profileName, java.util.HashMap profileData)</a>
<a href="#">IPushPublishCrossSession</a>	<a href="#">removeCrossSession(String contextStr)</a>

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait
--

(continued from last page)

## Fields

### CLASS

```
public static final java.lang.Class CLASS
```

---

### CLASSNAME

```
public static final java.lang.String CLASSNAME
```

Constant value: `PushPublishContext`

## Constructors

### PushPublishContext

```
public PushPublishContext(IVHost vhost)
```

## Methods

### getProfileData

```
public java.util.HashMap getProfileData(String profileName)
```

---

### putProfileData

```
public void putProfileData(String profileName,  
                           java.util.HashMap profileData)
```

---

### getNextMapIndex

```
public int getNextMapIndex()
```

---

### putCrossSession

```
public void putCrossSession(String contextStr,  
                           IPushPublishCrossSession crossSession)
```

---

### getCrossSession

```
public IPushPublishCrossSession getCrossSession(String contextStr)
```

(continued from last page)

## **removeCrossSession**

```
public IPushPublishCrossSession removeCrossSession(String contextStr)
```

---

## **doWatchdog**

```
public void doWatchdog()
```

---

## **getCrossSessionLock**

```
public Object getCrossSessionLock()
```

## com.wowza.wms.pushpublish.model Class PushPublishHTTPBase

```
java.lang.Object
+- com.wowza.wms.pushpublish.model.PushPublishBase
  +- com.wowza.wms.pushpublish.model.PushPublishHTTPBase
```

All Implemented Interfaces:

[IPushPublish](#)

public abstract class **PushPublishHTTPBase**

extends [PushPublishBase](#)

Fields inherited from class [com.wowza.wms.pushpublish.model.PushPublishBase](#)

```
appInstance, CLASS, CLASSNAME, dstStreamName, hostname, mapFileIndex, mImplementation, mType,
originalTimecodeThreshold, password, port, properties, pushPublishStreamDebug,
pushPublishType, reconnectWaitTime, sendingPerfomance, sendOriginalTimecodes, srcStream,
srcStreamName, user Name
```

Fields inherited from interface [com.wowza.wms.pushpublish.model.IPushPublish](#)

```
PUSHPUBLISH_TYPE_RTMP, PUSHPUBLISH_TYPE_RTP, PUSHPUBLISH_TYPE_UNKNOWN
```

## Constructor Summary

public	<a href="#">PushPublishHTTPBase()</a>
--------	---------------------------------------

Methods inherited from class [com.wowza.wms.pushpublish.model.PushPublishBase](#)

```
createPushPublishSession, getAppInstance, getBandwidth, getConfigDataOrDefault,
getConfigDataOrDefault, getConfigDataOrDefault, getContextStr, getDstStreamName,
getHostname, getImplementation, getMapFileIndex, getNumberFromString,
getOriginalTimecodeThreshold, getPassword, getPort, getProperties,
getPushPublishType, getPushPublishVersionStr, getReconnectWaitTime, getSrcStream,
getSrcStreamName, getUserName, init, isAdaptiveStreaming, isSendOriginalTimecodes,
load, setAdaptiveStreaming, setAppInstance, setDebugLog, setDstStreamName, setHost,
setHostname, setImplementation, setMapFileIndex, setOriginalTimecodeThreshold,
setOriginalTimecodeThreshold, setPassword, setPort, setPort, setReconnectWaitTime,
setSendOriginalTimecodes, setSrcStream, setSrcStreamName, setStreamName, setUserName
```

Methods inherited from class [java.lang.Object](#)

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait,
```

Methods inherited from interface [com.wowza.wms.pushpublish.model.IPushPublish](#)

```
connect, createPushPublishSession, disconnect, getContextStr, getDstStreamName,  
getHostname, getMapFileIndex, getPort, getProperties, getPushPublishType,  
getReconnectWaitTime, getSrcStream, getSrcStreamName, init, resetConnection,  
setDebugLog, setDstStreamName, setHost, setHostname, setMapFileIndex, setPort,  
setPort, setReconnectWaitTime, setSrcStream, setSrcStreamName, setStreamName
```

## Constructors

### PushPublishHTTPBase

```
public PushPublishHTTPBase()
```

## com.wowza.wms.pushpublish.model Class PushPublishSessionBase

```
java.lang.Object
+-com.wowza.wms.pushpublish.model.PushPublishSessionBase
```

### All Implemented Interfaces:

[IPushPublishSession](#)

### Direct Known Subclasses:

[PushPublishSessionRTP](#), [PushPublishSessionRTMP](#), [PushPublishSessionHTTPBase](#)

```
public class PushPublishSessionBase
extends Object
implements IPushPublishSession
```

## Field Summary

protected	<a href="#">profile</a>
protected	<a href="#">pushPublishImpl</a>
protected	<a href="#">stream</a>
protected	<a href="#">type</a>

### Fields inherited from interface [com.wowza.wms.pushpublish.model.IPushPublishSession](#)

[PUSHPUBLISHSESSION\\_TYPE\\_RTMP](#), [PUSHPUBLISHSESSION\\_TYPE\\_RTP](#),  
[PUSHPUBLISHSESSION\\_TYPE\\_UNKNOWN](#)

## Constructor Summary

public	<a href="#">PushPublishSessionBase()</a>
--------	--

## Method Summary

String	<a href="#">getContextStr()</a>
String	<a href="#">getProfile()</a>
<a href="#">IPushPublish</a>	<a href="#">getPushPublishImpl()</a>
<a href="#">IMediaStream</a>	<a href="#">getStream()</a>
int	<a href="#">getType()</a>

void	<a href="#"><u>setProfile</u></a> (String profile)
void	<a href="#"><u>setPushPublishImpl</u></a> (IPushPublish pushPublishImpl)
void	<a href="#"><u>setStream</u></a> (IMediaStream stream)

**Methods inherited from class** java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

**Methods inherited from interface** com.wowza.wms.pushpublish.model.IPushPublishSession

[getContextStr](#), [getProfile](#), [getPushPublishImpl](#), [getStream](#), [getType](#), [setProfile](#), [setPushPublishImpl](#), [setStream](#)

## Fields

**type**

```
protected int type
```

**pushPublishImpl**

```
protected com.wowza.wms.pushpublish.model.IPushPublish pushPublishImpl
```

**stream**

```
protected com.wowza.wms.stream.IMediaStream stream
```

**profile**

```
protected java.lang.String profile
```

## Constructors

**PushPublishSessionBase**

```
public PushPublishSessionBase()
```

## Methods

(continued from last page)

**getType**

```
public int getType()
```

---

**getPushPublishImpl**

```
public IPushPublish getPushPublishImpl()
```

---

**setPushPublishImpl**

```
public void setPushPublishImpl(IPushPublish pushPublishImpl)
```

---

**getStream**

```
public IMediaStream getStream()
```

---

**setStream**

```
public void setStream(IMediaStream stream)
```

---

**getProfile**

```
public String getProfile()
```

---

**setProfile**

```
public void setProfile(String profile)
```

---

**getContextStr**

```
public String getContextStr()
```

## com.wowza.wms.pushpublish.model Class PushPublishSessionHTTPBase

```
java.lang.Object
+- com.wowza.wms.pushpublish.model.PushPublishSessionBase
  +- com.wowza.wms.pushpublish.model.PushPublishSessionHTTPBase
```

### All Implemented Interfaces:

[IPushPublishSession](#)

public abstract class **PushPublishSessionHTTPBase**

extends [PushPublishSessionBase](#)

Fields inherited from class [com.wowza.wms.pushpublish.model.PushPublishSessionBase](#)

[profile](#), [pushPublishImpl](#), [stream](#), [type](#)

Fields inherited from interface [com.wowza.wms.pushpublish.model.IPushPublishSession](#)

[PUSHPUBLISHSESSION\\_TYPE\\_RTMP](#), [PUSHPUBLISHSESSION\\_TYPE RTP](#),  
[PUSHPUBLISHSESSION\\_TYPE\\_UNKNOWN](#)

## Constructor Summary

public	<a href="#">PushPublishSessionHTTPBase()</a>
--------	--

Methods inherited from class [com.wowza.wms.pushpublish.model.PushPublishSessionBase](#)

[getContextStr](#), [getProfile](#), [getPushPublishImpl](#), [getStream](#), [getType](#), [setProfile](#),  
[setPushPublishImpl](#), [setStream](#)

Methods inherited from class [java.lang.Object](#)

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#)

Methods inherited from interface [com.wowza.wms.pushpublish.model.IPushPublishSession](#)

[getContextStr](#), [getProfile](#), [getPushPublishImpl](#), [getStream](#), [getType](#), [setProfile](#),  
[setPushPublishImpl](#), [setStream](#)

## Constructors

### PushPublishSessionHTTPBase

public [PushPublishSessionHTTPBase\(\)](#)

## com.wowza.wms.pushpublish.model Class PushPublishSessionNotifyBase

```
java.lang.Object
+-com.wowza.wms.pushpublish.model.PushPublishSessionNotifyBase
```

All Implemented Interfaces:  
[IPushPublishSessionNotify](#)

public abstract class **PushPublishSessionNotifyBase**  
 extends Object  
 implements [IPushPublishSessionNotify](#)

Abstract base class for [IPushPublishSessionNotify](#) interface

See Also:

[IApplicationInstance.addPushPublishSessionListener\(IPushPublishSessionNotify\)](#)

### Constructor Summary

public	<a href="#">PushPublishSessionNotifyBase()</a>
--------	--

### Method Summary

void	<a href="#">onPushPublishSessionCreate(String streamName, IMediaStream stream, IPushPublishSession pushPublishSession)</a>
void	<a href="#">onPushPublishSessionDestroy(String streamName, IMediaStream stream, IPushPublishSession pushPublishSession)</a>

#### Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait`

#### Methods inherited from interface com.wowza.wms.pushpublish.model.IPushPublishSessionNotify

[onPushPublishSessionCreate](#), [onPushPublishSessionDestroy](#)

### Constructors

#### PushPublishSessionNotifyBase

public [PushPublishSessionNotifyBase\(\)](#)

### Methods

(continued from last page)

## onPushPublishSessionCreate

```
public void onPushPublishSessionCreate(String streamName,  
    IMediaStream stream,  
    IPushPublishSession pushPublishSession)
```

---

## onPushPublishSessionDestroy

```
public void onPushPublishSessionDestroy(String streamName,  
    IMediaStream stream,  
    IPushPublishSession pushPublishSession)
```

---

Package

**com.wowza.wms.pushpublish.protocol.rtmp**

## com.wowza.wms.pushpublish.protocol.rtmp Interface IPushPublishRTMPAuthProvider

All Known Implementing Classes:

[PushPublishRTMPAuthProviderBase](#)

---

public interface **IPushPublishRTMPAuthProvider**  
extends

### Method Summary

String	<a href="#">addAuthQueryString</a> (String queryString)
String	<a href="#">getPassword</a> ()
<a href="#">PushPublishRTMP</a>	<a href="#">getPublisher</a> ()
String	<a href="#">getUserName</a> ()
void	<a href="#">handleConnectResponse</a> (com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params)
void	<a href="#">init</a> ( <a href="#">PushPublishRTMP</a> publisher)
void	<a href="#">setPassword</a> (String password)
void	<a href="#">setUserName</a> (String userName)

### Methods

#### **init**

public void [init](#)([PushPublishRTMP](#) publisher)

---

#### **addAuthQueryString**

public String [addAuthQueryString](#)(String queryString)

---

#### **handleConnectResponse**

public void [handleConnectResponse](#)(com.wowza.wms.request.RequestFunction function,  
[AMFDataList](#) params)

---

### getUserName

```
public String getUserName()
```

---

### setUserName

```
public void setUserName(String userName)
```

---

### getPassword

```
public String getPassword()
```

---

### setPassword

```
public void setPassword(String password)
```

---

### getPublisher

```
public PushPublishRTMP getPublisher()
```

## com.wowza.wms.pushpublish.protocol.rtmp Interface IPushPublishRTMPNotify

All Known Implementing Classes:

[PushPublishRTMPNotifyBase](#)

---

public interface **IPushPublishRTMPNotify**

extends

### Method Summary

void	<a href="#">onAkamaiClientLogin(PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)</a>
void	<a href="#">onAkamaiSetChallenge(PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)</a>
void	<a href="#">onAkamaiSetOriginConnectionInfo(PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)</a>
void	<a href="#">onConnect(PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)</a>
void	<a href="#">onConnectFailure(PushPublishRTMPNetConnectionSession pushPublisherSession)</a>
void	<a href="#">onConnectStart(PushPublishRTMPNetConnectionSession pushPublisherSession)</a>
void	<a href="#">onConnectSuccess(PushPublishRTMPNetConnectionSession pushPublisherSession)</a>
void	<a href="#">onFCAnnounce(PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)</a>
void	<a href="#">onFCPublish(PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)</a>
void	<a href="#">onHandshakeResult(PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)</a>
void	<a href="#">onPublishHandlerPlay(PushPublishRTMPNetConnectionSession pushPublisherSession, java.io.OutputStream out, long[] playSizes)</a>
void	<a href="#">onPushPublisherSessionCreate(PushPublishRTMPNetConnectionSession pushPublisherSession)</a>

void	<a href="#">onPushPublisherSessionDestroy(PushPublishRTMPNetConnectionSession pushPublisherSession)</a>
void	<a href="#">onReleaseStream(PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)</a>
void	<a href="#">onSessionClosed(PushPublishRTMPNetConnectionSession pushPublisherSession)</a>
void	<a href="#">onSessionIdle(PushPublishRTMPNetConnectionSession pushPublisherSession)</a>
void	<a href="#">onSessionOpened(PushPublishRTMPNetConnectionSession pushPublisherSession)</a>
void	<a href="#">onStreamCreate(PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)</a>
void	<a href="#">onStreamOnPlayStatus(PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)</a>
void	<a href="#">onStreamOnStatus(PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)</a>
void	<a href="#">onValidateSession(PushPublishRTMPNetConnectionSession pushPublisherSession)</a>
void	<a href="#">onValidateSessionResult(PushPublishRTMPNetConnectionSession pushPublisherSession, boolean result)</a>

## Methods

### onConnectStart

```
public void onConnectStart(PushPublishRTMPNetConnectionSession pushPublisherSession)
```

### onConnectSuccess

```
public void onConnectSuccess(PushPublishRTMPNetConnectionSession pushPublisherSession)
```

### onConnectFailure

```
public void onConnectFailure(PushPublishRTMPNetConnectionSession pushPublisherSession)
```

(continued from last page)

## onHandshakeResult

```
public void onHandshakeResult(PushPublishRTMPNetConnectionSession
pushPublisherSession,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

---

## onAkamaiSetChallenge

```
public void onAkamaiSetChallenge(PushPublishRTMPNetConnectionSession
pushPublisherSession,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

---

## onAkamaiSetOriginConnectionInfo

```
public void onAkamaiSetOriginConnectionInfo(PushPublishRTMPNetConnectionSession
pushPublisherSession,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

---

## onAkamaiClientLogin

```
public void onAkamaiClientLogin(PushPublishRTMPNetConnectionSession
pushPublisherSession,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

---

## onFCPublish

```
public void onFCPublish(PushPublishRTMPNetConnectionSession pushPublisherSession,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

---

## onFCAnnounce

```
public void onFCAnnounce(PushPublishRTMPNetConnectionSession pushPublisherSession,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

---

## onReleaseStream

```
public void onReleaseStream(PushPublishRTMPNetConnectionSession pushPublisherSession,
    com.wowza.wms.request.RequestFunction function,
    AMFDataList params)
```

---

(continued from last page)

### onConnect

```
public void onConnect(PushPublishRTMPNetConnectionSession pushPublisherSession,  
com.wowza.wms.request.RequestFunction function,  
AMFDataList params)
```

---

### onStreamCreate

```
public void onStreamCreate(PushPublishRTMPNetConnectionSession pushPublisherSession,  
com.wowza.wms.request.RequestFunction function,  
AMFDataList params)
```

---

### onStreamOnPlayStatus

```
public void onStreamOnPlayStatus(PushPublishRTMPNetConnectionSession pushPublisherSession,  
com.wowza.wms.request.RequestFunction function,  
AMFDataList params)
```

---

### onStreamOnStatus

```
public void onStreamOnStatus(PushPublishRTMPNetConnectionSession pushPublisherSession,  
com.wowza.wms.request.RequestFunction function,  
AMFDataList params)
```

---

### onPublishHandlerPlay

```
public void onPublishHandlerPlay(PushPublishRTMPNetConnectionSession pushPublisherSession,  
java.io.OutputStream out,  
long[] playSizes)
```

---

### onSessionIdle

```
public void onSessionIdle(PushPublishRTMPNetConnectionSession pushPublisherSession)
```

---

### onSessionOpened

```
public void onSessionOpened(PushPublishRTMPNetConnectionSession pushPublisherSession)
```

---

### onSessionClosed

```
public void onSessionClosed(PushPublishRTMPNetConnectionSession pushPublisherSession)
```

---

## onValidateSession

```
public void onValidateSession(PushPublishRTMPNetConnectionSession  
pushPublisherSession)
```

---

## onValidateSessionResult

```
public void onValidateSessionResult(PushPublishRTMPNetConnectionSession  
pushPublisherSession,  
boolean result)
```

---

## onPushPublisherSessionCreate

```
public void onPushPublisherSessionCreate(PushPublishRTMPNetConnectionSession  
pushPublisherSession)
```

---

## onPushPublisherSessionDestroy

```
public void onPushPublisherSessionDestroy(PushPublishRTMPNetConnectionSession  
pushPublisherSession)
```

## com.wowza.wms.pushpublish.protocol.rtmp Class PushPublishRTMP

```
java.lang.Object
+- com.wowza.wms.pushpublish.model.PushPublishBase
    +- com.wowza.wms.pushpublish.protocol.rtmp.PushPublishRTMP
```

All Implemented Interfaces:

[IPushPublish](#)

Direct Known Subclasses:

[PushPublishRTMPPProfileHandler](#)

---

public class **PushPublishRTMP**

extends [PushPublishBase](#)

PushPublisher

Sample Code to push to Wowza Server or Adobe Media Server (no authentication)

This method should work with **Level 3** as well as long as the publishing application is not authenticated.

```
try
{
    PushPublisherRTMP publisher = new PushPublisherRTMP();

    // Source stream
    publisher.setAppInstance(appInstance);
    publisher.setSrcStreamName("myStream");

    // Destination stream
    publisher.setHostname("localhost");
    publisher.setPort(1935);
    publisher.setDstApplicationName("live");
    publisher.setDstStreamName("myStream");

    // Uncomment if pushing to Adobe Media Server
    //publisher.setSendOriginalTimecodes(true);
    //publisher.setOriginalTimecodeThreshold(0x100000);

    // SecureToken shared secret
    //publisher.setSecureTokenSharedSecret("#ed%h0#w@1");

    publisher.setSendFCPublish(true);
    publisher.setSendReleaseStream(true);
    publisher.setDebugLog(true);

    publisher.connect();
}
catch(Exception e)
{
    WMSLoggerFactory.getLogger(null).info(CLASSNAME+"RTMP: ", e);
}
```

## Sample Code to push to Wowza Server or Adobe Media Server (RTMP username/password authentication)

This method should work with **Level 3** as well if the publishing application is protected using RTMP authentication.

```

try
{
    PushPublisherRTMP publisher = new PushPublisherRTMP();

    // Source stream
    publisher.setAppInstance(appInstance);
    publisher.setSrcStreamName("myStream");

    // Destination stream
    publisher.setHostname("localhost");
    publisher.setPort(1935);
    publisher.setDstApplicationName("live");
    publisher.setDstStreamName("myStream");
    publisher.setConnectionFlashVersion(PushPublisherRTMP.CURRENTFMLEVERSION);

    // Uncomment if pushing to Adobe Media Server
    //publisher.setSendOriginalTimecodes(true);
    //publisher.setOriginalTimecodeThreshold(0x100000);

    // SecureToken shared secret
    //publisher.setSecureTokenSharedSecret("#ed%h0#w@1");

    publisher.setSendFCPublish(true);
    publisher.setSendReleaseStream(true);
    publisher.setDebugLog(true);

    PushPublishRTMPAuthProviderAdobe adobeRTMPAuthProvider = new
PushPublishRTMPAuthProviderAdobe();

    adobeRTMPAuthProvider.init(publisher);
    adobeRTMPAuthProvider.setUserName("username");
    adobeRTMPAuthProvider.setPassword("password");
    publisher.setRTMPAuthProvider(adobeRTMPAuthProvider);

    publisher.connect();
}
catch(Exception e)
{
    WMSLoggerFactory.getLogger(null).info(CLASSNAME+"RTMP: ", e);
}

```

## Sample Code to push to Akamai

```

try
{
    PushPublisherRTMP publisher = new PushPublisherRTMP();

    // Source stream
    publisher.setAppInstance(appInstance);
    publisher.setSrcStreamName("myStream");

    // Destination stream
    String streamId = "12345";
    String akamaiUsername = "56789";
    String akamaiPassword = "changeme";
    boolean isPrimary = true;
    String hostname = (isPrimary?"p":"b")+".ep"+streamId+".i.akamaientrypoint.net";
    String dstApplicationName = "EntryPoint";
    String dstStreamName = "myStream"+"@"+streamId;

    publisher.setHostname(hostname);
    publisher.setPort(1935);
    publisher.setDstApplicationName(dstApplicationName);
    publisher.setDstStreamName(dstStreamName);
    publisher.setDebugLog(true);

    publisher.setConnectionFlashVersion(PushPublisherRTMP.getPushPublishVersionStr());

    if (PushPublisherRTMP.isFlashVersionFMLE(publisher.getConnectionFlashVersion()))
    {
        PushPublishRTMPAuthProviderAdobe adobeRTMPAuthProvider = new
PushPublishRTMPAuthProviderAdobe();

        adobeRTMPAuthProvider.init(publisher);
        adobeRTMPAuthProvider.setUserName(akamaiUsername);
        adobeRTMPAuthProvider.setPassword(akamaiPassword);

        publisher.setRTMPAuthProvider(adobeRTMPAuthProvider);
    }
    else
    {
        publisher.setAkamaiUserName(akamaiUsername);
        publisher.setAkamaiPassword(akamaiPassword);
    }

    publisher.setSendFCPublish(true);
    publisher.setSendReleaseStream(true);
    publisher.setSendStreamCloseCommands(true);

    publisher.setSendOriginalTimecodes(true);
    publisher.setOriginalTimecodeThreshold(0x100000);

    publisher.connect();
}

```

```
catch(Exception e)
{
    WMSLoggerFactory.getLogger(null).info(CLASSNAME+"RTMP: ", e);
}
```

## Sample Code to push to **Limelight**

```
try
{
    PushPublisherRTMP publisher = new PushPublisherRTMP();

    // Source stream
    publisher.setAppInstance(appInstance);
    publisher.setSrcStreamName("myStream");

    // Destination stream
    publisher.setHostname("fmspush.lax.llnw.net");
    publisher.setPort(1935);
    publisher.setDstApplicationName("live");
    publisher.setDstStreamName("myStream");

    publisher.setSendFCPublish(true);
    publisher.setSendReleaseStream(true);
    publisher.setDebugLog(true);

    publisher.setSendOriginalTimecodes(true);
    publisher.setOriginalTimecodeThreshold(0x100000);

    PushPublishRTMPAuthProviderLimelight limelightRTMPAuthProvider = new
PushPublishRTMPAuthProviderLimelight();

    limelightRTMPAuthProvider.init(publisher);
    limelightRTMPAuthProvider.setUserName("username");
    limelightRTMPAuthProvider.setPassword("password");
    publisher.setRTMPAuthProvider(limelightRTMPAuthProvider);

    publisher.connect();
}
catch(Exception e)
{
    WMSLoggerFactory.getLogger(null).info(CLASSNAME+"RTMP: ", e);
}
```

Sample Code to push to **BitGravity** (streaming needs to be setup per instructions in **BG Live 3rd Party Encoder Configuration** guide.)

```

try
{
    PushPublisherRTMP publisher = new PushPublisherRTMP();

    // Source stream
    publisher.setAppInstance(client.getAppInstance());
    publisher.setSrcStreamName("myStream");

    // Destination stream
    publisher.setHostname("rtpdev1.iad1.bitgravity.com");
    publisher.setPort(1935);
    publisher.setDstApplicationName("rtmp");
    publisher.setDstAppInstanceName("push");

    publisher.setDstStreamName("test@test.com/71d73d4cfde2f2fed77238021a2cbbe/test/live/feed01")
    ;

    publisher.setSendFCPublish(true);
    publisher.setSendReleaseStream(true);
    publisher.setDebugLog(true);

    publisher.connect();
}
catch(Exception e)
{
    WMSLoggerFactory.getLogger(null).info(CLASSNAME+"RTMP: ", e);
}

```

## Field Summary

protected	<a href="#">akamaiOriginIp</a>
protected	<a href="#">akamaiOriginPort</a>
protected	<a href="#">akamaiPassword</a>
protected	<a href="#">akamaiUserName</a>
public	<a href="#">bytesAvg</a>
public static final	<a href="#">CLASS</a>

public static final	<u>CLASSNAME</u> Value: <b>PushPublishRTMP</b>
protected	<u>connectionFlashVersion</u>
protected	<u>connectionPageURL</u>
protected	<u>connectionQueryStr</u>
protected	<u>connectionSwfURL</u>
protected	<u>connectionTimeout</u>
protected	<u>connectLastAttempt</u>
protected	<u>connectLastSuccess</u>
protected	<u>connectMetaData</u>
public static final	<u>CURRENTFLASHVERSION</u> Value: <b>WIN 10,0,12,36</b>
public static final	<u>CURRENTFMLEVERSION</u> Value: <b>FMLE/3.0 (compatible; FMSc/1.0)</b>
protected	<u>debugLogLastPacketCount</u>
protected	<u>debugPackets</u>
protected	<u>doSendOnMetaData</u>
protected	<u>dstAppInstanceName</u>
protected	<u>dstApplicationName</u>
protected	<u>dstStreamName</u>
protected	<u>howToPublish</u>
protected	<u>ioSessionToSessionHolder</u>
protected	<u>isReconnectorRunning</u>
protected	<u>isSSL</u>
protected	<u>lastAudioTC</u>
protected	<u>lastDataTC</u>

protected	<a href="#">lastIdleReadTime</a>
protected	<a href="#">lastIdleWriteTime</a>
protected	<a href="#">lastVideoTC</a>
protected	<a href="#">listeners</a>
protected	<a href="#">lock</a>
protected	<a href="#">netconnectToSessionHolder</a>
public	<a href="#">offlineTime</a>
public	<a href="#">onlineTime</a>
protected	<a href="#">onMetadataToSetDataFrame</a>
public	<a href="#">outprint</a>
protected	<a href="#">packetsSentAudio</a>
protected	<a href="#">packetsSentData</a>
protected	<a href="#">packetsSentVideo</a>
protected	<a href="#">removeDefaultAppInstance</a>
protected	<a href="#">resetOnTimecodeOutOfOrder</a>
protected	<a href="#">rtmpAuthProvider</a>
protected	<a href="#">rtmpDebugLog</a>
protected	<a href="#">secureTokenSharedSecret</a>
protected	<a href="#">sendFCAnnounce</a>
protected	<a href="#">sendFCPublish</a>
protected	<a href="#">sendOnMetadata</a>
protected	<a href="#">sendOriginalTimecodes</a>
protected	<a href="#">sendReleaseStream</a>
protected	<a href="#">sendStreamCloseCommands</a>

protected	<a href="#">sendStreamCloseCommandsSendTime</a>
protected	<a href="#">sessionOrder</a>
protected	<a href="#">srcStreamBufferSize</a>
protected	<a href="#">streamCloseWaitTime</a>
protected	<a href="#">timecodesOutOfOrderThreshold</a>
protected	<a href="#">totalPacketsSent</a>
protected	<a href="#">tryConnect</a>
protected	<a href="#">validationFrequency</a>
protected	<a href="#">waitOnMetadataAudio</a>
protected	<a href="#">waitOnMetadataAudioDatarate</a>
protected	<a href="#">waitOnMetadataAvailable</a>
protected	<a href="#">waitOnMetadataReady</a>
protected	<a href="#">waitOnMetadataStartime</a>
protected	<a href="#">waitOnMetadataTimeout</a>
protected	<a href="#">waitOnMetadataVideo</a>
protected	<a href="#">waitOnMetadataVideoDatarate</a>

#### Fields inherited from class [com.wowza.wms.pushpublish.model.PushPublishBase](#)

[appInstance](#), [CLASS](#), [CLASSNAME](#), [dstStreamName](#), [hostname](#), [mapFileIndex](#), [mImplementation](#), [mType](#), [originalTimecodeThreshold](#), [password](#), [port](#), [properties](#), [pushPublishStreamDebug](#), [pushPublishType](#), [reconnectWaitTime](#), [sendingPerfomance](#), [sendOriginalTimecodes](#), [srcStream](#), [srcStreamName](#), [userName](#)

#### Fields inherited from interface [com.wowza.wms.pushpublish.model.IPushPublish](#)

[PUSHPUBLISH\\_TYPE\\_RTMP](#), [PUSHPUBLISH\\_TYPE\\_RTP](#), [PUSHPUBLISH\\_TYPE\\_UNKNOWN](#)

## Constructor Summary

public	<a href="#">PushPublishRTMP()</a>
--------	-----------------------------------

## Method Summary

void	<a href="#"><u>addConnectMetaData</u></a> (String key, <a href="#"><u>AMFData</u></a> value) Added metadata values to the connect metadata object
void	<a href="#"><u>addListener</u></a> ( <a href="#"><u>IPushPublishRTMPNotify</u></a> listener) Add IPushPublisherRTMPNotify listener
void	<a href="#"><u>addSession</u></a> ( <a href="#"><u>PushPublishRTMPNetConnectionSession</u></a> sessionHolder) Add session
void	<a href="#"><u>callConnect</u></a> ( <a href="#"><u>PushPublishRTMPNetConnectionSession</u></a> pushPublisherSession, String addQueryStr) Internal connect
void	<a href="#"><u>connect</u></a> () Connect to server
<a href="#"><u>AMFDataList</u></a>	<a href="#"><u>convertDataEventToAMFData</u></a> ( <a href="#"><u>AMFPacket</u></a> packet)
<a href="#"><u>AMFPacket</u></a>	<a href="#"><u>convertOnMetadataToSetDataFrame</u></a> ( <a href="#"><u>AMFPacket</u></a> packet)
<a href="#"><u>IPushPublishSession</u></a>	<a href="#"><u>createPushPublishSession</u></a> ()
void	<a href="#"><u>disconnect</u></a> () Disconnect from server
String	<a href="#"><u>getAkamaiOriginIp</u></a> () Get Akamai Origin IP
int	<a href="#"><u>getAkamaiOriginPort</u></a> () Get Akamai Origin Port
String	<a href="#"><u>getAkamaiPassword</u></a> () Get Akamai password
String	<a href="#"><u>getAkamaiSessionKey</u></a> (String sessionId) Get Akamai session key
String	<a href="#"><u>getAkamaiUserName</u></a> () Get Akamai user name
<a href="#"><u>IApplicationInstance</u></a>	<a href="#"><u>getAppInstance</u></a> () Get appInstance interface
String	<a href="#"><u>getConnectionFlashVerion</u></a> () Get Flash version used in connection metadata
String	<a href="#"><u>getConnectionFlashVersion</u></a> () Get Flash version used in connection metadata
String	<a href="#"><u>getConnectionPageURL</u></a> () Get Page URL used in connection metadata
String	<a href="#"><u>getConnectionQueryStr</u></a> () Get the connection query string
String	<a href="#"><u>getConnectionSwfURL</u></a> () Get SWF URL used in connection metadata

int	<a href="#"><u>getConnectionTimeout()</u></a> Get the connection timeout (milliseconds)
long	<a href="#"><u>getConnectLastAttempt()</u></a> Get timestamp of last connection attempt
long	<a href="#"><u>getConnectLastSuccess()</u></a> Get timestamp of last successful connection
String	<a href="#"><u>getContextStr()</u></a> Get the connection details
String	<a href="#"><u>getDstAppInstanceId()</u></a> Get destination appInstance name
String	<a href="#"><u>getDstApplicationName()</u></a> Get destination application name
String	<a href="#"><u>getDstStreamName()</u></a> Get destination stream name
String	<a href="#"><u>getHostname()</u></a> Get destination server host name
String	<a href="#"><u>getHowToPublish()</u></a> Get the howToPublish string (valid values are "live", "record" and "append")
long	<a href="#"><u>getLastAudioTC()</u></a> Get the absolute timecode of the last sent audio packet (milliseconds)
long	<a href="#"><u>getLastDataTC()</u></a> Get the absolute timecode of the last sent data packet (milliseconds)
<a href="#"><u>PushPublishRTMPNetConnectionSession</u></a>	<a href="#"><u>getLastSession()</u></a> Get the last session interface
long	<a href="#"><u>getLastVideoTC()</u></a> Get the absolute timecode of the last sent video packet (milliseconds)
java.util.List	<a href="#"><u>getLocalListeners()</u></a> Get a list of IPushPublisherRTMPNotify listeners
Object	<a href="#"><u>getLock()</u></a> Get the synchronization lock
<a href="#"><u>com.wowza.wms.netconnection.INetConnection</u></a>	<a href="#"><u>getNetConnection()</u></a> Get the underlying INetConnection interface for this session
long	<a href="#"><u>getOriginalTimecodeThreshold()</u></a> Get the original timecode threshold.
long	<a href="#"><u>getPacketsSentAudio()</u></a>
long	<a href="#"><u>getPacketsSentData()</u></a>
long	<a href="#"><u>getPacketsSentVideo()</u></a>

int	<a href="#"><u>getPort()</u></a> Get destination server port
static String	<a href="#"><u>getPushPublishVersionStr()</u></a>
int	<a href="#"><u>getReconnectDelay(long currTime)</u></a> Get the reconnect delay
<a href="#"><u>IPushPublishRTMPAuthProvider</u></a>	<a href="#"><u>getRTMPAuthProvider()</u></a> Get RTMP authentication provider
String	<a href="#"><u>getSecureTokenSharedSecret()</u></a> Get SecureToken shared secret
<a href="#"><u>PushPublishRTMPNetConnectionSession</u></a>	<a href="#"><u>getSessionHolderByNetConnection(com.wowza.wms.netconnection.INetConnection netConnection)</u></a> Get the session holder by INetConnection interface
<a href="#"><u>PushPublishRTMPNetConnectionSession</u></a>	<a href="#"><u>getSessionHolderBySession(org.apache.mina.common.IoSession session)</u></a> Get session holder by IoSession
String	<a href="#"><u>getSessionListDebug()</u></a> Get the list of active IoSession ids for this publishing session
long	<a href="#"><u>getSrcStreamBufferSize()</u></a> The amount of audio/video in milliseconds that must be available in the source streams buffer before push publishing will begin.
String	<a href="#"><u>getSrcStreamName()</u></a> Get source stream name
int	<a href="#"><u>getStreamCloseWaitTime()</u></a> Get stream close timeout (milliseconds)
int	<a href="#"><u>getTimecodesOutOfOrderThreshold()</u></a>
long	<a href="#"><u>getTotalPacketsSent()</u></a> Get total number of packets sent during push publishing session
long	<a href="#"><u>getValidationFrequency()</u></a> Get connection validation frequency (millisecond).
long	<a href="#"><u>getWaitOnMetadataTimeout()</u></a> Timeout (milliseconds) that the wait on onMetadata system will wait looking for video and audio data to be populated.
org.apache.mina.common.IoSession	<a href="#"><u>internalConnect(PushPublishRTMPNetConnectionSession pushPublisherSession)</u></a> Internal connect method
boolean	<a href="#"><u>isCurrentSession(org.apache.mina.common.IoSession session)</u></a> Test to see if passed in session is current session
boolean	<a href="#"><u>isDebugEnabled()</u></a> Is debug logging turned on
boolean	<a href="#"><u>isDebugEnabled()</u></a> Set to true to log every packet being sent

static boolean	<a href="#"><u>isFlashVerionFMLE</u></a> (String flashVersion) Returns true if the Flash player version string is a Flash Media Live Encoder version (start with FMLE/ or FME/)
static boolean	<a href="#"><u>isFlashVersionFMLE</u></a> (String flashVersion) Returns true if the Flash player version string is a Flash Media Live Encoder version (start with FMLE/ or FME/)
boolean	<a href="#"><u>isOnMetadataToSetDataFrame</u></a> ( ) If true all onMetadata events are converted to @setDataFrame calls
boolean	<a href="#"><u>isRemoveDefaultAppInstance</u></a> ( ) If true and destination application instance is _definst_ then the application instance name will not be sent to the remove server as part of the connection information.
boolean	<a href="#"><u>isResetOnTimecodeOutOfOrder</u></a> ( ) Set to true if you wish push to reset if timecodes jump out of order.
boolean	<a href="#"><u>isSendFCAnnounce</u></a> ( ) If true call FCAnnounce(streamName) after connecting to server
boolean	<a href="#"><u>isSendFCPublish</u></a> ( ) If true call FCPublish(streamName) after connecting to server
boolean	<a href="#"><u>isSendOnMetadata</u></a> ( ) True if sending onMetadata event at start of stream
boolean	<a href="#"><u>isSendOriginalTimecodes</u></a> ( ) true if sending the original timecodes of source stream.
boolean	<a href="#"><u>isSendReleaseStream</u></a> ( ) If true call releaseStream(streamName) after connecting to server
boolean	<a href="#"><u>isSendStreamCloseCommands</u></a> ( ) If true will send FCUnpublish, closeStream, deleteStream on disconnect
boolean	<a href="#"><u>isSSL</u></a> ( )
boolean	<a href="#"><u>isWaitOnMetadataAudio</u></a> ( ) If true and waitOnMetadataAvailable, publishing will not start until onMetadata packet is populated with audio data.
boolean	<a href="#"><u>isWaitOnMetadataAudioDatarate</u></a> ( )
boolean	<a href="#"><u>isWaitOnMetadataAvailable</u></a> ( ) Set to true to turn on system that will monitor onMetadata packet and wait for it to be populated before it starts sending the stream.
boolean	<a href="#"><u>isWaitOnMetadataVideo</u></a> ( ) If true and waitOnMetadataAvailable, publishing will not start until onMetadata packet is populated with video data.
boolean	<a href="#"><u>isWaitOnMetadataVideoDatarate</u></a> ( )
void	<a href="#"><u>notifyAkamaiClientLogin</u></a> (PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, <a href="#"><u>AMFDatalist</u></a> params) Notify Akamai ClientLogin

void	<code>notifyAkamaiSetChallenge(PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)</code> Notify Akamai SetChallenge
void	<code>notifyAkamaiSetOriginConnectionInfo(PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)</code> Notify Akamai SetOriginConnectionInfo
void	<code>notifyConnect(PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)</code> Notify connect
void	<code>notifyConnectFailure(PushPublishRTMPNetConnectionSession pushPublisherSession)</code> Notify connect failure
void	<code>notifyConnectStart(PushPublishRTMPNetConnectionSession pushPublisherSession)</code> Notify connect start
void	<code>notifyConnectSuccess(PushPublishRTMPNetConnectionSession pushPublisherSession)</code> Notify connect success
void	<code>notifyFCAnnounce(PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)</code> Notify FCAnnounce
void	<code>notifyFCPublish(PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)</code> Notify FCPublish
void	<code>notifyHandshakeResult(PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)</code> Notify handshake result
void	<code>notifyPublishHandlerPlay(PushPublishRTMPNetConnectionSession pushPublisherSession, java.io.OutputStream out, long[] playSizes)</code> Notify publish handler play
void	<code>notifyPushPublisherSessionCreate(PushPublishRTMPNetConnectionSession pushPublisherSession)</code> Notify session create
void	<code>notifyPushPublisherSessionDestroy(PushPublishRTMPNetConnectionSession pushPublisherSession)</code> Notify session destroy
void	<code>notifySessionClosed(PushPublishRTMPNetConnectionSession pushPublisherSession)</code> Notify session closed
void	<code>notifySessionIdle(PushPublishRTMPNetConnectionSession pushPublisherSession)</code> Notify session idle event
void	<code>notifySessionOpened(PushPublishRTMPNetConnectionSession pushPublisherSession)</code> Notify session opened

void	<code>notifyStreamCreate(PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)</code> Notify stream create
void	<code>notifyStreamOnPlayStatus(PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)</code> Notify OnPlayStatus
void	<code>notifyStreamOnStatus(PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)</code> Notify StreamOnStatus
void	<code>notifyValidateSession(PushPublishRTMPNetConnectionSession pushPublisherSession)</code>
void	<code>notifyValidateSessionResult(PushPublishRTMPNetConnectionSession pushPublisherSession, boolean result)</code>
boolean	<code>removeListener(IPushPublishRTMPNotify listener)</code> remove IPushPublisherRTMPNotify listener
void	<code>removeSession(PushPublishRTMPNetConnectionSession sessionHolder)</code> Remove session
void	<code>resetConnection()</code> Reset server connection
void	<code>resetConnection(boolean noDelay)</code> Reset server connection
void	<code>sessionClosed(org.apache.mina.common.IoSession session)</code> Internal callback
void	<code>sessionOpened(org.apache.mina.common.IoSession session)</code> Internal callback
void	<code>setAkamaiOriginIp(String akamaiOriginIp)</code> Set Akamai Origin IP
void	<code>setAkamaiOriginPort(int akamaiOriginPort)</code> Set Akamai Origin Port
void	<code>setAkamaiPassword(String akamaiPassword)</code> Set Akamai password
void	<code>setAkamaiUserName(String akamaiUserName)</code> Set Akamai user name
void	<code>setAppInstance(IApplicationInstance appInstance)</code> Set appInstance interface
void	<code>setConnectionFlashVerion(String connectionFlashVersion)</code> Set Flash version used in connection metadata
void	<code>setConnectionFlashVersion(String connectionFlashVersion)</code> Set Flash version used in connection metadata

void	<a href="#"><u>setConnectionPageURL</u></a> (String connectionPageURL) Set Page URL used in connection metadata
void	<a href="#"><u>setConnectionQueryStr</u></a> (String connectionQueryStr) Set the connection query string
void	<a href="#"><u>setConnectionSwfURL</u></a> (String connectionSwfURL) Set SWF URL used in connection metadata
void	<a href="#"><u>setConnectionTimeout</u></a> (int connectionTimeout) Set the connection timeout (milliseconds)
void	<a href="#"><u>setConnectLastAttempt</u></a> (long connectLastAttempt) Set timestamp of last connection attempt
void	<a href="#"><u>setConnectLastSuccess</u></a> (long connectLastSuccess) Set timestamp of last successful connection
void	<a href="#"><u>setDebugLog</u></a> (boolean debugLog) Set debug logging
void	<a href="#"><u>setDebugPackets</u></a> (boolean debugPackets) Set to true to log every packet being sent
void	<a href="#"><u>setDstAppInstanceName</u></a> (String dstAppInstanceName) Set destination appInstance name
void	<a href="#"><u>setDstApplicationName</u></a> (String dstApplicationName) Set destination application name
void	<a href="#"><u>setDstStreamName</u></a> (String dstStreamName) Set destination stream name
void	<a href="#"><u>setHostname</u></a> (String hostname) Set destination server host name
void	<a href="#"><u>setHowToPublish</u></a> (String howToPublish) Set the howToPublish string (valid values are "live", "record" and "append")
void	<a href="#"><u>setOnMetadataToSetDataFrame</u></a> (boolean onMetadataToSetDataFrame) If true all onMetadata events are converted to @setDataFrame calls
void	<a href="#"><u>setOriginalTimecodeThreshold</u></a> (long originalTimecodeThreshold) Set the original timecode threshold.
void	<a href="#"><u>setPort</u></a> (int port) Set destination server port
void	<a href="#"><u>setRemoveDefaultAppInstance</u></a> (boolean removeDefaultAppInstance) If true and destination application instance is _definst_ then the application instance name will not be sent to the remove server as part of the connection information.
void	<a href="#"><u>setResetOnTimecodeOutOfOrder</u></a> (boolean resetOnTimecodeOutOfOrder) Set to true if you wish push to reset if timecodes jump out of order.
void	<a href="#"><u>setRTMPAuthProvider</u></a> ( <a href="#">IPushPublishRTMPAuthProvider</a> rtmpAuthProvider) Set RTMP authentication provider
void	<a href="#"><u>setSecureTokenSharedSecret</u></a> (String secureTokenSharedSecret) Set SecureToken shared secret

void	<a href="#"><u>setSendFCAnnounce(boolean sendFCAnnounce)</u></a> If true call FCAnnounce(streamName) after connecting to server
void	<a href="#"><u>setSendFCPublish(boolean sendFCPublish)</u></a> If true call FCPublish(streamName) after connecting to server
void	<a href="#"><u>setSendOnMetadata(boolean sendOnMetadata)</u></a> True if sending onMetadata event at start of stream
void	<a href="#"><u>setSendOriginalTimecodes(boolean sendOriginalTimecodes)</u></a> true if sending the original timecodes of source stream.
void	<a href="#"><u>setSendReleaseStream(boolean sendReleaseStream)</u></a> If true call releaseStream(streamName) after connecting to server
void	<a href="#"><u>setSendStreamCloseCommands(boolean sendStreamCloseCommands)</u></a> If true will send FCUnpublish, closeStream, deleteStream on disconnect
void	<a href="#"><u>setSrcStreamBufferSize(long srcStreamBufferSize)</u></a> The amount of audio/video in milliseconds that must be available in the source streams buffer before push publishing will begin.
void	<a href="#"><u>setSrcStreamName(String srcStreamName)</u></a> Set source stream name
void	<a href="#"><u>setSSL(boolean isSSL)</u></a>
void	<a href="#"><u>setStreamCloseWaitTime(int streamCloseWaitTime)</u></a> Set stream close timeout (milliseconds)
void	<a href="#"><u>setTimecodesOutOfOrderThreshold(int timecodesOutOfOrderThreshold)</u></a>
void	<a href="#"><u>setValidationFrequency(long validationFrequency)</u></a> Set connection validation frequency (millisecond).
void	<a href="#"><u>setWaitOnMetadataAudio(boolean waitOnMetadataAudio)</u></a> If true and waitOnMetadataAvailable, publishing will not start until onMetadata packet is populated with audio data.
void	<a href="#"><u>setWaitOnMetadataAudioDatarate(boolean waitOnMetadataAudioDatarate)</u></a>
void	<a href="#"><u>setWaitOnMetadataAvailable(boolean waitOnMetadataAvailable)</u></a> Set to true to turn on system that will monitor onMetadata packet and wait for it to be populated before it starts sending the stream.
void	<a href="#"><u>setWaitOnMetadataTimeout(long waitOnMetadataTimeout)</u></a> Timeout (milliseconds) that the wait on onMetadata system will wait looking for video and audio data to be populated.
void	<a href="#"><u>setWaitOnMetadataVideo(boolean waitOnMetadataVideo)</u></a> If true and waitOnMetadataAvailable, publishing will not start until onMetadata packet is populated with video data.
void	<a href="#"><u>setWaitOnMetadataVideoDatarate(boolean waitOnMetadataVideoDatarate)</u></a>
void	<a href="#"><u>tryToConnect()</u></a> Try to connect to server

Methods inherited from class [com.wowza.wms.pushpublish.model.PushPublishBase](#)

```
createPushPublishSession, getAppInstance, getBandwidth, getConfigDataOrDefault,
getConfigDataOrDefault, getConfigDataOrDefault, getContextStr, getDstStreamName,
getHostname, getImplementation, getMapFileIndex, getNumberFromString,
getOriginalTimecodeThreshold, getPassword, getPort, getProperties,
getPushPublishType, getPushPublishVersionStr, getReconnectWaitTime, getSrcStream,
getSrcStreamName, getUserName, init, isAdaptiveStreaming, isSendOriginalTimecodes,
load, setAdaptiveStreaming, setAppInstance, setDebugLog, setDstStreamName, setHost,
setHostname, setImplementation, setMapFileIndex, setOriginalTimecodeThreshold,
setOriginalTimecodeThreshold, setPassword, setPort, setPort, setReconnectWaitTime,
setSendOriginalTimecodes, setSrcStream, setSrcStreamName, setStreamName, setUserName
```

#### Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait,
wait
```

#### Methods inherited from interface com.wowza.wms.pushpublish.model.IPushPublish

```
connect, createPushPublishSession, disconnect, getContextStr, getDstStreamName,
getHostname, getMapFileIndex, getPort, getProperties, getPushPublishType,
getReconnectWaitTime, getSrcStream, getSrcStreamName, init, resetConnection,
setDebugLog, setDstStreamName, setHost, setHostname, setMapFileIndex, setPort,
setPort, setReconnectWaitTime, setSrcStream, setSrcStreamName, setStreamName
```

## Fields

### CLASS

```
public static final java.lang.Class CLASS
```

---

### CLASSNAME

```
public static final java.lang.String CLASSNAME
```

Constant value: **PushPublishRTMP**

---

### CURRENTFMLEVERSION

```
public static final java.lang.String CURRENTFMLEVERSION
```

Constant value: **FMLE/3.0 (compatible; FMSc/1.0)**

---

### CURRENTFLASHVERSION

```
public static final java.lang.String CURRENTFLASHVERSION
```

Constant value: **WIN 10,0,12,36**

(continued from last page)

**rtmpDebugLog**protected boolean **rtmpDebugLog****debugLogLastPacketCount**protected long **debugLogLastPacketCount****debugPackets**protected boolean **debugPackets****netconnectToSessionHolder**protected java.util.Map **netconnectToSessionHolder****ioSessionToSessionHolder**protected java.util.Map **ioSessionToSessionHolder****sessionOrder**protected java.util.List **sessionOrder****listeners**protected java.util.List **listeners****dstStreamName**protected java.lang.String **dstStreamName****dstApplicationName**protected java.lang.String **dstApplicationName****dstAppInstanceName**protected java.lang.String **dstAppInstanceName**

(continued from last page)

---

**lock**

```
protected java.lang.Object lock
```

---

**howToPublish**

```
protected java.lang.String howToPublish
```

---

**sendFCPublish**

```
protected boolean sendFCPublish
```

---

**sendFCAnnounce**

```
protected boolean sendFCAnnounce
```

---

**sendReleaseStream**

```
protected boolean sendReleaseStream
```

---

**sendStreamCloseCommands**

```
protected boolean sendStreamCloseCommands
```

---

**sendOnMetadata**

```
protected boolean sendOnMetadata
```

---

**onMetadataToSetDataFrame**

```
protected boolean onMetadataToSetDataFrame
```

---

**sendOriginalTimecodes**

```
protected boolean sendOriginalTimecodes
```

---

(continued from last page)

**timecodesOutOfOrderThreshold**protected int **timecodesOutOfOrderThreshold****tryConnect**protected boolean **tryConnect****isReconnectorRunning**protected boolean **isReconnectorRunning****connectionQueryStr**protected java.lang.String **connectionQueryStr****connectionFlashVersion**protected java.lang.String **connectionFlashVersion****connectionSwfURL**protected java.lang.String **connectionSwfURL****connectionPageURL**protected java.lang.String **connectionPageURL****secureTokenSharedSecret**protected java.lang.String **secureTokenSharedSecret****akamaiUserName**protected java.lang.String **akamaiUserName****akamaiPassword**protected java.lang.String **akamaiPassword**

(continued from last page)

---

**akamaiOriginIp**protected java.lang.String **akamaiOriginIp**

---

**akamaiOriginPort**protected java.lang.Integer **akamaiOriginPort**

---

**validationFrequency**protected long **validationFrequency**

---

**connectionTimeout**protected int **connectionTimeout**

---

**connectLastAttempt**protected long **connectLastAttempt**

---

**connectLastSuccess**protected long **connectLastSuccess**

---

**doSendOnMetaData**protected boolean **doSendOnMetaData**

---

**removeDefaultAppInstance**protected boolean **removeDefaultAppInstance**

---

**resetOnTimecodeOutOfOrder**protected boolean **resetOnTimecodeOutOfOrder**

(continued from last page)

**sendStreamCloseCommandsSendTime**protected long **sendStreamCloseCommandsSendTime****streamCloseWaitTime**protected int **streamCloseWaitTime****lastIdleReadTime**protected long **lastIdleReadTime****lastIdleWriteTime**protected long **lastIdleWriteTime****totalPacketsSent**protected long **totalPacketsSent****lastVideoTC**protected long **lastVideoTC****lastAudioTC**protected long **lastAudioTC****lastDataTC**protected long **lastDataTC****packetsSentAudio**protected long **packetsSentAudio****packetsSentVideo**protected long **packetsSentVideo**

(continued from last page)

---

**packetsSentData**

```
protected long packetsSentData
```

---

**waitOnMetadataAvailable**

```
protected boolean waitOnMetadataAvailable
```

---

**waitOnMetadataVideo**

```
protected boolean waitOnMetadataVideo
```

---

**waitOnMetadataVideoDatarate**

```
protected boolean waitOnMetadataVideoDatarate
```

---

**waitOnMetadataAudio**

```
protected boolean waitOnMetadataAudio
```

---

**waitOnMetadataAudioDatarate**

```
protected boolean waitOnMetadataAudioDatarate
```

---

**waitOnMetadataTimeout**

```
protected long waitOnMetadataTimeout
```

---

**waitOnMetadataReady**

```
protected boolean waitOnMetadataReady
```

---

**waitOnMetadataStartime**

```
protected long waitOnMetadataStartime
```

---

(continued from last page)

**srcStreamBufferSize**

```
protected long srcStreamBufferSize
```

**rtmpAuthProvider**

```
protected com.wowza.wms.pushpublish.protocol.rtmp.IPushPublishRTMPAuthProvider  
rtmpAuthProvider
```

**connectMetaData**

```
protected java.util.Map connectMetaData
```

**offlineTime**

```
public long offlineTime
```

**onlineTime**

```
public long onlineTime
```

**outprint**

```
public int outprint
```

**bytesAvg**

```
public float bytesAvg
```

**isSSL**

```
protected boolean isSSL
```

**Constructors****PushPublishRTMP**

```
public PushPublishRTMP()
```

**Methods**

(continued from last page)

## getNetConnection

```
public com.wowza.wms.netconnection.INetConnection getNetConnection()
```

Get the underlying INetConnection interface for this session

**Returns:**

INetConnection interface for this session

---

## getLock

```
public Object getLock()
```

Get the synchronization lock

**Returns:**

synchronization lock

---

## getConnectionTimeout

```
public int getConnectionTimeout()
```

Get the connection timeout (milliseconds)

**Returns:**

connection timeout (milliseconds)

---

## setConnectionTimeout

```
public void setConnectionTimeout(int connectionTimeout)
```

Set the connection timeout (milliseconds)

**Parameters:**

connectionTimeout - connection timeout (milliseconds)

---

## getHowToPublish

```
public String getHowToPublish()
```

Get the howToPublish string (valid values are "live", "record" and "append")

**Returns:**

howToPublish string

---

## setHowToPublish

```
public void setHowToPublish(String howToPublish)
```

Set the howToPublish string (valid values are "live", "record" and "append")

**Parameters:**

howToPublish - howToPublish string

---

## isSendFCPublish

```
public boolean isSendFCPublish()
```

---

(continued from last page)

If true call FCPublish(streamName) after connecting to server

**Returns:**

true if calling FCPublish(streamName) after connecting to server

---

## setSendFCPublish

```
public void setSendFCPublish(boolean sendFCPublish)
```

If true call FCPublish(streamName) after connecting to server

**Parameters:**

sendFCPublish - true if calling FCPublish(streamName) after connecting to server

---

## isSendReleaseStream

```
public boolean isSendReleaseStream()
```

If true call releaseStream(streamName) after connecting to server

**Returns:**

true if calling releaseStream(streamName) after connecting to server

---

## setSendReleaseStream

```
public void setSendReleaseStream(boolean sendReleaseStream)
```

If true call releaseStream(streamName) after connecting to server

**Parameters:**

sendReleaseStream - true if calling releaseStream(streamName) after connecting to server

---

## getSessionListDebug

```
public String getSessionListDebug()
```

Get the list of active IoSession ids for this publishing session

**Returns:**

list of active IoSession ids for this publishing session

---

## isCurrentSession

```
public boolean isCurrentSession(org.apache.mina.common.IoSession session)
```

Test to see if passed in session is current session

**Parameters:**

session - IoSession interface

**Returns:**

true if current session

---

## getLastSession

```
public PushPublishRTMPNetConnectionSession getLastSession()
```

Get the last session interface

(continued from last page)

**Returns:**

last session interface

---

**addSession**

```
public void addSession(PushPublishRTMPNetConnectionSession sessionHolder)
```

Add session

**Parameters:**

sessionHolder - session holder

---

**removeSession**

```
public void removeSession(PushPublishRTMPNetConnectionSession sessionHolder)
```

Remove session

**Parameters:**

sessionHolder - session holder

---

**getSessionHolderBySession**

```
public PushPublishRTMPNetConnectionSession
getSessionHolderBySession(org.apache.mina.common.IoSession session)
```

Get session holder by IoSession

**Parameters:**

session - IoSession

**Returns:**

session holder

---

**getSessionHolderByNetConnection**

```
public PushPublishRTMPNetConnectionSession
getSessionHolderByNetConnection(com.wowza.wms.netconnection.INetConnection
netConnection)
```

Get the session holder by INetConnection interface

**Parameters:**

netConnection - INetConnection interface

**Returns:**

session holder

---

**sessionOpened**

```
public void sessionOpened(org.apache.mina.common.IoSession session)
```

Internal callback

**Parameters:**

session - IoSession

(continued from last page)

## sessionClosed

```
public void sessionClosed(org.apache.mina.common.IoSession session)
```

Internal callback

**Parameters:**

session - IoSession

---

## tryToConnect

```
protected void tryToConnect()
```

Try to connect to server

---

## resetConnection

```
public void resetConnection()
```

Reset server connection

---

## resetConnection

```
public void resetConnection(boolean noDelay)
```

Reset server connection

**Parameters:**

noDelay - if true reset immediately

---

## getContextStr

```
public String getContextStr()
```

Get the connection details

**Returns:**

connection details

---

## internalConnect

```
protected org.apache.mina.common.IoSession
```

```
internalConnect(PushPublishRTMPNetConnectionSession pushPublisherSession)
```

Internal connect method

**Parameters:**

pushPublisherSession - session holder

**Returns:**

Io session

---

## getConnectionQueryStr

```
public String getConnectionQueryStr()
```

Get the connection query string

**Returns:**

(continued from last page)

connection query string

---

**setConnectionQueryStr**

```
public void setConnectionQueryStr(String connectionQueryStr)
```

Set the connection query string

**Parameters:**

connectionQueryStr - connection query string

---

**connect**

```
public void connect()
```

Connect to server

---

**callConnect**

```
protected void callConnect(PushPublishRTMPNetConnectionSession pushPublisherSession,  
String addQueryStr)
```

Internal connect

**Parameters:**

pushPublisherSession - session holder  
addQueryStr - additional query params

---

**getAkamaiSessionKey**

```
public String getAkamaiSessionKey(String sessionId)
```

Get Akamai session key

**Parameters:**

sessionId - session Id

**Returns:**

Akamai session key

---

**getConnectionFlashVerion**

```
public String getConnectionFlashVerion()
```

Get Flash version used in connection metadata

**Returns:**

Flash version used in connection metadata

---

**getConnectionFlashVersion**

```
public String getConnectionFlashVersion()
```

Get Flash version used in connection metadata

**Returns:**

Flash version used in connection metadata

---

## setConnectionFlashVersion

```
public void setConnectionFlashVersion(String connectionFlashVersion)
```

Set Flash version used in connection metadata

**Parameters:**

connectionFlashVersion - Flash version used in connection metadata

---

## setConnectionFlashVerion

```
public void setConnectionFlashVerion(String connectionFlashVersion)
```

Set Flash version used in connection metadata

**Parameters:**

connectionFlashVersion - Flash version used in connection metadata

---

## getConnectionSwfURL

```
public String getConnectionSwfURL()
```

Get SWF URL used in connection metadata

**Returns:**

SWF URL used in connection metadata

---

## setConnectionSwfURL

```
public void setConnectionSwfURL(String connectionSwfURL)
```

Set SWF URL used in connection metadata

**Parameters:**

connectionSwfURL - SWF URL used in connection metadata

---

## getConnectionPageURL

```
public String getConnectionPageURL()
```

Get Page URL used in connection metadata

**Returns:**

Page URL used in connection metadata

---

## setConnectionPageURL

```
public void setConnectionPageURL(String connectionPageURL)
```

Set Page URL used in connection metadata

**Parameters:**

connectionPageURL - Page URL used in connection metadata

---

## getSecureTokenSharedSecret

```
public String getSecureTokenSharedSecret()
```

---

(continued from last page)

Get SecureToken shared secret

**Returns:**

SecureToken shared secret

---

### **setSecureTokenSharedSecret**

```
public void setSecureTokenSharedSecret(String secureTokenSharedSecret)
```

Set SecureToken shared secret

**Parameters:**

secureTokenSharedSecret - SecureToken shared secret

---

### **getAkamaiUserName**

```
public String getAkamaiUserName()
```

Get Akamai user name

**Returns:**

Akamai user name

---

### **setAkamaiUserName**

```
public void setAkamaiUserName(String akamaiUserName)
```

Set Akamai user name

**Parameters:**

akamaiUserName - Akamai user name

---

### **getAkamaiPassword**

```
public String getAkamaiPassword()
```

Get Akamai password

**Returns:**

Akamai password

---

### **setAkamaiPassword**

```
public void setAkamaiPassword(String akamaiPassword)
```

Set Akamai password

**Parameters:**

akamaiPassword - Akamai password

---

### **getAkamaiOriginIp**

```
public String getAkamaiOriginIp()
```

Get Akamai Origin IP

**Returns:**

Akamai Origin IP

---

## setAkamaiOriginIp

```
public void setAkamaiOriginIp(String akamaiOriginIp)
```

Set Akamai Origin IP

**Parameters:**

akamaiOriginIp - Akamai Origin IP

---

## getAkamaiOriginPort

```
public int getAkamaiOriginPort()
```

Get Akamai Origin Port

**Returns:**

Akamai Origin Port

---

## setAkamaiOriginPort

```
public void setAkamaiOriginPort(int akamaiOriginPort)
```

Set Akamai Origin Port

**Parameters:**

akamaiOriginPort - Akamai Origin Port

---

## isSendFCAnnounce

```
public boolean isSendFCAnnounce()
```

If true call FCAnnounce(streamName) after connecting to server

**Returns:**

true if calling FCAnnounce(streamName) after connecting to server

---

## setSendFCAnnounce

```
public void setSendFCAnnounce(boolean sendFCAnnounce)
```

If true call FCAnnounce(streamName) after connecting to server

**Parameters:**

sendFCAnnounce - true if calling FCAnnounce(streamName) after connecting to server

---

## getValidationFrequency

```
public long getValidationFrequency()
```

Get connection validation frequency (millisecond). Zero to turn off validation.

**Returns:**

connection validation frequency (millisecond)

---

## setValidationFrequency

```
public void setValidationFrequency(long validationFrequency)
```

---

(continued from last page)

Set connection validation frequency (millisecond). Zero to turn off validation.

**Parameters:**

validationFrequency - connection validation frequency (millisecond)

---

## addListener

```
public void addListener(IPushPublishRTMPNotify listener)
```

Add IPushPublisherRTMPNotify listener

**Parameters:**

listener - IPushPublisherRTMPNotify listener

---

## removeListener

```
public boolean removeListener(IPushPublishRTMPNotify listener)
```

remove IPushPublisherRTMPNotify listener

**Parameters:**

listener - IPushPublisherRTMPNotify listener

**Returns:**

true if removed

---

## getLocalListeners

```
protected java.util.List getLocalListeners()
```

Get a list of IPushPublisherRTMPNotify listeners

**Returns:**

list of IPushPublisherRTMPNotify listeners

---

## notifyConnectStart

```
protected void notifyConnectStart(PushPublishRTMPNetConnectionSession  
pushPublisherSession)
```

Notify connect start

**Parameters:**

pushPublisherSession - session holder

---

## notifyConnectSuccess

```
protected void notifyConnectSuccess(PushPublishRTMPNetConnectionSession  
pushPublisherSession)
```

Notify connect success

**Parameters:**

pushPublisherSession - session holder

---

(continued from last page)

## notifyConnectFailure

```
protected void notifyConnectFailure(PushPublishRTMPNetConnectionSession  
pushPublisherSession)
```

Notify connect failure

**Parameters:**

pushPublisherSession - session holder

## notifyHandshakeResult

```
protected void notifyHandshakeResult(PushPublishRTMPNetConnectionSession  
pushPublisherSession,  
com.wowza.wms.request.RequestFunction function,  
AMFDataList params)
```

Notify handshake result

**Parameters:**

pushPublisherSession - session holder

function - function

params - params

## notifyAkamaiSetChallenge

```
protected void notifyAkamaiSetChallenge(PushPublishRTMPNetConnectionSession  
pushPublisherSession,  
com.wowza.wms.request.RequestFunction function,  
AMFDataList params)
```

Notify Akamai SetChallenge

**Parameters:**

pushPublisherSession - session holder

function - function

params - params

## notifyAkamaiSetOriginConnectionInfo

```
protected void notifyAkamaiSetOriginConnectionInfo(PushPublishRTMPNetConnectionSession  
pushPublisherSession,  
com.wowza.wms.request.RequestFunction function,  
AMFDataList params)
```

Notify Akamai SetOriginConnectionInfo

**Parameters:**

pushPublisherSession - session holder

function - function

params - params

## notifyAkamaiClientLogin

```
protected void notifyAkamaiClientLogin(PushPublishRTMPNetConnectionSession  
pushPublisherSession,  
com.wowza.wms.request.RequestFunction function,  
AMFDataList params)
```

Notify Akamai ClientLogin

(continued from last page)

**Parameters:**

pushPublisherSession - session holder  
function - function  
params - params

---

## notifyFCPublish

```
protected void notifyFCPublish(PushPublishRTMPNetConnectionSession  
pushPublisherSession,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Notify FCPublish

**Parameters:**

pushPublisherSession - session holder  
function - function  
params - params

---

## notifyFCAnnounce

```
protected void notifyFCAnnounce(PushPublishRTMPNetConnectionSession  
pushPublisherSession,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Notify FCAnnounce

**Parameters:**

pushPublisherSession - session holder  
function - function  
params - params

---

## notifyConnect

```
protected void notifyConnect(PushPublishRTMPNetConnectionSession pushPublisherSession,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Notify connect

**Parameters:**

pushPublisherSession - session holder  
function - function  
params - params

---

## notifyStreamCreate

```
protected void notifyStreamCreate(PushPublishRTMPNetConnectionSession  
pushPublisherSession,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

Notify stream create

**Parameters:**

pushPublisherSession - session holder  
function - function  
params - params

---

## notifyStreamOnPlayStatus

```
protected void notifyStreamOnPlayStatus(PushPublishRTMPNetConnectionSession
pushPublisherSession,
com.wowza.wms.request.RequestFunction function,
AMFDataList params)
```

Notify OnPlayStatus

**Parameters:**

pushPublisherSession - session holder  
function - function  
params - params

---

## notifyStreamOnStatus

```
protected void notifyStreamOnStatus(PushPublishRTMPNetConnectionSession
pushPublisherSession,
com.wowza.wms.request.RequestFunction function,
AMFDataList params)
```

Notify StreamOnStatus

**Parameters:**

pushPublisherSession - session holder  
function - function  
params - params

---

## notifyPublishHandlerPlay

```
protected void notifyPublishHandlerPlay(PushPublishRTMPNetConnectionSession
pushPublisherSession,
java.io.OutputStream out,
long[] playSizes)
```

Notify publish handler play

**Parameters:**

pushPublisherSession - session holder  
out - output stream  
playSizes - play sizes

---

## notifySessionIdle

```
protected void notifySessionIdle(PushPublishRTMPNetConnectionSession
pushPublisherSession)
```

Notify session idle event

**Parameters:**

pushPublisherSession - session holder

---

## notifySessionOpened

```
protected void notifySessionOpened(PushPublishRTMPNetConnectionSession
pushPublisherSession)
```

Notify session opened

(continued from last page)

**Parameters:**

pushPublisherSession - session holder

**notifySessionClosed**

```
protected void notifySessionClosed(PushPublishRTMPNetConnectionSession  
pushPublisherSession)
```

Notify session closed

**Parameters:**

pushPublisherSession - session holder

**notifyValidateSession**

```
protected void notifyValidateSession(PushPublishRTMPNetConnectionSession  
pushPublisherSession)
```

**notifyValidateSessionResult**

```
protected void notifyValidateSessionResult(PushPublishRTMPNetConnectionSession  
pushPublisherSession,  
boolean result)
```

**notifyPushPublisherSessionCreate**

```
protected void notifyPushPublisherSessionCreate(PushPublishRTMPNetConnectionSession  
pushPublisherSession)
```

Notify session create

**Parameters:**

pushPublisherSession - session holder

**notifyPushPublisherSessionDestroy**

```
protected void notifyPushPublisherSessionDestroy(PushPublishRTMPNetConnectionSession  
pushPublisherSession)
```

Notify session destroy

**Parameters:**

pushPublisherSession - session holder

**getConnectLastAttempt**

```
public long getConnectLastAttempt()
```

Get timestamp of last connection attempt

**Returns:**

timestamp of last connection attempt

(continued from last page)

## setConnectLastAttempt

```
public void setConnectLastAttempt(long connectLastAttempt)
```

Set timestamp of last connection attempt

**Parameters:**

connectLastAttempt - timestamp of last connection attempt

---

## getReconnectDelay

```
protected int getReconnectDelay(long currTime)
```

Get the reconnect delay

**Parameters:**

currTime - current timestamp

**Returns:**

delay in milliseconds

---

## getConnectLastSuccess

```
public long getConnectLastSuccess()
```

Get timestamp of last successful connection

**Returns:**

timestamp of last successful connection

---

## setConnectLastSuccess

```
public void setConnectLastSuccess(long connectLastSuccess)
```

Set timestamp of last successful connection

**Parameters:**

connectLastSuccess - timestamp of last successful connection

---

## isSendOriginalTimecodes

```
public boolean isSendOriginalTimecodes()
```

true if sending the original timecodes of source stream. false if starting at zero.

**Returns:**

true if sending the original timecodes of source stream

---

## setSendOriginalTimecodes

```
public void setSendOriginalTimecodes(boolean sendOriginalTimecodes)
```

true if sending the original timecodes of source stream. false if starting at zero.

**Parameters:**

sendOriginalTimecodes - true if sending the original timecodes of source stream

(continued from last page)

## getOriginalTimecodeThreshold

```
public long getOriginalTimecodeThreshold()
```

Get the original timecode threshold. This value will attempt to reduce the magnitude of absolute timecodes while keeping multiple-bitrate alignment. It is probably best to set this value quite high to a value such as 0x100000. The value is in milliseconds. The following formula is used to calculate a timecode offset  $\text{timecodeOffset} = \text{timecode} - (\text{timecode} \% \text{originalTimecodeThreshold})$ ;

**Returns:**

original timecode threshold

---

## setOriginalTimecodeThreshold

```
public void setOriginalTimecodeThreshold(long originalTimecodeThreshold)
```

Set the original timecode threshold. This value will attempt to reduce the magnitude of absolute timecodes while keeping multiple-bitrate alignment. It is probably best to set this value quite high to a value such as 0x100000. The value is in milliseconds. The following formula is used to calculate a timecode offset  $\text{timecodeOffset} = \text{timecode} - (\text{timecode} \% \text{originalTimecodeThreshold})$ ;

**Parameters:**

originalTimecodeThreshold - original timecode threshold

---

## isDebugLog

```
public boolean isDebugLog()
```

Is debug logging turned on

**Returns:**

true if debug logging is turned on

---

## setDebugLog

```
public void setDebugLog(boolean debugLog)
```

Set debug logging

**Parameters:**

debugLog - debug logging

---

## getAppInstance

```
public IApplicationInstance getAppInstance()
```

Get appInstance interface

**Returns:**

appInstance interface

---

## setAppInstance

```
public void setAppInstance(IApplicationInstance appInstance)
```

Set appInstance interface

**Parameters:**

appInstance - appInstance interface

## getSrcStreamName

```
public String getSrcStreamName()
```

Get source stream name

**Returns:**

source stream name

---

## setSrcStreamName

```
public void setSrcStreamName(String srcStreamName)
```

Set source stream name

**Parameters:**

srcStreamName - source stream name

---

## getDstStreamName

```
public String getDstStreamName()
```

Get destination stream name

**Returns:**

destination stream name

---

## setDstStreamName

```
public void setDstStreamName(String dstStreamName)
```

Set destination stream name

**Parameters:**

dstStreamName - destination stream name

---

## getHostname

```
public String getHostname()
```

Get destination server host name

**Returns:**

destination server host name

---

## setHostname

```
public void setHostname(String hostname)
```

Set destination server host name

**Parameters:**

hostname - destination server host name

---

## getPort

```
public int getPort()
```

---

(continued from last page)

Get destination server port

**Returns:**

destination server port

---

## setPort

```
public void setPort(int port)
```

Set destination server port

**Parameters:**

port - destination server port

---

## getDstApplicationName

```
public String getDstApplicationName()
```

Get destination application name

**Returns:**

destination application name

---

## setDstApplicationName

```
public void setDstApplicationName(String dstApplicationName)
```

Set destination application name

**Parameters:**

dstApplicationName - destination application name

---

## getDstAppInstanceName

```
public String getDstAppInstanceName()
```

Get destination appInstance name

**Returns:**

destination appInstance name

---

## setDstAppInstanceName

```
public void setDstAppInstanceName(String dstAppInstanceName)
```

Set destination appInstance name

**Parameters:**

dstAppInstanceName - destination appInstance name

---

## getRTMPAuthProvider

```
public IPushPublishRTMPAuthProvider getRTMPAuthProvider()
```

Get RTMP authentication provider

**Returns:**

RTMP authentication provider

---

## setRTMPAuthProvider

```
public void setRTMPAuthProvider(IPushPublishRTMPAuthProvider rtmpAuthProvider)
```

Set RTMP authentication provider

**Parameters:**

rtmpAuthProvider - RTMP authentication provider

---

## isSendOnMetadata

```
public boolean isSendOnMetadata()
```

True if sending onMetadata event at start of stream

**Returns:**

True if sending onMetadata event at start of stream

---

## setSendOnMetadata

```
public void setSendOnMetadata(boolean sendOnMetadata)
```

True if sending onMetadata event at start of stream

**Parameters:**

sendOnMetadata - True if sending onMetadata event at start of stream

---

## isRemoveDefaultAppInstance

```
public boolean isRemoveDefaultAppInstance()
```

If true and destination application instance is \_definst\_ then the application instance name will not be sent to the remove server as part of the connection information.

**Returns:**

true if omitting \_definst\_

---

## setRemoveDefaultAppInstance

```
public void setRemoveDefaultAppInstance(boolean removeDefaultAppInstance)
```

If true and destination application instance is \_definst\_ then the application instance name will not be sent to the remove server as part of the connection information.

**Parameters:**

removeDefaultAppInstance - true if omitting \_definst\_

---

## isOnMetadataToSetDataFrame

```
public boolean isOnMetadataToSetDataFrame()
```

If true all onMetadata events are converted to @setDataFrame calls

**Returns:**

true all onMetadata events are converted to @setDataFrame calls

---

---

(continued from last page)

## setOnMetadataToSetDataFrame

```
public void setOnMetadataToSetDataFrame(boolean onMetadataToSetDataFrame)
```

If true all onMetadata events are converted to @setDataFrame calls

**Parameters:**

onMetadataToSetDataFrame - true all onMetadata events are converted to @setDataFrame calls

---

## isSendStreamCloseCommands

```
public boolean isSendStreamCloseCommands()
```

If true will send FCUnpublish, closeStream, deleteStream on disconnect

**Returns:**

true will send close commands

---

## setSendStreamCloseCommands

```
public void setSendStreamCloseCommands(boolean sendStreamCloseCommands)
```

If true will send FCUnpublish, closeStream, deleteStream on disconnect

**Parameters:**

sendStreamCloseCommands - true will send close commands

---

## convertDataEvenToAMFData

```
public AMFDataList convertDataEvenToAMFData(AMFPacket packet)
```

---

## convertOnMetadataToSetDataFrame

```
public AMFPacket convertOnMetadataToSetDataFrame(AMFPacket packet)
```

---

## getStreamCloseWaitTime

```
public int getStreamCloseWaitTime()
```

Get stream close timeout (milliseconds)

**Returns:**

stream close timeout (milliseconds)

---

## setStreamCloseWaitTime

```
public void setStreamCloseWaitTime(int streamCloseWaitTime)
```

Set stream close timeout (milliseconds)

**Parameters:**

streamCloseWaitTime - stream close timeout (milliseconds)

---

(continued from last page)

## disconnect

```
public void disconnect()
```

Disconnect from server

---

## isFlashVerionFMLE

```
public static boolean isFlashVerionFMLE(String flashVersion)
```

Returns true if the Flash player version string is a Flash Media Live Encoder version (start with FMLE/ or FME/)

**Parameters:**

flashVersion - true if Flash Media Live Encoder version

**Returns:**

true if flash

---

## isFlashVersionFMLE

```
public static boolean isFlashVersionFMLE(String flashVersion)
```

Returns true if the Flash player version string is a Flash Media Live Encoder version (start with FMLE/ or FME/)

**Parameters:**

flashVersion - true if Flash Media Live Encoder version

**Returns:**

true if flash

---

## isDebugPackets

```
public boolean isDebugPackets()
```

Set to true to log every packet being sent

**Returns:**

true to log every packet being sent

---

## setDebugPackets

```
public void setDebugPackets(boolean debugPackets)
```

Set to true to log every packet being sent

**Parameters:**

debugPackets - true to log every packet being sent

---

## getTotalPacketsSent

```
public long getTotalPacketsSent()
```

Get total number of packets sent during push publishing session

**Returns:**

total number of packets sent during push publishing session

---

---

(continued from last page)

## isResetOnTimecodeOutOfOrder

```
public boolean isResetOnTimecodeOutOfOrder()
```

Set to true if you wish push to reset if timecodes jump out of order.

**Returns:**

true if you wish push to reset if timecodes jump out of order

---

## setResetOnTimecodeOutOfOrder

```
public void setResetOnTimecodeOutOfOrder(boolean resetOnTimecodeOutOfOrder)
```

Set to true if you wish push to reset if timecodes jump out of order.

**Parameters:**

resetOnTimecodeOutOfOrder - true if you wish push to reset if timecodes jump out of order

---

## getTimecodesOutOfOrderThreshold

```
public int getTimecodesOutOfOrderThreshold()
```

---

## setTimecodesOutOfOrderThreshold

```
public void setTimecodesOutOfOrderThreshold(int timecodesOutOfOrderThreshold)
```

---

## getLastVideoTC

```
public long getLastVideoTC()
```

Get the absolute timecode of the last sent video packet (milliseconds)

**Returns:**

absolute timecode of the last sent video packet (milliseconds)

---

## getLastAudioTC

```
public long getLastAudioTC()
```

Get the absolute timecode of the last sent audio packet (milliseconds)

**Returns:**

absolute timecode of the last sent audio packet (milliseconds)

---

## getLastDataTC

```
public long getLastDataTC()
```

Get the absolute timecode of the last sent data packet (milliseconds)

**Returns:**

absolute timecode of the last sent data packet (milliseconds)

---

(continued from last page)

## isWaitOnMetadataAvailable

```
public boolean isWaitOnMetadataAvailable()
```

Set to true to turn on system that will monitor onMetadata packet and wait for it to be populated before it starts sending the stream. Default is true. If stream is audio or video only, be sure to use setWaitOnMetadataVideo and setWaitOnMetadataAudio to properly turn off the channel you are not using or the system will always timeout.

**Returns:**

true to turn on system that will monitor onMetadata packet

---

## setWaitOnMetadataAvailable

```
public void setWaitOnMetadataAvailable(boolean waitOnMetadataAvailable)
```

Set to true to turn on system that will monitor onMetadata packet and wait for it to be populated before it starts sending the stream. Default is true. If stream is audio or video only, be sure to use setWaitOnMetadataVideo and setWaitOnMetadataAudio to properly turn off the channel you are not using or the system will always timeout.

**Parameters:**

waitOnMetadataAvailable - true to turn on system that will monitor onMetadata packet

---

## isWaitOnMetadataVideo

```
public boolean isWaitOnMetadataVideo()
```

If true and waitOnMetadataAvailable, publishing will not start until onMetadata packet is populated with video data.

**Returns:**

publishing will not start until onMetadata packet is populated with video data

---

## setWaitOnMetadataVideo

```
public void setWaitOnMetadataVideo(boolean waitOnMetadataVideo)
```

If true and waitOnMetadataAvailable, publishing will not start until onMetadata packet is populated with video data.

**Parameters:**

waitOnMetadataVideo - publishing will not start until onMetadata packet is populated with video data

---

## isWaitOnMetadataAudio

```
public boolean isWaitOnMetadataAudio()
```

If true and waitOnMetadataAvailable, publishing will not start until onMetadata packet is populated with audio data.

**Returns:**

publishing will not start until onMetadata packet is populated with audio data

---

## setWaitOnMetadataAudio

```
public void setWaitOnMetadataAudio(boolean waitOnMetadataAudio)
```

If true and waitOnMetadataAvailable, publishing will not start until onMetadata packet is populated with audio data.

**Parameters:**

waitOnMetadataAudio - publishing will not start until onMetadata packet is populated with audio data

---

(continued from last page)

## getWaitOnMetadataTimeout

```
public long getWaitOnMetadataTimeout()
```

Timeout (milliseconds) that the wait on onMetadata system will wait looking for video and audio data to be populated.

**Returns:**

timeout (milliseconds) that the wait on onMetadata system will wait looking for video and audio data to be populated

---

## setWaitOnMetadataTimeout

```
public void setWaitOnMetadataTimeout(long waitOnMetadataTimeout)
```

Timeout (milliseconds) that the wait on onMetadata system will wait looking for video and audio data to be populated.

**Parameters:**

waitOnMetadataTimeout - timeout (milliseconds) that the wait on onMetadata system will wait looking for video and audio data to be populated

---

## isWaitOnMetadataVideoDatarate

```
public boolean isWaitOnMetadataVideoDatarate()
```

---

## setWaitOnMetadataVideoDatarate

```
public void setWaitOnMetadataVideoDatarate(boolean waitOnMetadataVideoDatarate)
```

---

## isWaitOnMetadataAudioDatarate

```
public boolean isWaitOnMetadataAudioDatarate()
```

---

## setWaitOnMetadataAudioDatarate

```
public void setWaitOnMetadataAudioDatarate(boolean waitOnMetadataAudioDatarate)
```

---

## getPushPublishVersionStr

```
public static String getPushPublishVersionstr()
```

---

## getSrcStreamBufferSize

```
public long getSrcStreamBuffersize()
```

The amount of audio/video in milliseconds that must be available in the source streams buffer before push publishing will begin.

**Returns:**

amount of audio/video in milliseconds that must be available

## setSrcStreamBufferSize

```
public void setSrcStreamBufferSize(long srcStreamBufferSize)
```

The amount of audio/video in milliseconds that must be available in the source streams buffer before push publishing will begin.

**Parameters:**

srcStreamBufferSize - amount of audio/video in milliseconds that must be available

---

## addConnectMetaData

```
public void addConnectMetaData(String key,  
    AMFData value)
```

Added metadata values to the connect metadata object

**Parameters:**

key - key

value - AMFDataObj value

---

## createPushPublishSession

```
public IPushPublishSession createPushPublishSession()
```

---

## getPacketsSentAudio

```
public long getPacketsSentAudio()
```

---

## getPacketsSentVideo

```
public long getPacketsSentVideo()
```

---

## getPacketsSentData

```
public long getPacketsSentData()
```

---

## isSSL

```
public boolean isSSL()
```

---

## setSSL

```
public void setSSL(boolean isSSL)
```

---

## com.wowza.wms.pushpublish.protocol.rtmp Class PushPublishRTMPAkamaiHandler

```

java.lang.Object
+-com.wowza.wms.pushpublish.model.PushPublishBase
  +-com.wowza.wms.pushpublish.protocol.rtmp.PushPublishRTMP
    +-com.wowza.wms.pushpublish.protocol.rtmp.PushPublishRTMPPProfileHandler
      +-com.wowza.wms.pushpublish.protocol.rtmp.PushPublishRTMPAkamaiHandler

```

All Implemented Interfaces:

[IPushPublish](#)

public class **PushPublishRTMPAkamaiHandler**  
 extends [PushPublishRTMPPProfileHandler](#)

### Field Summary

public static final	<a href="#">CLASS</a>
public static final	<a href="#">CLASSNAME</a>  Value: <b>PushPublishRTMPAkamaiHandler</b>
protected	<a href="#">SendToBackupServer</a>  The is backup.

Fields inherited from class [com.wowza.wms.pushpublish.protocol.rtmp.PushPublishRTMPPProfileHandler](#)

[CLASS](#), [CLASSNAME](#), [streamId](#)

Fields inherited from class [com.wowza.wms.pushpublish.protocol.rtmp.PushPublishRTMP](#)

[akamaiOriginIp](#), [akamaiOriginPort](#), [akamaiPassword](#), [akamaiUserName](#), [bytesAvg](#), [CLASS](#), [CLASSNAME](#), [connectionFlashVersion](#), [connectionPageURL](#), [connectionQueryStr](#), [connectionSwfURL](#), [connectionTimeout](#), [connectLastAttempt](#), [connectLastSuccess](#), [connectMetaData](#), [CURRENTFLASHVERSION](#), [CURRENTFMLEVERSION](#), [debugLogLastPacketCount](#), [debugPackets](#), [doSendOnMetaData](#), [dstAppInstanceName](#), [dstApplicationName](#), [dstStreamName](#), [howToPublish](#), [ioSessionToSessionHolder](#), [isReconnectorRunning](#), [isSSL](#), [lastAudioTC](#), [lastDataTC](#), [lastIdleReadTime](#), [lastIdleWriteTime](#), [lastVideoTC](#), [listeners](#), [lock](#), [netconnectToSessionHolder](#), [offlineTime](#), [onlineTime](#), [onMetadataToSetDataFrame](#), [outprint](#), [packetsSentAudio](#), [packetsSentData](#), [packetsSentVideo](#), [removeDefaultAppInstance](#), [resetOnTimecodeOutOfOrder](#), [rtmpAuthProvider](#), [rtmpDebugLog](#), [secureTokenSharedSecret](#), [sendFCAnnounce](#), [sendFCPublish](#), [sendOnMetadata](#), [sendOriginalTimecodes](#), [sendReleaseStream](#), [sendStreamCloseCommands](#), [sendStreamCloseCommandsSendTime](#), [sessionOrder](#), [srcStreamBufferSize](#), [streamCloseWaitTime](#), [timecodesOutOfOrderThreshold](#), [totalPacketsSent](#), [tryConnect](#), [validationFrequency](#), [waitOnMetadataAudio](#), [waitOnMetadataAudioDatarate](#), [waitOnMetadataAvailable](#), [waitOnMetadataReady](#), [waitOnMetadataStartime](#), [waitOnMetadataTimeout](#), [waitOnMetadataVideo](#), [waitOnMetadataVideoDatarate](#)

Fields inherited from class [com.wowza.wms.pushpublish.model.PushPublishBase](#)

```
appInstance, CLASS, CLASSNAME, dstStreamName, hostname, mapFileIndex, mImplementation, mType,
originalTimecodeThreshold, password, port, properties, pushPublishStreamDebug,
pushPublishType, reconnectWaitTime, sendingPerfomance, sendOriginalTimecodes, srcStream,
srcStreamName, userName
```

Fields inherited from interface [com.wowza.wms.pushpublish.model.IPushPublish](#)

[PUSHPUBLISH\\_TYPE\\_RTMP](#), [PUSHPUBLISH\\_TYPE\\_RTP](#), [PUSHPUBLISH\\_TYPE\\_UNKNOWN](#)

## Constructor Summary

public	<a href="#">PushPublishRTMPAkamaiHandler()</a>
--------	--

## Method Summary

String	<a href="#">createAkamaiStreamName(String dstStreamName)</a>
String	<a href="#">getStreamAngle()</a>
String	<a href="#">getStreamBandWidth()</a>
void	<a href="#">init(IApplicationInstance appInstance2, String streamName, IMediaStream stream, java.util.HashMap profileData, java.util.HashMap maps, boolean debug)</a>
void	<a href="#">load(java.util.HashMap dataMap)</a>
void	<a href="#">setAkamaiOriginIp(String akamaiOriginIp)</a> Set Akamai Origin IP.
void	<a href="#">setAkamaiOriginPort(String akamaiOriginPort)</a>
void	<a href="#">setAkamaiPassword(String string)</a>
void	<a href="#">setAkamaiStreamId(String id)</a> Sets the akamai stream id.
void	<a href="#">setAkamaiUserName(String string)</a>
void	<a href="#">setPassword(String password)</a>
void	<a href="#">setStreamAngle(String streamAngleIn)</a>
void	<a href="#">setStreamBandWidth(String streamBandWidth)</a>
void	<a href="#">setStreamId(String id)</a>
void	<a href="#">setUserName(String userName)</a>

Methods inherited from class

[com.wowza.wms.pushpublish.protocol.rtmp.PushPublishRTMPPProfileHandler](#)

```

connect_Out, getDstAppInstanceName, getDstApplicationName, init, isDebugEnabled,
isDebugEnabledPackets, load, setApplication, setConnectionFlashVersion,
setConnectionInfoPageURL, setConnectionInfoQueryString, setConnectionInfoSwfURL,
setConnectionTimeout, setConnectLastAttempt, setConnectLastSuccess, setDebugPackets,
setDebugPackets, setDstAppInstanceName, setDstApplicationName, setDstStreamName,
setOnMetadataToSetDataFrame, setOriginalTimecodeThreshold, setReconnectWaitTime,
setRemoveDefaultAppInstance, setResetOnTimecodeOutOfOrder,
setSecureTokenSharedSecret, setSendFCAnnounce, setSendFCPublish, setSendOnMetadata,
setSendOriginalTimecodes, setSendReleaseStream, setSendStreamCloseCommands, setSSL,
setStreamCloseWaitTime, setTimecodesOutOfOrderThreshold, setValidationFrequency,
setWaitOnMetadataAudio, setWaitOnMetadataAudioDatarate, setWaitOnMetadataAvailable,
setWaitOnMetadataVideo, setWaitOnMetadataVideoDatarate

```

#### Methods inherited from class [com.wowza.wms.pushpublish.protocol.rtmp.PushPublishRTMP](#)

```

addConnectMetaData, addListener, addSession, callConnect, connect,
convertDataEvenToAMFData, convertOnMetadataToSetDataFrame, createPushPublishSession,
disconnect, getAkamaiOriginIp, getAkamaiOriginPort, getAkamaiPassword,
getAkamaiSessionKey, getAkamaiUserName, getAppInstance, getConnectionFlashVerion,
getConnectionFlashVersion, getConnectionPageURL, getConnectionQueryStr,
getConnectionSwfURL, getConnectionTimeout, getConnectLastAttempt,
getConnectLastSuccess, getContextStr, getDstAppInstanceName, getDstApplicationName,
getDstStreamName, getHostname, getHowToPublish, getLastAudioTC, getLastDataTC,
getLastSession, getLastVideoTC, getLocalListeners, getLock, getNetConnection,
getOriginalTimecodeThreshold, getPacketsSentAudio, getPacketsSentData,
getPacketsSentVideo, getPort, getPushPublishVersionStr, getReconnectDelay,
getRTMPAuthProvider, getSecureTokenSharedSecret, getSessionHolderByNetConnection,
getSessionHolderBySession, getSessionListDebug, getSrcStreamBufferSize,
getSrcStreamName, getStreamCloseWaitTime, getTimecodesOutOfOrderThreshold,
getTotalPacketsSent, getValidationFrequency, getWaitOnMetadataTimeout,
internalConnect, isCurrentSession, isDebugEnabled, isDebugEnabledPackets, isFlashVerionFMLE,
isFlashVersionFMLE, isOnMetadataToSetDataFrame, isRemoveDefaultAppInstance,
isResetOnTimecodeOutOfOrder, isSendFCAnnounce, isSendFCPublish, isSendOnMetadata,
isSendOriginalTimecodes, isSendReleaseStream, isSendStreamCloseCommands, isSSL,
isWaitOnMetadataAudio, isWaitOnMetadataAudioDatarate, isWaitOnMetadataAvailable,
isWaitOnMetadataVideo, isWaitOnMetadataVideoDatarate, notifyAkamaiClientLogin,
notifyAkamaiSetChallenge, notifyAkamaiSetOriginConnectionInfo, notifyConnect,
notifyConnectFailure, notifyConnectStart, notifyConnectSuccess, notifyFCAnnounce,
notifyFCPublish, notifyHandshakeResult, notifyPublishHandlerPlay,
notifyPushPublisherSessionCreate, notifyPushPublisherSessionDestroy,
notifySessionClosed, notifySessionIdle, notifySessionOpened, notifyStreamCreate,
notifyStreamOnPlayStatus, notifyStreamOnStatus, notifyValidateSession,
notifyValidateSessionResult, removeListener, removeSession, resetConnection,
resetConnection, sessionClosed, sessionOpened, setAkamaiOriginIp,
setAkamaiOriginPort, setAkamaiPassword, setAkamaiUserName, setAppInstance,
setConnectionFlashVerion, setConnectionFlashVersion, setConnectionPageURL,
setConnectionQueryStr, setConnectionSwfURL, setConnectionTimeout,
setConnectLastAttempt, setConnectLastSuccess, setDebugLog, setDebugPackets,
setDstAppInstanceName, setDstApplicationName, setDstStreamName, setHostname,
setHowToPublish, setOnMetadataToSetDataFrame, setOriginalTimecodeThreshold, setPort,
setRemoveDefaultAppInstance, setResetOnTimecodeOutOfOrder, setRTMPAuthProvider,
setSecureTokenSharedSecret, setSendFCAnnounce, setSendFCPublish, setSendOnMetadata,
setSendOriginalTimecodes, setSendReleaseStream, setSendStreamCloseCommands,
setSrcStreamBufferSize, setSrcStreamName, setSSL, setStreamCloseWaitTime,
setTimecodesOutOfOrderThreshold, setValidationFrequency, setWaitOnMetadataAudio,
setWaitOnMetadataAudioDatarate, setWaitOnMetadataAvailable, setWaitOnMetadataTimeout,
setWaitOnMetadataVideo, setWaitOnMetadataVideoDatarate, tryToConnect

```

**Methods inherited from class** [com.wowza.wms.pushpublish.model.PushPublishBase](#)

```
createPushPublishSession, getAppInstance, getBandwidth, getConfigDataOrDefault,
getConfigDataOrDefault, getContextStr, getDstStreamName,
getHostname, getImplementation, getMapFileIndex, getNumberFromString,
getOriginalTimecodeThreshold, getPassword, getPort, getProperties,
getPushPublishType, getPushPublishVersionStr, getReconnectWaitTime, getSrcStream,
getSrcStreamName, getUserName, init, isAdaptiveStreaming, isSendOriginalTimecodes,
load, setAdaptiveStreaming, setAppInstance, setDebugLog, setDstStreamName, setHost,
setHostname, setImplementation, setMapFileIndex, setOriginalTimecodeThreshold,
setOriginalTimecodeThreshold, setPassword, setPort, setPort, setReconnectWaitTime,
setSendOriginalTimecodes, setSrcStream, setSrcStreamName, setStreamName, setUserName
```

**Methods inherited from class** [java.lang.Object](#)

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait
```

**Methods inherited from interface** [com.wowza.wms.pushpublish.model.IPushPublish](#)

```
connect, createPushPublishSession, disconnect, getContextStr, getDstStreamName,
getHostname, getMapFileIndex, getPort, getProperties, getPushPublishType,
getReconnectWaitTime, getSrcStream, getSrcStreamName, init, resetConnection,
setDebugLog, setDstStreamName, setHost, setHostname, setMapFileIndex, setPort,
setPort, setReconnectWaitTime, setSrcStream, setSrcStreamName, setStreamName
```

## Fields

### CLASS

```
public static final java.lang.Class CLASS
```

### CLASSNAME

```
public static final java.lang.String CLASSNAME
```

Constant value: **PushPublishRTMPAkamaiHandler**

### SendToBackupServer

```
protected boolean SendToBackupServer
```

The is backup.

## Constructors

### PushPublishRTMPAkamaiHandler

```
public PushPublishRTMPAkamaiHandler()
```

(continued from last page)

## Methods

### init

```
public void init(IApplicationInstance appInstance2,  
String streamName,  
IMediaStream stream,  
java.util.HashMap profileData,  
java.util.HashMap maps,  
boolean debug)
```

Inits the instance.

### load

```
public void load(java.util.HashMap dataMap)
```

### setUserName

```
public void setUserName(String userName)
```

Set Akamai user name.

### setPassword

```
public void setPassword(String password)
```

Set Akamai password.

### setAkamaiOriginIp

```
public void setAkamaiOriginIp(String akamaiOriginIp)
```

Set Akamai Origin IP.

**Parameters:**

akamaiOriginIp - Akamai Origin IP

### setAkamaiOriginPort

```
public void setAkamaiOriginPort(String akamaiOriginPort)
```

### setAkamaiStreamId

```
public void setAkamaiStreamId(String id)
```

Sets the akamai stream id.

**Parameters:**

id - the new akamai stream id

### setStreamId

```
public void setStreamId(String id)
```

(continued from last page)

---

**setAkamaiUserName**

```
public void setAkamaiUserName(String string)
```

Set Akamai user name

---

**setAkamaiPassword**

```
public void setAkamaiPassword(String string)
```

Set Akamai password

---

**getStreamAngle**

```
public String getStreamAngle()
```

---

**setStreamAngle**

```
public void setStreamAngle(String streamAngleIn)
```

---

**getStreamBandWidth**

```
public String getStreamBandWidth()
```

---

**setStreamBandWidth**

```
public void setStreamBandWidth(String streamBandWidth)
```

---

**createAkamaiStreamName**

```
public String createAkamaiStreamName(String dstStreamName)
```

## com.wowza.wms.pushpublish.protocol.rtmp Class PushPublishRTMPAuthProviderAdobe

```
java.lang.Object
+- com.wowza.wms.pushpublish.protocol.rtmp.PushPublishRTMPAuthProviderBase
  +- com.wowza.wms.pushpublish.protocol.rtmp.PushPublishRTMPAuthProviderAdobe
```

### All Implemented Interfaces:

[IPushPublishRTMPAuthProvider](#)

```
public class PushPublishRTMPAuthProviderAdobe
extends PushPublishRTMPAuthProviderBase
```

### Field Summary

protected	<a href="#">authMode</a>
protected	<a href="#">challenge</a>
public static final	<a href="#">CLASS</a>
public static final	<a href="#">CLASSNAME</a> Value: <b>PushPublishRTMPAuthProviderAdobe</b>
protected	<a href="#">md5Digest</a>
protected	<a href="#">opaque</a>
protected	<a href="#">salt</a>

#### Fields inherited from class

[com.wowza.wms.pushpublish.protocol.rtmp.PushPublishRTMPAuthProviderBase](#)

[lock](#), [PARAM1](#), [password](#), [publisher](#), [userName](#)

### Constructor Summary

public	<a href="#">PushPublishRTMPAuthProviderAdobe()</a>
--------	--

### Method Summary

String	<a href="#">addAuthQueryString</a> (String queryString)
void	<a href="#">handleConnectResponse</a> (com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params)
void	<a href="#">init</a> ( <a href="#">PushPublishRTMP</a> publisher)

**Methods inherited from class**[com.wowza.wms.pushpublish.protocol.rtmp.PushPublishRTMPAuthProviderBase](#)[getParamObj](#), [getPassword](#), [getPublisher](#), [getUserName](#), [init](#), [setPassword](#), [setUserName](#), [splitQueryStr](#)**Methods inherited from class** java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait

**Methods inherited from interface**[com.wowza.wms.pushpublish.protocol.rtmp.IPushPublishRTMPAuthProvider](#)[addAuthQueryString](#), [getPassword](#), [getPublisher](#), [getUserName](#), [handleConnectResponse](#), [init](#), [setPassword](#), [setUserName](#)

## Fields

### CLASS

public static final java.lang.Class CLASS

### CLASSNAME

public static final java.lang.String CLASSNAME

Constant value: PushPublishRTMPAuthProviderAdobe

### authMode

protected java.lang.String authMode

### salt

protected java.lang.String salt

### challenge

protected java.lang.String challenge

### opaque

protected java.lang.String opaque

## **md5Digest**

```
protected java.security.MessageDigest md5Digest
```

## Constructors

### **PushPublishRTMPAuthProviderAdobe**

```
public PushPublishRTMPAuthProviderAdobe()
```

## Methods

### **init**

```
public void init(PushPublishRTMP publisher)
```

---

### **addAuthQueryString**

```
public String addAuthQueryString(String queryString)
```

---

### **handleConnectResponse**

```
public void handleConnectResponse(com.wowza.wms.request.RequestFunction function,  
                                 AMFDataList params)
```

## com.wowza.wms.pushpublish.protocol.rtmp Class PushPublishRTMPAuthProviderBase

```
java.lang.Object
+-com.wowza.wms.pushpublish.protocol.rtmp.PushPublishRTMPAuthProviderBase
```

### All Implemented Interfaces:

[IPushPublishRTMPAuthProvider](#)

### Direct Known Subclasses:

[PushPublishRTMPAuthProviderLimelight](#), [PushPublishRTMPAuthProviderAdobe](#)

public abstract class **PushPublishRTMPAuthProviderBase**  
 extends Object  
 implements [IPushPublishRTMPAuthProvider](#)

## Field Summary

protected	<a href="#">lock</a>
public static final	<a href="#">PARAM1</a>  Value: 3
protected	<a href="#">password</a>
protected	<a href="#">publisher</a>
protected	<a href="#">userName</a>

## Constructor Summary

public	<a href="#">PushPublishRTMPAuthProviderBase()</a>
--------	---

## Method Summary

<a href="#">AMFDataObj</a>	<a href="#">getParamObj(<a href="#">AMFDataList</a> params, int index)</a>
String	<a href="#">getPassword()</a>
<a href="#">PushPublishRTMP</a>	<a href="#">getPublisher()</a>
String	<a href="#">getUserName()</a>
void	<a href="#">init(<a href="#">PushPublishRTMP</a> publisher)</a>
void	<a href="#">setPassword(String password)</a>

void	<a href="#"><u>setUserName</u></a> (String userName)
java.util.Map	<a href="#"><u>splitQueryStr</u></a> (String queryStr)

**Methods inherited from class** java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait

**Methods inherited from interface**

[com.wowza.wms.pushpublish.protocol.rtmp.IPushPublishRTMPAuthProvider](#)

[addAuthQueryString](#), [getPassword](#), [getPublisher](#), [getUserName](#), [handleConnectResponse](#),  
[init](#), [setPassword](#), [setUserName](#)

## Fields

**PARAM1**

```
public static final int PARAM1
```

Constant value: 3

**publisher**

```
protected com.wowza.wms.pushpublish.protocol.rtmp.PushPublishRTMP publisher
```

**userName**

```
protected java.lang.String userName
```

**password**

```
protected java.lang.String password
```

**lock**

```
protected java.lang.Object lock
```

## Constructors

**PushPublishRTMPAuthProviderBase**

```
public PushPublishRTMPAuthProviderBase()
```

## Methods

### init

```
public void init(PushPublishRTMP publisher)
```

---

### getParamObj

```
protected AMFDataObj getParamObj(AMFDataList params,  
int index)
```

---

### getUserName

```
public String getUserName()
```

---

### setUserName

```
public void setUserName(String userName)
```

---

### getPassword

```
public String getPassword()
```

---

### setPassword

```
public void setPassword(String password)
```

---

### getPublisher

```
public PushPublishRTMP getPublisher()
```

---

### splitQueryStr

```
public java.util.Map splitQueryStr(String queryStr)
```

---

## com.wowza.wms.pushpublish.protocol.rtmp Class PushPublishRTMPAuthProviderLimelight

```
java.lang.Object
+- com.wowza.wms.pushpublish.protocol.rtmp.PushPublishRTMPAuthProviderBase
  +- com.wowza.wms.pushpublish.protocol.rtmp.PushPublishRTMPAuthProviderLimelight
```

### All Implemented Interfaces:

[IPushPublishRTMPAuthProvider](#)

```
public class PushPublishRTMPAuthProviderLimelight
extends PushPublishRTMPAuthProviderBase
```

## Field Summary

protected	<a href="#">authMode</a>
public static final	<a href="#">CLASS</a>
public static final	<a href="#">CLASSNAME</a> Value: <b>PushPublishRTMPAuthProviderLimelight</b>
protected	<a href="#">conce</a>
protected	<a href="#">methodValue</a>
protected	<a href="#">ncCounter</a>
protected	<a href="#">nonce</a>
protected	<a href="#">qopValue</a>
protected	<a href="#">realm</a>

### Fields inherited from class

[com.wowza.wms.pushpublish.protocol.rtmp.PushPublishRTMPAuthProviderBase](#)

[lock](#), [PARAM1](#), [password](#), [publisher](#), [userName](#)

## Constructor Summary

public	<a href="#">PushPublishRTMPAuthProviderLimelight()</a>
--------	--

## Method Summary

String	<a href="#">addAuthQueryString</a> (String queryString)
--------	---

String	<a href="#">getAuthMode()</a>
String	<a href="#">getConce()</a>
String	<a href="#">getMethodValue()</a>
int	<a href="#">getNCCounter()</a>
String	<a href="#">getNonce()</a>
<a href="#">PushPublishRTMP</a>	<a href="#">getPublisher()</a>
String	<a href="#">getQOPValue()</a>
String	<a href="#">getRealm()</a>
void	<a href="#">handleConnectResponse(com.wowza.wms.request.RequestFunction function, AMFDataList params)</a>
void	<a href="#">init(<a href="#">PushPublishRTMP</a> publisher)</a>
void	<a href="#">initConce()</a>
void	<a href="#">setAuthMode(String authMode)</a>
void	<a href="#">setConce(String conce)</a>
void	<a href="#">setMethodValue(String methodValue)</a>
void	<a href="#">setNCCounter(int ncCounter)</a>
void	<a href="#">setNonce(String nonce)</a>
void	<a href="#">setQOPValue(String qopValue)</a>
void	<a href="#">setRealm(String realm)</a>

**Methods inherited from class**[com.wowza.wms.pushpublish.protocol.rtmp.PushPublishRTMPAuthProviderBase](#)

```
getParamObj, getPassword, getPublisher, getUserName, init, setPassword, setUserName, splitQueryStr
```

**Methods inherited from class** `java.lang.Object`

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait
```

**Methods inherited from interface**[com.wowza.wms.pushpublish.protocol.rtmp.IPushPublishRTMPAuthProvider](#)

```
addAuthQueryString, getPassword, getPublisher, getUserName, handleConnectResponse,  
init, setPassword, setUserName
```

## Fields

### CLASS

```
public static final java.lang.Class CLASS
```

### CLASSNAME

```
public static final java.lang.String CLASSNAME
```

Constant value: **PushPublishRTMPAuthProviderLimelight**

### realm

```
protected java.lang.String realm
```

### methodValue

```
protected java.lang.String methodValue
```

### qopValue

```
protected java.lang.String qopValue
```

### authMode

```
protected java.lang.String authMode
```

### nonce

```
protected java.lang.String nonce
```

### conce

```
protected java.lang.String conce
```

(continued from last page)

**ncCounter**

```
protected int ncCounter
```

## Constructors

**PushPublishRTMPAuthProviderLimelight**

```
public PushPublishRTMPAuthProviderLimelight()
```

## Methods

**init**

```
public void init(PushPublishRTMP publisher)
```

**initCOnce**

```
protected void initCOnce()
```

**addAuthQueryString**

```
public String addAuthQueryString(String queryString)
```

**handleConnectResponse**

```
public void handleConnectResponse(com.wowza.wms.request.RequestFunction function,  
                                 AMFDataList params)
```

**getPublisher**

```
public PushPublishRTMP getPublisher()
```

**getRealm**

```
public String getRealm()
```

**setRealm**

```
public void setRealm(String realm)
```

---

### **getMethodValue**

```
public String getMethodValue( )
```

---

### **setMethodValue**

```
public void setMethodValue(String methodValue)
```

---

### **getQOPValue**

```
public String getQOPValue( )
```

---

### **setQOPValue**

```
public void setQOPValue(String qopValue)
```

---

### **getAuthMode**

```
public String getAuthMode( )
```

---

### **setAuthMode**

```
public void setAuthMode(String authMode)
```

---

### **getNonce**

```
public String getNonce( )
```

---

### **setNonce**

```
public void setNonce(String nonce)
```

---

### **getConce**

```
public String getConce( )
```

---

### **setConce**

```
public void setConce(String conce)
```

---

(continued from last page)

---

### **getNCCounter**

```
public int getNCCounter()
```

---

### **setNCCounter**

```
public void setNCCounter(int ncCounter)
```

## com.wowza.wms.pushpublish.protocol.rtmp Class PushPublishRTMPNetConnectionSession

```
java.lang.Object
+-com.wowza.wms.pushpublish.protocol.rtmp.PushPublishRTMPNetConnectionSession
```

```
public class PushPublishRTMPNetConnectionSession
extends Object
```

### Field Summary

protected	<a href="#">netConnection</a>
protected	<a href="#">publishStream</a>
protected	<a href="#">session</a>
protected	<a href="#">streamsToPublish</a>

### Constructor Summary

public	<a href="#">PushPublishRTMPNetConnectionSession</a> (com.wowza.wms.netconnection.INetConnection netConnection, org.apache.mina.common.IoSession session)
--------	--

### Method Summary

com.wowza.wms.netconnection.INetConnection	<a href="#">getNetConnection()</a>
<a href="#">IMediaStream</a>	<a href="#">getPublishStream()</a>
org.apache.mina.common.IoSession	<a href="#">getSession()</a>
java.util.List	<a href="#">getStreamsToPublish()</a>
void	<a href="#">setNetConnection</a> (com.wowza.wms.netconnection.INetConnection netConnection)
void	<a href="#">setPublishStream</a> ( <a href="#">IMediaStream</a> publishStream)
void	<a href="#">setSession</a> (org.apache.mina.common.IoSession session)
void	<a href="#">setStreamsToPublish</a> (java.util.List streamsToPublish)

Methods inherited from class `java.lang.Object`

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait,  
wait
```

## Fields

### session

```
protected org.apache.mina.common.IoSession session
```

### netConnection

```
protected com.wowza.wms.netconnection.INetConnection netConnection
```

### streamsToPublish

```
protected java.util.List streamsToPublish
```

### publishStream

```
protected com.wowza.wms.stream.IMediaStream publishStream
```

## Constructors

### PushPublishRTMPNetConnectionSession

```
public PushPublishRTMPNetConnectionSession(com.wowza.wms.netconnection.INetConnection  
netConnection,  
                                         org.apache.mina.common.IoSession session)
```

## Methods

### getSession

```
public org.apache.mina.common.IoSession getSession()
```

### setSession

```
public void setSession(org.apache.mina.common.IoSession session)
```

(continued from last page)

**getNetConnection**

```
public com.wowza.wms.netconnection.INetConnection getNetConnection()
```

---

**setNetConnection**

```
public void setNetConnection(com.wowza.wms.netconnection.INetConnection netConnection)
```

---

**getStreamsToPublish**

```
public java.util.List getStreamsToPublish()
```

---

**setStreamsToPublish**

```
public void setStreamsToPublish(java.util.List streamsToPublish)
```

---

**getPublishStream**

```
public IMediaStream getPublishStream()
```

---

**setPublishStream**

```
public void setPublishStream(IMediaStream publishStream)
```

## com.wowza.wms.pushpublish.protocol.rtmp Class PushPublishRTMPNotifyBase

```
java.lang.Object
+-com.wowza.wms.pushpublish.protocol.rtmp.PushPublishRTMPNotifyBase
```

### All Implemented Interfaces:

[IPushPublishRTMPNotify](#)

```
public abstract class PushPublishRTMPNotifyBase
extends Object
implements IPushPublishRTMPNotify
```

Abstract base class for [IPushPublishRTMPNotify](#) interface

### See Also:

[PushPublishRTMP.addListener\(IPushPublishRTMPNotify\)](#)

## Constructor Summary

public	<a href="#">PushPublishRTMPNotifyBase()</a>
--------	---

## Method Summary

void	<a href="#">onAkamaiClientLogin(PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)</a>
void	<a href="#">onAkamaiSetChallenge(PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)</a>
void	<a href="#">onAkamaiSetOriginConnectionInfo(PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)</a>
void	<a href="#">onConnect(PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)</a>
void	<a href="#">onConnectFailure(PushPublishRTMPNetConnectionSession pushPublisherSession)</a>
void	<a href="#">onConnectStart(PushPublishRTMPNetConnectionSession pushPublisherSession)</a>
void	<a href="#">onConnectSuccess(PushPublishRTMPNetConnectionSession pushPublisherSession)</a>
void	<a href="#">onFCAnnounce(PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)</a>
void	<a href="#">onFCPublish(PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)</a>

void	<a href="#">onHandshakeResult(PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)</a>
void	<a href="#">onPublishHandlerPlay(PushPublishRTMPNetConnectionSession pushPublisherSession, java.io.OutputStream out, long[] playSizes)</a>
void	<a href="#">onPushPublisherSessionCreate(PushPublishRTMPNetConnectionSession pushPublisherSession)</a>
void	<a href="#">onPushPublisherSessionDestroy(PushPublishRTMPNetConnectionSession pushPublisherSession)</a>
void	<a href="#">onReleaseStream(PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)</a>
void	<a href="#">onSessionClosed(PushPublishRTMPNetConnectionSession pushPublisherSession)</a>
void	<a href="#">onSessionIdle(PushPublishRTMPNetConnectionSession pushPublisherSession)</a>
void	<a href="#">onSessionOpened(PushPublishRTMPNetConnectionSession pushPublisherSession)</a>
void	<a href="#">onStreamCreate(PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)</a>
void	<a href="#">onStreamOnPlayStatus(PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)</a>
void	<a href="#">onStreamOnStatus(PushPublishRTMPNetConnectionSession pushPublisherSession, com.wowza.wms.request.RequestFunction function, AMFDataList params)</a>
void	<a href="#">onValidateSession(PushPublishRTMPNetConnectionSession pushPublisherSession)</a>
void	<a href="#">onValidateSessionResult(PushPublishRTMPNetConnectionSession pushPublisherSession, boolean result)</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.pushpublish.protocol.rtmp.IPushPublishRTMPNotify**

[onAkamaiClientLogin](#), [onAkamaiSetChallenge](#), [onAkamaiSetOriginConnectionInfo](#), [onConnect](#), [onConnectFailure](#), [onConnectStart](#), [onConnectSuccess](#), [onFCAnnounce](#), [onFCPublish](#), [onHandshakeResult](#), [onPublishHandlerPlay](#), [onPushPublisherSessionCreate](#), [onPushPublisherSessionDestroy](#), [onReleaseStream](#), [onSessionClosed](#), [onSessionIdle](#), [onSessionOpened](#), [onStreamCreate](#), [onStreamOnPlayStatus](#), [onStreamOnStatus](#), [onValidateSession](#), [onValidateSessionResult](#)

## Constructors

### PushPublishRTMPNotifyBase

```
public PushPublishRTMPNotifyBase()
```

## Methods

### onConnectStart

```
public void onConnectStart(PushPublishRTMPNetConnectionSession pushPublisherSession)
```

---

### onConnectSuccess

```
public void onConnectSuccess(PushPublishRTMPNetConnectionSession pushPublisherSession)
```

---

### onConnectFailure

```
public void onConnectFailure(PushPublishRTMPNetConnectionSession pushPublisherSession)
```

---

### onHandshakeResult

```
public void onHandshakeResult(PushPublishRTMPNetConnectionSession  
pushPublisherSession,  
com.wowza.wms.request.RequestFunction function,  
AMFDataList params)
```

---

### onAkamaiSetChallenge

```
public void onAkamaiSetChallenge(PushPublishRTMPNetConnectionSession  
pushPublisherSession,  
com.wowza.wms.request.RequestFunction function,  
AMFDataList params)
```

---

### onAkamaiSetOriginConnectionInfo

```
public void onAkamaiSetOriginConnectionInfo(PushPublishRTMPNetConnectionSession  
pushPublisherSession,  
com.wowza.wms.request.RequestFunction function,  
AMFDataList params)
```

(continued from last page)

## onAkamaiClientLogin

```
public void onAkamaiClientLogin(PushPublishRTMPNetConnectionSession
pushPublisherSession,
    com.wowza.wms.request.RequestFunction function,
AMFDataList params)
```

## onFCPublish

```
public void onFCPublish(PushPublishRTMPNetConnectionSession pushPublisherSession,
    com.wowza.wms.request.RequestFunction function,
AMFDataList params)
```

## onFCAnnounce

```
public void onFCAnnounce(PushPublishRTMPNetConnectionSession pushPublisherSession,
    com.wowza.wms.request.RequestFunction function,
AMFDataList params)
```

## onReleaseStream

```
public void onReleaseStream(PushPublishRTMPNetConnectionSession pushPublisherSession,
    com.wowza.wms.request.RequestFunction function,
AMFDataList params)
```

## onConnect

```
public void onConnect(PushPublishRTMPNetConnectionSession pushPublisherSession,
    com.wowza.wms.request.RequestFunction function,
AMFDataList params)
```

## onStreamCreate

```
public void onStreamCreate(PushPublishRTMPNetConnectionSession pushPublisherSession,
    com.wowza.wms.request.RequestFunction function,
AMFDataList params)
```

## onStreamOnPlayStatus

```
public void onStreamOnPlayStatus(PushPublishRTMPNetConnectionSession
pushPublisherSession,
    com.wowza.wms.request.RequestFunction function,
AMFDataList params)
```

(continued from last page)

### onStreamOnStatus

```
public void onStreamOnStatus(PushPublishRTMPNetConnectionSession pushPublisherSession,  
    com.wowza.wms.request.RequestFunction function,  
    AMFDataList params)
```

---

### onPublishHandlerPlay

```
public void onPublishHandlerPlay(PushPublishRTMPNetConnectionSession  
pushPublisherSession,  
    java.io.OutputStream out,  
    long[] playSizes)
```

---

### onSessionIdle

```
public void onSessionIdle(PushPublishRTMPNetConnectionSession pushPublisherSession)
```

---

### onSessionOpened

```
public void onSessionOpened(PushPublishRTMPNetConnectionSession pushPublisherSession)
```

---

### onSessionClosed

```
public void onSessionClosed(PushPublishRTMPNetConnectionSession pushPublisherSession)
```

---

### onValidateSession

```
public void onValidateSession(PushPublishRTMPNetConnectionSession  
pushPublisherSession)
```

---

### onValidateSessionResult

```
public void onValidateSessionResult(PushPublishRTMPNetConnectionSession  
pushPublisherSession,  
    boolean result)
```

---

### onPushPublisherSessionCreate

```
public void onPushPublisherSessionCreate(PushPublishRTMPNetConnectionSession  
pushPublisherSession)
```

---

(continued from last page)

## onPushPublisherSessionDestroy

```
public void onPushPublisherSessionDestroy(PushPublishRTMPNetConnectionSession  
pushPublisherSession)
```

## com.wowza.wms.pushpublish.protocol.rtmp Class PushPublishRTMPPProfileHandler

```
java.lang.Object
+-com.wowza.wms.pushpublish.model.PushPublishBase
  +-com.wowza.wms.pushpublish.protocol.rtmp.PushPublishRTMP
    +-com.wowza.wms.pushpublish.protocol.rtmp.PushPublishRTMPPProfileHandler
```

All Implemented Interfaces:

[IPushPublish](#)

Direct Known Subclasses:

[PushPublishRTMPUstream](#), [PushPublishRTMPAkamaiHandler](#)

```
public class PushPublishRTMPPProfileHandler
extends PushPublishRTMP
```

### Field Summary

public static final	<a href="#">CLASS</a>
public static final	<a href="#">CLASSNAME</a> Value: <b>PushPublishRTMPPProfileHandler</b>
protected	<a href="#">streamId</a> The stream id.

### Fields inherited from class com.wowza.wms.pushpublish.protocol.rtmp.PushPublishRTMP

```
akamaiOriginIp, akamaiOriginPort, akamaiPassword, akamaiUserName, bytesAvg, CLASS, CLASSNAME,
connectionFlashVersion, connectionPageURL, connectionQueryStr, connectionSwfURL,
connectionTimeout, connectLastAttempt, connectLastSuccess, connectMetaData,
CURRENTFLASHVERSION, CURRENTFMLEVERSION, debugLogLastPacketCount, debugPackets,
doSendOnMetaData, dstAppInstanceName, dstApplicationName, dstStreamName, howToPublish,
ioSessionToSessionHolder, isReconnectorRunning, isSSL, lastAudioTC, lastDataTC,
lastIdleReadTime, lastIdleWriteTime, lastVideoTC, listeners, lock,
netConnectToSessionHolder, offlineTime, onlineTime, onMetadataToSetDataFrame, outprint,
packetsSentAudio, packetsToSendData, packetsSentVideo, removeDefaultAppInstance,
resetOnTimecodeOutOfOrder, rtmpAuthProvider, rtmpDebugLog, secureTokenSharedSecret,
sendFCAnnounce, sendFCPublish, sendOnMetadata, sendOriginalTimecodes, sendReleaseStream,
sendStreamCloseCommands, sendStreamCloseCommandsSendTime, sessionOrder,
srcStreamBufferSize, streamCloseWaitTime, timecodesOutOfOrderThreshold, totalPacketsSent,
tryConnect, validationFrequency, waitOnMetadataAudio, waitOnMetadataAudioDatarate,
waitOnMetadataAvailable, waitOnMetadataReady, waitOnMetadataStartTime,
waitOnMetadataTimeout, waitOnMetadataVideo, waitOnMetadataVideoDatarate
```

### Fields inherited from class com.wowza.wms.pushpublish.model.PushPublishBase

```
appInstance, CLASS, CLASSNAME, dstStreamName, hostname, mapFileIndex, mImplementation, mType,
originalTimecodeThreshold, password, port, properties, pushPublishStreamDebug,
pushPublishType, reconnectWaitTime, sendingPerfomance, sendOriginalTimecodes, srcStream,
srcStreamName, userName
```

Fields inherited from interface [com.wowza.wms.pushpublish.model.IPushPublish](#)

[PUSHPUBLISH\\_TYPE\\_RTMP](#), [PUSHPUBLISH\\_TYPE\\_RTP](#), [PUSHPUBLISH\\_TYPE\\_UNKNOWN](#)

## Constructor Summary

public	<a href="#">PushPublishRTMPProfileHandler()</a>
	Instantiates a new push publisher rtmp.

## Method Summary

void	<a href="#">connect_Out()</a> Connect to server.
String	<a href="#">getDstAppInstanceName()</a> Get destination appInstance name.
String	<a href="#">getDstApplicationName()</a> Get destination application name.
void	<a href="#">init(IApplicationInstance appInstance2, String streamName, IMediaStream stream, java.util.HashMap profileData, java.util.HashMap maps, boolean debug)</a>
boolean	<a href="#">isDebugEnabled()</a> Is debug logging turned on.
boolean	<a href="#">isDebugEnabled()</a> Checks if is debug packets.
void	<a href="#">load(java.util.HashMap dataMap)</a>
void	<a href="#">setApplication(String dstApplicationNameIN)</a> Set destination application name.
void	<a href="#">setConnectionFlashVersion(String connectionFlashVersion)</a> Set Flash version used in connection metadata.
void	<a href="#">setConnectionInfoPageURL(String connectionPageURL)</a> Set Page URL used in connection metadata.
void	<a href="#">setConnectionInfoQueryString(String connectionQueryStr)</a> Set the connection query string.
void	<a href="#">setConnectionInfoSwfURL(String connectionSwfURL)</a> Set SWF URL used in connection metadata.
void	<a href="#">setConnectionTimeout(String connectionTimeout)</a>
void	<a href="#">setConnectLastAttempt(String connectLastAttempt)</a>

void	<a href="#"><u>setConnectLastSuccess</u></a> (String connectLastSuccess)
void	<a href="#"><u>setDebugPackets</u></a> (boolean debugPackets) Sets the debug packets.
void	<a href="#"><u>setDebugPackets</u></a> (String debugPacketsIn) Sets the debug packets.
void	<a href="#"><u>setDstAppInstanceName</u></a> (String dstAppInstanceName)
void	<a href="#"><u>setDstApplicationName</u></a> (String string) Sets the dst application name.
void	<a href="#"><u>setDstStreamName</u></a> (String dstStreamNameIn)
void	<a href="#"><u>setOnMetadataToSetDataFrame</u></a> (String onMetadataToSetDataFrame)
void	<a href="#"><u>setOriginalTimecodeThreshold</u></a> (String originalTimecodeThreshold) Sets the original timecode threshold.
void	<a href="#"><u>setReconnectWaitTime</u></a> (String reconnectWaitTime)
void	<a href="#"><u>setRemoveDefaultAppInstance</u></a> (String removeDefaultAppInstance) Sets the removes the default app instance.
void	<a href="#"><u>setResetOnTimecodeOutOfOrder</u></a> (String resetOnTimecodeOutOfOrder) Set to true if you wish push to reset if timecodes jump out of order.
void	<a href="#"><u>setSecureTokenSharedSecret</u></a> (String secureTokenSharedSecret) Set SecureToken shared secret.
void	<a href="#"><u>setSendFCAnnounce</u></a> (String sendFCAnnounce)
void	<a href="#"><u>setSendFCPublish</u></a> (String sendFCPublish) Sets the send fc publish.
void	<a href="#"><u>setSendOnMetadata</u></a> (String sendOnMetadata)
void	<a href="#"><u>setSendOriginalTimecodes</u></a> (String sendOriginalTimecodes) Sets the send original timecodes.
void	<a href="#"><u>setSendReleaseStream</u></a> (String sendReleaseStream) Sets the send release stream.
void	<a href="#"><u>setSendStreamCloseCommands</u></a> (String sendStreamCloseCommands) Sets the send stream close commands.
void	<a href="#"><u>setSSL</u></a> (String sslStr)
void	<a href="#"><u>setStreamCloseWaitTime</u></a> (String streamCloseWaitTime)
void	<a href="#"><u>setTimecodesOutOfOrderThreshold</u></a> (String timecodesOutOfOrderThreshold)
void	<a href="#"><u>setValidationFrequency</u></a> (String validationFrequency)

void	<a href="#">setWaitOnMetadataAudio(String waitOnMetadataAudio)</a>
void	<a href="#">setWaitOnMetadataAudioDatarate(String waitOnMetadataAudioDatarate)</a>
void	<a href="#">setWaitOnMetadataAvailable(String waitOnMetadataAvailable)</a>
void	<a href="#">setWaitOnMetadataVideo(String waitOnMetadataVideo)</a>
void	<a href="#">setWaitOnMetadataVideoDatarate(String waitOnMetadataVideoDatarate)</a>

**Methods inherited from class com.wowza.wms.pushpublish.protocol.rtmp.PushPublishRTMP**

[addConnectMetaDate](#), [addListener](#), [addSession](#), [callConnect](#), [connect](#),  
[convertDataEvenToAMFData](#), [convertOnMetadataToSetDataFrame](#), [createPushPublishSession](#),  
[disconnect](#), [getAkamaiOriginIp](#), [getAkamaiOriginPort](#), [getAkamaiPassword](#),  
[getAkamaiSessionKey](#), [getAkamaiUserName](#), [getAppInstance](#), [getConnectionFlashVerion](#),  
[getConnectionFlashVersion](#), [getConnectionPageURL](#), [getConnectionQueryStr](#),  
[getConnectionSwfURL](#), [getConnectionTimeout](#), [getConnectLastAttempt](#),  
[getConnectLastSuccess](#), [getContextStr](#), [getDstAppInstanceName](#), [getDstApplicationName](#),  
[getDstStreamName](#), [getHostname](#), [getHowToPublish](#), [getLastAudioTC](#), [getLastDataTC](#),  
[getLastSession](#), [getLastVideoTC](#), [getLocalListeners](#), [getLock](#), [getNetConnection](#),  
[getOriginalTimecodeThreshold](#), [getPacketsSentAudio](#), [getPacketsSentData](#),  
[getPacketsSentVideo](#), [getPort](#), [getPushPublishVersionStr](#), [getReconnectDelay](#),  
[getRTMPAuthProvider](#), [getSecureTokenSharedSecret](#), [getSessionHolderByNetConnection](#),  
[getSessionHolderBySession](#), [getSessionListDebug](#), [getSrcStreamBufferSize](#),  
[getSrcStreamName](#), [getStreamCloseWaitTime](#), [getTimecodesOutOfOrderThreshold](#),  
[getTotalPacketsSent](#), [getValidationFrequency](#), [getWaitOnMetadataTimeout](#),  
[internalConnect](#), [isCurrentSession](#), [isDebugLog](#), [isDebugPackets](#), [isFlashVerionFMLE](#),  
[isFlashVersionFMLE](#), [isOnMetadataToSetDataFrame](#), [isRemoveDefaultAppInstance](#),  
[isResetOnTimecodeOutOfOrder](#), [isSendFCAnnounce](#), [isSendFCPublish](#), [isSendOnMetadata](#),  
[isSendOriginalTimecodes](#), [isSendReleaseStream](#), [isSendStreamCloseCommands](#), [isSSL](#),  
[isWaitOnMetadataAudio](#), [isWaitOnMetadataAudioDatarate](#), [isWaitOnMetadataAvailable](#),  
[isWaitOnMetadataVideo](#), [isWaitOnMetadataVideoDatarate](#), [notifyAkamaiClientLogin](#),  
[notifyAkamaiSetChallenge](#), [notifyAkamaiSetOriginConnectionInfo](#), [notifyConnect](#),  
[notifyConnectFailure](#), [notifyConnectStart](#), [notifyConnectSuccess](#), [notifyFCAnnounce](#),  
[notifyFCPublish](#), [notifyHandshakeResult](#), [notifyPublishHandlerPlay](#),  
[notifyPushPublisherSessionCreate](#), [notifyPushPublisherSessionDestroy](#),  
[notifySessionClosed](#), [notifySessionIdle](#), [notifySessionOpened](#), [notifyStreamCreate](#),  
[notifyStreamOnPlayStatus](#), [notifyStreamOnStatus](#), [notifyValidateSession](#),  
[notifyValidateSessionResult](#), [removeListener](#), [removeSession](#), [resetConnection](#),  
[resetConnection](#), [sessionClosed](#), [sessionOpened](#), [setAkamaiOriginIp](#),  
[setAkamaiOriginPort](#), [setAkamaiPassword](#), [setAkamaiUserName](#), [setAppInstance](#),  
[setConnectionFlashVerion](#), [setConnectionFlashVersion](#), [setConnectionPageURL](#),  
[setConnectionQueryStr](#), [setConnectionSwfURL](#), [setConnectionTimeout](#),  
[setConnectLastAttempt](#), [setConnectLastSuccess](#), [setDebugLog](#), [setDebugPackets](#),  
[setDstAppInstanceName](#), [setDstApplicationName](#), [setDstStreamName](#), [setHostname](#),  
[setHowToPublish](#), [setOnMetadataToSetDataFrame](#), [setOriginalTimecodeThreshold](#), [setPort](#),  
[setRemoveDefaultAppInstance](#), [setResetOnTimecodeOutOfOrder](#), [setRTMPAuthProvider](#),  
[setSecureTokenSharedSecret](#), [setSendFCAnnounce](#), [setSendFCPublish](#), [setSendOnMetadata](#),  
[setSendOriginalTimecodes](#), [setSendReleaseStream](#), [setSendStreamCloseCommands](#),  
[setSrcStreamBufferSize](#), [setSrcStreamName](#), [setSSL](#), [setStreamCloseWaitTime](#),  
 [setTimecodesOutOfOrderThreshold](#), [setValidationFrequency](#), [setWaitOnMetadataAudio](#),  
[setWaitOnMetadataAudioDatarate](#), [setWaitOnMetadataAvailable](#), [setWaitOnMetadataTimeout](#),  
[setWaitOnMetadataVideo](#), [setWaitOnMetadataVideoDatarate](#), [tryToConnect](#)

**Methods inherited from class com.wowza.wms.pushpublish.model.PushPublishBase**

```
createPushPublishSession, getAppInstance, getBandwidth, getConfigDataOrDefault,
getConfigDataOrDefault, getConfigDataOrDefault, getContextStr, getDstStreamName,
getHostname, getImplementation, getMapFileIndex, getNumberFromString,
getOriginalTimecodeThreshold, getPassword, getPort, getProperties,
getPushPublishType, getPushPublishVersionStr, getReconnectWaitTime, getSrcStream,
getSrcStreamName, getUserName, init, isAdaptiveStreaming, isSendOriginalTimecodes,
load, setAdaptiveStreaming, setAppInstance, setDebugLog, setDstStreamName, setHost,
setHostname, setImplementation, setMapFileIndex, setOriginalTimecodeThreshold,
setOriginalTimecodeThreshold, setPassword, setPort, setPort, setReconnectWaitTime,
setSendOriginalTimecodes, setSrcStream, setSrcStreamName, setStreamName, setUserName
```

#### Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait,
wait
```

#### Methods inherited from interface com.wowza.wms.pushpublish.model.IPushPublish

```
connect, createPushPublishSession, disconnect, getContextStr, getDstStreamName,
getHostname, getMapFileIndex, getPort, getProperties, getPushPublishType,
getReconnectWaitTime, getSrcStream, getSrcStreamName, init, resetConnection,
setDebugLog, setDstStreamName, setHost, setHostname, setMapFileIndex, setPort,
setPort, setReconnectWaitTime, setSrcStream, setSrcStreamName, setStreamName
```

## Fields

### CLASS

```
public static final java.lang.Class CLASS
```

### CLASSNAME

```
public static final java.lang.String CLASSNAME
```

Constant value: **PushPublishRTMPPProfileHandler**

### streamId

```
protected java.lang.String streamId
```

The stream id.

## Constructors

### PushPublishRTMPPProfileHandler

```
public PushPublishRTMPPProfileHandler()
```

Instantiates a new push publisher rtmp.

#### Throws:

LicensingException - the licensing exception

## Methods

### **setConnectionTimeout**

```
public void setConnectionTimeout(String connectionTimeout)
```

---

### **setSendFCPublish**

```
public void setSendFCPublish(String sendFCPublish)
```

Sets the send fc publish.

**Parameters:**

sendFCPublish - the new send fc publish

---

### **setSendReleaseStream**

```
public void setSendReleaseStream(String sendReleaseStream)
```

Sets the send release stream.

**Parameters:**

sendReleaseStream - the new send release stream

---

### **setConnectionInfoQueryString**

```
public void setConnectionInfoQueryString(String connectionQueryStr)
```

Set the connection query string.

**Parameters:**

connectionQueryStr - connection query string

---

### **connect\_Out**

```
public void connect_Out()
```

Connect to server.

---

### **setConnectionFlashVersion**

```
public void setConnectionFlashVersion(String connectionFlashVersion)
```

Set Flash version used in connection metadata.

**Parameters:**

connectionFlashVersion - Flash version used in connection metadata

---

### **setConnectionInfoSwfURL**

```
public void setConnectionInfoSwfURL(String connectionSwfURL)
```

Set SWF URL used in connection metadata.

**Parameters:**

(continued from last page)

connectionSwfURL - SWF URL used in connection metadata

---

## setConnectionInfoPageURL

```
public void setConnectionInfoPageURL(String connectionPageURL)
```

Set Page URL used in connection metadata.

**Parameters:**

connectionPageURL - Page URL used in connection metadata

---

## setSecureTokenSharedSecret

```
public void setSecureTokenSharedSecret(String secureTokenSharedSecret)
```

Set SecureToken shared secret.

**Parameters:**

secureTokenSharedSecret - SecureToken shared secret

---

## setSendFCAnnounce

```
public void setSendFCAnnounce(String sendFCAnnounce)
```

---

## setValidationFrequency

```
public void setValidationFrequency(String validationFrequency)
```

---

## setConnectLastAttempt

```
public void setConnectLastAttempt(String connectLastAttempt)
```

---

## setReconnectWaitTime

```
public void setReconnectWaitTime(String reconnectWaitTime)
```

---

## setConnectLastSuccess

```
public void setConnectLastSuccess(String connectLastSuccess)
```

---

## setSendOriginalTimecodes

```
public void setSendOriginalTimecodes(String sendOriginalTimecodes)
```

Sets the send original timecodes.

**Parameters:**

sendOriginalTimecodes - the new send original timecodes

---

## setSendOnMetadata

```
public void setSendOnMetadata(String sendOnMetadata)
```

---

## setRemoveDefaultAppInstance

```
public void setRemoveDefaultAppInstance(String removeDefaultAppInstance)
```

Sets the removes the default app instance.

**Parameters:**

removeDefaultAppInstance - the new removes the default app instance

---

## setOnMetadataToSetDataFrame

```
public void setOnMetadataToSetDataFrame(String onMetadataToSetDataFrame)
```

---

## setSendStreamCloseCommands

```
public void setSendStreamCloseCommands(String sendStreamCloseCommands)
```

Sets the send stream close commands.

**Parameters:**

sendStreamCloseCommands - the new send stream close commands

---

## setStreamCloseWaitTime

```
public void setStreamCloseWaitTime(String streamCloseWaitTime)
```

---

## setOriginalTimecodeThreshold

```
public void setOriginalTimecodeThreshold(String originalTimecodeThreshold)
```

Sets the original timecode threshold.

**Parameters:**

originalTimecodeThreshold - the new original timecode threshold

---

## setResetOnTimecodeOutOfOrder

```
public void setResetOnTimecodeOutOfOrder(String resetOnTimecodeOutOfOrder)
```

Set to true if you wish push to reset if timecodes jump out of order.

**Parameters:**

resetOnTimecodeOutOfOrder - true if you wish push to reset if timecodes jump out of order

---

## setTimecodesOutOfOrderThreshold

```
public void setTimecodesOutOfOrderThreshold(String timecodesOutOfOrderThreshold)
```

---

(continued from last page)

---

**setWaitOnMetadataAvailable**

```
public void setWaitOnMetadataAvailable(String waitOnMetadataAvailable)
```

---

**setWaitOnMetadataVideo**

```
public void setWaitOnMetadataVideo(String waitOnMetadataVideo)
```

---

**setWaitOnMetadataAudio**

```
public void setWaitOnMetadataAudio(String waitOnMetadataAudio)
```

---

**setWaitOnMetadataVideoDatarate**

```
public void setWaitOnMetadataVideoDatarate(String waitOnMetadataVideoDatarate)
```

---

**setWaitOnMetadataAudioDatarate**

```
public void setWaitOnMetadataAudioDatarate(String waitOnMetadataAudioDatarate)
```

---

**setSSL**

```
public void setSSL(String sslStr)
```

---

**load**

```
public void load(java.util.HashMap dataMap)
```

---

**isDebugPackets**

```
public boolean isDebugPackets()
```

Checks if is debug packets.

**Returns:**

true, if is debug packets

---

**setDebugPackets**

```
public void setDebugPackets(boolean debugPackets)
```

Sets the debug packets.

---

(continued from last page)

**Parameters:**

debugPackets - the new debug packets

**setDebugPackets**

```
public void setDebugPackets(String debugPacketsIn)
```

Sets the debug packets.

**Parameters:**

debugPacketsIn - the new debug packets

**isDebugLog**

```
public boolean isDebugLog()
```

Is debug logging turned on.

**Returns:**

true if debug logging is turned on

**getDstApplicationName**

```
public String getDstApplicationName()
```

Get destination application name.

**Returns:**

destination application name

**setApplication**

```
public void setApplication(String dstApplicationNameIN)
```

Set destination application name.

**Parameters:**

dstApplicationNameIN - destination application name

**setDstApplicationName**

```
public void setDstApplicationName(String string)
```

Sets the dst application name.

**Parameters:**

string - the new dst application name

**getDstAppInstanceName**

```
public String getDstAppInstanceName()
```

Get destination appInstance name.

**Returns:**

destination appInstance name

**init**

```
public void init(IApplicationInstance appInstance2,  
    String streamName,  
    IMediaStream stream,  
    java.util.HashMap profileData,  
    java.util.HashMap maps,  
    boolean debug)
```

Inits the instance.

---

**setDstStreamName**

```
public void setDstStreamName(String dstStreamNameIn)
```

Set destination stream name

---

**setDstAppInstanceName**

```
public void setDstAppInstanceName(String dstAppInstanceName)
```

Set destination appInstance name

## com.wowza.wms.pushpublish.protocol.rtmp Class PushPublishRTMPUstream

```
java.lang.Object
+-com.wowza.wms.pushpublish.model.PushPublishBase
  +-com.wowza.wms.pushpublish.protocol.rtmp.PushPublishRTMP
    +-com.wowza.wms.pushpublish.protocol.rtmp.PushPublishRTMPPProfileHandler
      +-com.wowza.wms.pushpublish.protocol.rtmp.PushPublishRTMPUstream
```

All Implemented Interfaces:

[IPushPublish](#)

public class **PushPublishRTMPUstream**  
 extends [PushPublishRTMPPProfileHandler](#)

### Field Summary

public static final	<a href="#">CLASS</a>
public static final	<a href="#">CLASSNAME</a>  Value: <b>PushPublishRTMPUstream</b>

Fields inherited from class [com.wowza.wms.pushpublish.protocol.rtmp.PushPublishRTMPPProfileHandler](#)

[CLASS](#), [CLASSNAME](#), [streamId](#)

Fields inherited from class [com.wowza.wms.pushpublish.protocol.rtmp.PushPublishRTMP](#)

[akamaiOriginIp](#), [akamaiOriginPort](#), [akamaiPassword](#), [akamaiUserName](#), [bytesAvg](#), [CLASS](#), [CLASSNAME](#), [connectionFlashVersion](#), [connectionPageURL](#), [connectionQueryStr](#), [connectionSwfURL](#), [connectionTimeout](#), [connectLastAttempt](#), [connectLastSuccess](#), [connectMetaData](#), [CURRENTFLASHVERSION](#), [CURRENTFMLEVERSION](#), [debugLogLastPacketCount](#), [debugPackets](#), [doSendOnMetaData](#), [dstAppInstanceName](#), [dstApplicationName](#), [dstStreamName](#), [howToPublish](#), [ioSessionToSessionHolder](#), [isReconnectorRunning](#), [isSSL](#), [lastAudioTC](#), [lastDataTC](#), [lastIdleReadTime](#), [lastIdleWriteTime](#), [lastVideoTC](#), [listeners](#), [lock](#), [netconnectToSessionHolder](#), [offlineTime](#), [onlineTime](#), [onMetadataToSetDataFrame](#), [outprint](#), [packetsSentAudio](#), [packetsSentData](#), [packetsSentVideo](#), [removeDefaultAppInstance](#), [resetOnTimecodeOutOfOrder](#), [rtmpAuthProvider](#), [rtmpDebugLog](#), [secureTokenSharedSecret](#), [sendFCAnnounce](#), [sendFCPublish](#), [sendOnMetadata](#), [sendOriginalTimecodes](#), [sendReleaseStream](#), [sendStreamCloseCommands](#), [sendStreamCloseCommandsSendTime](#), [sessionOrder](#), [srcStreamBufferSize](#), [streamCloseWaitTime](#), [timecodesOutOfOrderThreshold](#), [totalPacketsSent](#), [tryConnect](#), [validationFrequency](#), [waitOnMetadataAudio](#), [waitOnMetadataAudioDatarate](#), [waitOnMetadataAvailable](#), [waitOnMetadataReady](#), [waitOnMetadataStartime](#), [waitOnMetadataTimeout](#), [waitOnMetadataVideo](#), [waitOnMetadataVideoDatarate](#)

Fields inherited from class [com.wowza.wms.pushpublish.model.PushPublishBase](#)

```
appInstance, CLASS, CLASSNAME, dstStreamName, hostname, mapFileIndex, mImplementation, mType,
originalTimecodeThreshold, password, port, properties, pushPublishStreamDebug,
pushPublishType, reconnectWaitTime, sendingPerfomance, sendOriginalTimecodes, srcStream,
srcStreamName, userName
```

Fields inherited from interface [com.wowza.wms.pushpublish.model.IPushPublish](#)

[PUSHPUBLISH\\_TYPE\\_RTMP](#), [PUSHPUBLISH\\_TYPE\\_RTP](#), [PUSHPUBLISH\\_TYPE\\_UNKNOWN](#)

## Constructor Summary

public	<a href="#">PushPublishRTMPUstream()</a>
--------	--

## Method Summary

void	<a href="#">callConnect(PushPublishRTMPNetConnectionSession pushPublisherSession, String addQueryStr)</a>
void	<a href="#">init(IApplicationInstance appInstance2, String streamName, IMediaStream stream, java.util.HashMap profileData, java.util.HashMap maps, boolean debug)</a>
org.apache.mina.common.IoSession	<a href="#">internalConnect(PushPublishRTMPNetConnectionSession pushPublisherSession)</a>
boolean	<a href="#">isUseUstreamPingPong()</a>
void	<a href="#">load(java.util.HashMap dataMap)</a>
void	<a href="#">setUseUstreamPingPong(boolean useUstreamPingPong)</a>

### Methods inherited from class

[com.wowza.wms.pushpublish.protocol.rtmp.PushPublishRTMPPProfileHandler](#)

```
connect_Out, getDstAppInstanceName, getDstApplicationName, init, isDebugEnabled,
isDebugEnabledPackets, load, setApplication, setConnectionFlashVersion,
setConnectionInfoPageURL, setConnectionInfoQueryString, setConnectionInfoSwfURL,
setConnectionTimeout, setConnectLastAttempt, setConnectLastSuccess, setDebugEnabledPackets,
setDebugEnabledPackets, setDstAppInstanceName, setDstApplicationName, setDstStreamName,
setOnMetadataToSetDataFrame, setOriginalTimecodeThreshold, setReconnectWaitTime,
setRemoveDefaultAppInstance, setResetOnTimecodeOutOfOrder,
setSecureTokenSharedSecret, setSendFCAnnounce, setSendFCPublish, setSendOnMetadata,
setSendOriginalTimecodes, setSendReleaseStream, setSendStreamCloseCommands, setSSL,
setStreamCloseWaitTime, setTimecodesOutOfOrderThreshold, setValidationFrequency,
setWaitOnMetadataAudio, setWaitOnMetadataAudioDatarate, setWaitOnMetadataAvailable,
setWaitOnMetadataVideo, setWaitOnMetadataVideoDatarate
```

Methods inherited from class [com.wowza.wms.pushpublish.protocol.rtmp.PushPublishRTMP](#)

```

addConnectMetaData, addListener, addSession, callConnect, connect,
convertDataEvenToAMFData, convertOnMetadataToSetDataFrame, createPushPublishSession,
disconnect, getAkamaiOriginIp, getAkamaiOriginPort, getAkamaiPassword,
getAkamaiSessionKey, getAkamaiUserName, getAppInstance, getConnectionFlashVerion,
getConnectionFlashVersion, getConnectionPageURL, getConnectionQueryStr,
getConnectionSwfURL, getConnectionTimeout, getConnectLastAttempt,
getConnectLastSuccess, getContextStr, getDstAppInstanceName, getDstApplicationName,
getDstStreamName, getHostname, getHowToPublish, getLastAudioTC, getLastDataTC,
getLastSession, getLastVideoTC, getLocalListeners, getLock, getNetConnection,
getOriginalTimecodeThreshold, getPacketsSentAudio, getPacketsSentData,
getPacketsSentVideo, getPort, getPushPublishVersionStr, getReconnectDelay,
getRTMPAuthProvider, getSecureTokenSharedSecret, getSessionHolderByNetConnection,
getSessionHolderBySession, getSessionListDebug, getSrcStreamBufferSize,
getSrcStreamName, getStreamCloseWaitTime, getTimecodesOutOfOrderThreshold,
getTotalPacketsSent, getValidationFrequency, getWaitOnMetadataTimeout,
internalConnect, isCurrentSession, isDebugLog, isDebugPackets, isFlashVerionFMLE,
isFlashVersionFMLE, isOnMetadataToSetDataFrame, isRemoveDefaultAppInstance,
isResetOnTimecodeOutOfOrder, isSendFCAnnounce, isSendFCPublish, isSendOnMetadata,
isSendOriginalTimecodes, isSendReleaseStream, isSendStreamCloseCommands, isSSL,
isWaitOnMetadataAudio, isWaitOnMetadataAudioDatarate, isWaitOnMetadataAvailable,
isWaitOnMetadataVideo, isWaitOnMetadataVideoDatarate, notifyAkamaiClientLogin,
notifyAkamaiSetChallenge, notifyAkamaiSetOriginConnectionInfo, notifyConnect,
notifyConnectFailure, notifyConnectStart, notifyConnectSuccess, notifyFCAnnounce,
notifyFCPublish, notifyHandshakeResult, notifyPublishHandlerPlay,
notifyPushPublisherSessionCreate, notifyPushPublisherSessionDestroy,
notifySessionClosed, notifySessionIdle, notifySessionOpened, notifyStreamCreate,
notifyStreamOnPlayStatus, notifyStreamOnStatus, notifyValidateSession,
notifyValidateSessionResult, removeListener, removeSession, resetConnection,
resetConnection, sessionClosed, sessionOpened, setAkamaiOriginIp,
setAkamaiOriginPort, setAkamaiPassword, setAkamaiUserName, setAppInstance,
setConnectionFlashVerion, setConnectionFlashVersion, setConnectionPageURL,
setConnectionQueryStr, setConnectionSwfURL, setConnectionTimeout,
setConnectLastAttempt, setConnectLastSuccess, setDebugLog, setDebugPackets,
setDstAppInstanceName, setDstApplicationName, setDstStreamName, setHostname,
setHowToPublish, setOnMetadataToSetDataFrame, setOriginalTimecodeThreshold, setPort,
setRemoveDefaultAppInstance, setResetOnTimecodeOutOfOrder, setRTMPAuthProvider,
setSecureTokenSharedSecret, setSendFCAnnounce, setSendFCPublish, setSendOnMetadata,
setSendOriginalTimecodes, setSendReleaseStream, setSendStreamCloseCommands,
setSrcStreamBufferSize, setSrcStreamName, setSSL, setStreamCloseWaitTime,
setTimecodesOutOfOrderThreshold, setValidationFrequency, setWaitOnMetadataAudio,
setWaitOnMetadataAudioDatarate, setWaitOnMetadataAvailable, setWaitOnMetadataTimeout,
setWaitOnMetadataVideo, setWaitOnMetadataVideoDatarate, tryToConnect

```

#### Methods inherited from class com.wowza.wms.pushpublish.model.PushPublishBase

```

createPushPublishSession, getAppInstance, getBandwidth, getConfigDataOrDefault,
getConfigDataOrDefault, getConfigDataOrDefault, getContextStr, getDstStreamName,
getHostname, getImplementation, getMapFileIndex, getNumberFromString,
getOriginalTimecodeThreshold, getPassword, getPort, getProperties,
getPushPublishType, getPushPublishVersionStr, getReconnectWaitTime, getSrcStream,
getSrcStreamName, getUserName, init, isAdaptiveStreaming, isSendOriginalTimecodes,
load, setAdaptiveStreaming, setAppInstance, setDebugLog, setDstStreamName, setHost,
setHostname, setImplementation, setMapFileIndex, setOriginalTimecodeThreshold,
setOriginalTimecodeThreshold, setPassword, setPort, setPort, setReconnectWaitTime,
setSendOriginalTimecodes, setSrcStream, setSrcStreamName, setStreamName, setUserName

```

#### Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait,
wait
```

#### Methods inherited from interface com.wowza.wms.pushpublish.model.IPushPublish

```
connect, createPushPublishSession, disconnect, getContextStr, getDstStreamName,
getHostname, getMapFileIndex, getPort, getProperties, getPushPublishType,
getReconnectWaitTime, getSrcStream, getSrcStreamName, init, resetConnection,
setDebugLog, setDstStreamName, setHost, setHostname, setMapFileIndex, setPort,
setPort, setReconnectWaitTime, setSrcStream, setSrcStreamName, setStreamName
```

## Fields

### CLASS

```
public static final java.lang.Class CLASS
```

---

### CLASSNAME

```
public static final java.lang.String CLASSNAME
```

Constant value: PushPublishRTMPUstream

## Constructors

### PushPublishRTMPUstream

```
public PushPublishRTMPUstream()
```

## Methods

### isUseUstreamPingPong

```
public boolean isUseUstreamPingPong()
```

---

### setUseUstreamPingPong

```
public void setUseUstreamPingPong(boolean useUstreamPingPong)
```

(continued from last page)

**init**

```
public void init(IAplicationInstance appInstance2,  
    String streamName,  
    IMediaStream stream,  
    java.util.HashMap profileData,  
    java.util.HashMap maps,  
    boolean debug)
```

Inits the instance.

---

**load**

```
public void load(java.util.HashMap dataMap)
```

---

**internalConnect**

```
protected org.apache.mina.common.IoSession  
internalConnect(PushPublishRTMPNetConnectionSession pushPublisherSession)
```

Internal connect method

---

**callConnect**

```
protected void callConnect(PushPublishRTMPNetConnectionSession pushPublisherSession,  
    String addQueryStr)
```

Internal connect

## com.wowza.wms.pushpublish.protocol.rtmp Class PushPublishSessionRTMP

```
java.lang.Object
+- com.wowza.wms.pushpublish.model.PushPublishSessionBase
  +- com.wowza.wms.pushpublish.protocol.rtmp.PushPublishSessionRTMP
```

### All Implemented Interfaces:

[IPushPublishSession](#)

public class **PushPublishSessionRTMP**

extends [PushPublishSessionBase](#)

Fields inherited from class [com.wowza.wms.pushpublish.model.PushPublishSessionBase](#)

[profile](#), [pushPublishImpl](#), [stream](#), [type](#)

Fields inherited from interface [com.wowza.wms.pushpublish.model.IPushPublishSession](#)

[PUSHPUBLISHSESSION\\_TYPE\\_RTMP](#), [PUSHPUBLISHSESSION\\_TYPE RTP](#),  
[PUSHPUBLISHSESSION\\_TYPE\\_UNKNOWN](#)

## Constructor Summary

public	<a href="#">PushPublishSessionRTMP()</a>
--------	--

Methods inherited from class [com.wowza.wms.pushpublish.model.PushPublishSessionBase](#)

[getContextStr](#), [getProfile](#), [getPushPublishImpl](#), [getStream](#), [getType](#), [setProfile](#),  
[setPushPublishImpl](#), [setStream](#)

Methods inherited from class [java.lang.Object](#)

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#)

Methods inherited from interface [com.wowza.wms.pushpublish.model.IPushPublishSession](#)

[getContextStr](#), [getProfile](#), [getPushPublishImpl](#), [getStream](#), [getType](#), [setProfile](#),  
[setPushPublishImpl](#), [setStream](#)

## Constructors

### PushPublishSessionRTMP

public [PushPublishSessionRTMP\(\)](#)

---

**Package**

**com.wowza.wms.pushpublish.protocol.rtp**

## com.wowza.wms.pushpublish.protocol.rtp Class PushPublishRTP

```
java.lang.Object
+- com.wowza.wms.pushpublish.model.PushPublishBase
  +- com.wowza.wms.pushpublish.protocol.rtp.PushPublishRTP
```

### All Implemented Interfaces:

[IPushPublish](#)

public class **PushPublishRTP**

extends [PushPublishBase](#)

The Class PushPublisherRTP.

### Field Summary

protected	<a href="#">audioHost</a> The audio host.
protected	<a href="#">audioPort</a> The audio port.
public static final	<a href="#">CLASS</a>
public static final	<a href="#">CLASSNAME</a> Value: <b>PushPublishRTP</b>
protected	<a href="#">isRTPWrapped</a> The is rtp wrapped.
protected	<a href="#">mRtpDestination</a> The m rtp destination.
protected	<a href="#">removeSDPFiles</a> The remove sdp files
protected	<a href="#">rtpPushPublishSession</a> The rtp push publish session.
public static final	<a href="#">SDPFILEEXTENSION</a> The Constant SDPFILEEXTENSION. Value: <b>.sdp</b>
protected	<a href="#">singleStreamAV</a> The stream packetizer.
protected	<a href="#">videoHost</a> The video host.
protected	<a href="#">videoPort</a> The video port.

protected	<a href="#"><u>writeSDPFiles</u></a> The write sdp files.
-----------	--

**Fields inherited from class** [com.wowza.wms.pushpublish.model.PushPublishBase](#)

[appInstance](#), [CLASS](#), [CLASSNAME](#), [dstStreamName](#), [hostname](#), [mapFileIndex](#), [mImplementation](#), [mType](#), [originalTimecodeThreshold](#), [password](#), [port](#), [properties](#), [pushPublishStreamDebug](#), [pushPublishType](#), [reconnectWaitTime](#), [sendingPerfomance](#), [sendOriginalTimecodes](#), [srcStream](#), [srcStreamName](#), [userName](#)

**Fields inherited from interface** [com.wowza.wms.pushpublish.model.IPushPublish](#)

[PUSHPUBLISH\\_TYPE\\_RTMP](#), [PUSHPUBLISH\\_TYPE\\_RTP](#), [PUSHPUBLISH\\_TYPE\\_UNKNOWN](#)

## Constructor Summary

public	<a href="#"><u>PushPublishRTP()</u></a>
--------	---

## Method Summary

void	<a href="#"><u>connect()</u></a>
<a href="#">IPushPublishSession</a>	<a href="#"><u>createPushPublishSession()</u></a>
void	<a href="#"><u>disconnect()</u></a>
<a href="#">IApplicationInstance</a>	<a href="#"><u>getAppInstance()</u></a>
String	<a href="#"><u>getAudioHost()</u></a> Gets the audio host.
int	<a href="#"><u>getAudioPort()</u></a> Gets the audio port.
String	<a href="#"><u>getContextStr()</u></a>
String	<a href="#"><u>getHostname()</u></a>
String	<a href="#"><u>getHostType()</u></a> Gets the host type.
<a href="#">RTPDestination</a>	<a href="#"><u>getmRtpDestination()</u></a>
<a href="#">RTPPushPublishSession</a>	<a href="#"><u>getRTPPushPublishSession()</u></a> Gets the rTP push publish session.
long	<a href="#"><u>getRtpStreamWaitTimeout()</u></a>
String	<a href="#"><u>getSrcStreamName()</u></a>
String	<a href="#"><u>getVideoHost()</u></a> Gets the video host.

int	<a href="#"><u>getVideoPort()</u></a> Gets the video port.
void	<a href="#"><u>init(IApplicationInstance appInstance, String streamName, IMediaStream stream, java.util.HashMap profileData, java.util.HashMap maps, boolean debug)</u></a>
boolean	<a href="#"><u>isRTPWrapped()</u></a> Checks if is rTP wrapped.
boolean	<a href="#"><u>isSingleStreamAV()</u></a> Checks if is stream is packatized.
void	<a href="#"><u>load(java.util.HashMap dataMap)</u></a>
void	<a href="#"><u>resetConnection()</u></a>
void	<a href="#"><u>resetConnection(int delay)</u></a>
void	<a href="#"><u>setAppInstance(IApplicationInstance appInstance)</u></a>
void	<a href="#"><u>setAudioHost(String audioHost)</u></a> Sets the audio host.
void	<a href="#"><u>setAudioPort(int audioPort)</u></a> Sets the audio port.
void	<a href="#"><u>setAudioPort(String audioPort)</u></a> Sets the audio port.
void	<a href="#"><u>setHost(String hostname)</u></a>
void	<a href="#"><u>setHostType(String hostType)</u></a> Sets the host type.
void	<a href="#"><u>setPort(int port)</u></a> Set destination server port.
void	<a href="#"><u>setPort(String portIn)</u></a>
void	<a href="#"><u>setRTPPushPublishSession(RTPPushPublishSession rtpPushPublishSession)</u></a> Sets the rTP push publish session.
void	<a href="#"><u>setRtpStreamWaitTimeout(long rtpStreamWaitTimeout)</u></a>
void	<a href="#"><u>setRtpStreamWaitTimeout(String rtpStreamWaitTimeout)</u></a>
void	<a href="#"><u>setRTPWrapped(String string)</u></a> Sets the rTP wrapped.
void	<a href="#"><u>setSingleStreamAV(boolean inSingleStreamAV)</u></a> Sets the stream packetizer.
void	<a href="#"><u>setSingleStreamAV(String inSingleStreamAV)</u></a>

void	<a href="#">setSrcStreamName</a> (String srcStreamName)
void	<a href="#">setTimeToLive</a> (String ttl) Sets the time to live.
void	<a href="#">setVideoHost</a> (String videoHost) Sets the video host.
void	<a href="#">setVideoPort</a> (String videoPort) Sets the video port.

**Methods inherited from class** [com.wowza.wms.pushpublish.model.PushPublishBase](#)

[createPushPublishSession](#), [getAppInstance](#), [getBandwidth](#), [getConfigDataOrDefault](#), [getConfigDataOrDefault](#), [getContextStr](#), [getDstStreamName](#), [getHostname](#), [getImplementation](#), [getMapFileIndex](#), [getNumberFromString](#), [getOriginalTimecodeThreshold](#), [getPassword](#), [getPort](#), [getProperties](#), [getPushPublishType](#), [getPushPublishVersionStr](#), [getReconnectWaitTime](#), [getSrcStream](#), [getSrcStreamName](#), [getUserName](#), [init](#), [isAdaptiveStreaming](#), [isSendOriginalTimecodes](#), [load](#), [setAdaptiveStreaming](#), [setAppInstance](#), [setDebugLog](#), [setDstStreamName](#), [setHost](#), [setHostname](#), [setImplementation](#), [setMapFileIndex](#), [setOriginalTimecodeThreshold](#), [setOriginalTimecodeThreshold](#), [setPassword](#), [setPort](#), [setPort](#), [setReconnectWaitTime](#), [setSendOriginalTimecodes](#), [setSrcStream](#), [setSrcStreamName](#), [setStreamName](#), [setUserName](#)

**Methods inherited from class** [java.lang.Object](#)

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#)

**Methods inherited from interface** [com.wowza.wms.pushpublish.model.IPushPublish](#)

[connect](#), [createPushPublishSession](#), [disconnect](#), [getContextStr](#), [getDstStreamName](#), [getHostname](#), [getMapFileIndex](#), [getPort](#), [getProperties](#), [getPushPublishType](#), [getReconnectWaitTime](#), [getSrcStream](#), [getSrcStreamName](#), [init](#), [resetConnection](#), [setDebugLog](#), [setDstStreamName](#), [setHost](#), [setHostname](#), [setMapFileIndex](#), [setPort](#), [setPort](#), [setReconnectWaitTime](#), [setSrcStream](#), [setSrcStreamName](#), [setStreamName](#)

## Fields

### CLASS

```
public static final java.lang.Class CLASS
```

### CLASSNAME

```
public static final java.lang.String CLASSNAME
```

Constant value: **PushPublishRTP**

### SDPFILEEXTENSION

```
public static final java.lang.String SDPFILEEXTENSION
```

(continued from last page)

The Constant SDPFILEEXTENSION.  
Constant value: `.sdp`

---

## rtpPushPublishSession

protected com.wowza.wms.rtp.model.RTPPushPublishSession **rtpPushPublishSession**

The rtp push publish session.

---

## singleStreamAV

protected boolean **singleStreamAV**

The stream packetizer.

---

## audioPort

protected int **audioPort**

The audio port.

---

## videoPort

protected int **videoPort**

The video port.

---

## audioHost

protected java.lang.String **audioHost**

The audio host.

---

## videoHost

protected java.lang.String **videoHost**

The video host.

---

## isRTPWrapped

protected boolean **isRTPWrapped**

The is rtp wrapped.

---

## writeSDPFiles

protected boolean **writeSDPFiles**

The write sdp files.

---

## removeSDPFiles

protected boolean **removeSDPFiles**

The remove sdp files

---

(continued from last page)

## mRtpDestination

```
protected com.wowza.wms.rtp.model.RTPDestination mRtpDestination
```

The m rtp destination.

## Constructors

### PushPublishRTP

```
public PushPublishRTP()
```

## Methods

### init

```
public void init(IAplicationInstance appInstance,  
String streamName,  
IMediaStream stream,  
java.util.HashMap profileData,  
java.util.HashMap maps,  
boolean debug)
```

Inits the instance.

### connect

```
public void connect()
```

### disconnect

```
public void disconnect()
```

### resetConnection

```
public void resetConnection()
```

### resetConnection

```
public void resetConnection(int delay)
```

### getRTPPushPublishSession

```
public RTTPushPublishSession getRTTPushPublishSession()
```

Gets the rTP push publish session.

#### Returns:

the rTP push publish session

## setRTPPushPublishSession

```
public void setRTPPushPublishSession(RTPPushPublishSession rtpPushPublishSession)
```

Sets the rTP push publish session.

**Parameters:**

rtpPushPublishSession - the new rTP push publish session

---

## getAppInstance

```
public IApplicationInstance getAppInstance()
```

Get appInstance interface.

---

## setAppInstance

```
public void setAppInstance(IApplicationInstance appInstance)
```

Set appInstance interface.

---

## getSrcStreamName

```
public String getSrcStreamName()
```

Get source stream name.

---

## setSrcStreamName

```
public void setSrcStreamName(String srcStreamName)
```

Set source stream name.

---

## isSingleStreamAV

```
public boolean isSingleStreamAV()
```

Checks if is stream is packatized.

**Returns:**

true, if is stream packetizer

---

## setSingleStreamAV

```
public void setSingleStreamAV(boolean inSingleStreamAV)
```

Sets the stream packetizer.

**Parameters:**

inSingleStreamAV - whether stream is packatized

---

## setSingleStreamAV

```
public void setSingleStreamAV(String inSingleStreamAV)
```

**getHostname**

```
public String getHostname()
```

Get destination server host name.

---

**setHost**

```
public void setHost(String hostname)
```

Set destination server host name.

---

**setPort**

```
public void setPort(int port)
```

Set destination server port.

**Parameters:**

port - destination server port

---

**setPort**

```
public void setPort(String portIn)
```

**getAudioPort**

```
public int getAudioPort()
```

Gets the audio port.

**Returns:**

the audio port

---

**setAudioPort**

```
public void setAudioPort(int audioPort)
```

Sets the audio port.

**Parameters:**

audioPort - the new audio port

---

**getAudioHost**

```
public String getAudioHost()
```

Gets the audio host.

**Returns:**

the audio host

---

**setAudioHost**

```
public void setAudioHost(String audioHost)
```

---

(continued from last page)

Sets the audio host.

**Parameters:**

audioHost - the new audio host

---

## getVideoHost

```
public String getVideoHost()
```

Gets the video host.

**Returns:**

the video host

---

## setVideoHost

```
public void setVideoHost(String videoHost)
```

Sets the video host.

**Parameters:**

videoHost - the new video host

---

## setAudioPort

```
public void setAudioPort(String audioPort)
```

Sets the audio port.

**Parameters:**

audioPort - the new audio port

---

## getVideoPort

```
public int getVideoPort()
```

Gets the video port.

**Returns:**

the video port

---

## setVideoPort

```
public void setVideoPort(String videoPort)
```

Sets the video port.

**Parameters:**

videoPort - the new video port

---

## isRTPWrapped

```
public boolean isRTPWrapped()
```

Checks if is rTP wrapped.

**Returns:**

true, if is rTP wrapped

## setRTPWrapped

```
public void setRTPWrapped(String string)
```

Sets the rTP wrapped.

**Parameters:**

string - the new rTP wrapped

---

## getHostType

```
public String getHostType()
```

Gets the host type.

**Returns:**

the host type

---

## setHostType

```
public void setHostType(String hostType)
```

Sets the host type.

**Parameters:**

hostType - the new host type

---

## setTimeToLive

```
public void setTimeToLive(String ttl)
```

Sets the time to live.

**Parameters:**

ttl - the new time to live

---

## load

```
public void load(java.util.HashMap dataMap)
```

---

## getRtpStreamWaitTimeout

```
public long getRtpStreamWaitTimeout()
```

---

## setRtpStreamWaitTimeout

```
public void setRtpStreamWaitTimeout(String rtpStreamWaitTimeout)
```

---

## setRtpStreamWaitTimeout

```
public void setRtpStreamWaitTimeout(long rtpStreamWaitTimeout)
```

---

(continued from last page)

---

### **createPushPublishSession**

```
public IPushPublishSession createPushPublishSession()
```

---

### **getContextStr**

```
public String getContextStr()
```

---

### **getmRtpDestination**

```
public RTPDestination getmRtpDestination()
```

## com.wowza.wms.pushpublish.protocol.rtp Class PushPublishRTPRunner

```
java.lang.Object
  +-- java.lang.Thread
    +-- com.wowza.wms.pushpublish.protocol.rtp.PushPublishRTPRunner
```

All Implemented Interfaces:  
Runnable

---

```
public class PushPublishRTPRunner
extends Thread
```

The Class RTPStarter. Utility class to starts an RTP session for push publishing

---

### Field Summary

public static final	<a href="#">CLASS</a>
public static final	<a href="#">CLASSNAME</a> Value: <b>RTPStarter</b>

### Fields inherited from class java.lang.Thread

```
MAX_PRIORITY, MIN_PRIORITY, NORM_PRIORITY
```

### Constructor Summary

public	<a href="#">PushPublishRTPRunner</a> ( <a href="#">PushPublishRTP</a> pushPublisherRTP, <a href="#">RTPDestination</a> rtpDestination)
	Instantiates a new rTP starter.

### Method Summary

void	<a href="#">run()</a>
------	-----------------------

### Methods inherited from class java.lang.Thread

```
activeCount, checkAccess, clone, countStackFrames, currentThread, destroy, dumpStack,
enumerate, getAllStackTraces, getContextClassLoader,
getDefaultUncaughtExceptionHandler, getId, getName, getPriority, getStackTrace,
getState, getThreadGroup, getUncaughtExceptionHandler, holdsLock, interrupt,
interrupted, isAlive, isDaemon, isInterrupted, join, join, join, resume, run,
setContextClassLoader, setDaemon, setDefaultUncaughtExceptionHandler, setName,
setPriority, setUncaughtExceptionHandler, sleep, sleep, start, stop, stop, suspend,
toString, yield
```

### Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait,
```

Methods inherited from interface java.lang.Runnable

run

## Fields

### CLASS

```
public static final java.lang.Class CLASS
```

---

### CLASSNAME

```
public static final java.lang.String CLASSNAME
```

Constant value: RTPStarter

## Constructors

### PushPublishRTPRunner

```
public PushPublishRTPRunner(PushPublishRTP pushPublisherRTP,  
RTPDestination rtpDestination)
```

Instantiates a new rTP starter.

**Parameters:**

pushPublisherRTP - the push publisher rtp  
rtpDestination - the rtp destination

## Methods

### run

```
public void run()
```

## com.wowza.wms.pushpublish.protocol.rtp Class PushPublishRTSPActionNotify

```
java.lang.Object
+-com.wowza.wms.pushpublish.protocol.rtp.PushPublishRTSPActionNotify
```

All Implemented Interfaces:  
[IRTSPActionNotify](#)

---

```
public class PushPublishRTSPActionNotify
extends Object
implements IRTSPActionNotify
```

The Class PushPublishRTSPActionNotify.

### Constructor Summary

<code>public</code>	<code><a href="#">PushPublishRTSPActionNotify</a>(com.wowza.wms.pushpublish.module.ModulePushPublish modulePushPublish)</code>
	Instantiates a new push publish rtsp action notify.

### Method Summary

<code>void</code>	<code><a href="#">onAnnounce(RTPSession</a> rtspSession, com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp)</code>
<code>void</code>	<code><a href="#">onDescribe(RTPSession</a> rtspSession, com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp)</code>
<code>void</code>	<code><a href="#">onGetParameter(RTPSession</a> rtspSession, com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp)</code>
<code>void</code>	<code><a href="#">onOptions(RTPSession</a> rtspSession, com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp)</code>
<code>void</code>	<code><a href="#">onPause(RTPSession</a> rtspSession, com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp)</code>
<code>void</code>	<code><a href="#">onPlay(RTPSession</a> rtspSession, com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp)</code>
<code>void</code>	<code><a href="#">onRecord(RTPSession</a> rtspSession, com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp)</code>
<code>void</code>	<code><a href="#">onRedirect(RTPSession</a> rtspSession, com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp)</code>

void	<a href="#"><u>onSetParameter(RTPSession rtspSession, com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp)</u></a>
void	<a href="#"><u>onSetup(RTPSession rtspSession, com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp)</u></a>
void	<a href="#"><u>onTeardown(RTPSession rtspSession, com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp)</u></a>

**Methods inherited from class java.lang.Object**

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait`

**Methods inherited from interface com.wowza.wms.rtp.model.IRTSPActionNotify**

[onAnnounce](#), [onDescribe](#), [onGetParameter](#), [onOptions](#), [onPause](#), [onPlay](#), [onRecord](#), [onRedirect](#), [onSetParameter](#), [onSetup](#), [onTeardown](#)

## Constructors

### PushPublishRTSPActionNotify

```
public PushPublishRTSPActionNotify(com.wowza.wms.pushpublish.module.ModulePushPublish modulePushPublish)
```

Instantiates a new push publish rtsp action notify.

**Parameters:**

modulePushPublish - the module push publish

## Methods

### onDescribe

```
public void onDescribe(RTPSession rtspSession, com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp)
```

### onAnnounce

```
public void onAnnounce(RTPSession rtspSession, com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp)
```

### onGetParameter

```
public void onGetParameter(RTPSession rtspSession, com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp)
```

(continued from last page)

---

**onSetParameter**

```
public void onSetParameter(RTPSession rtspSession,  
    com.wowza.wms.rtsp.RTSPRequestMessage req,  
    com.wowza.wms.rtsp.RTSPResponseMessages resp)
```

---

**onOptions**

```
public void onOptions(RTPSession rtspSession,  
    com.wowza.wms.rtsp.RTSPRequestMessage req,  
    com.wowza.wms.rtsp.RTSPResponseMessages resp)
```

---

**onPause**

```
public void onPause(RTPSession rtspSession,  
    com.wowza.wms.rtsp.RTSPRequestMessage req,  
    com.wowza.wms.rtsp.RTSPResponseMessages resp)
```

---

**onPlay**

```
public void onPlay(RTPSession rtspSession,  
    com.wowza.wms.rtsp.RTSPRequestMessage req,  
    com.wowza.wms.rtsp.RTSPResponseMessages resp)
```

---

**onRecord**

```
public void onRecord(RTPSession rtspSession,  
    com.wowza.wms.rtsp.RTSPRequestMessage req,  
    com.wowza.wms.rtsp.RTSPResponseMessages resp)
```

---

**onRedirect**

```
public void onRedirect(RTPSession rtspSession,  
    com.wowza.wms.rtsp.RTSPRequestMessage req,  
    com.wowza.wms.rtsp.RTSPResponseMessages resp)
```

---

**onSetup**

```
public void onSetup(RTPSession rtspSession,  
    com.wowza.wms.rtsp.RTSPRequestMessage req,  
    com.wowza.wms.rtsp.RTSPResponseMessages resp)
```

---

(continued from last page)

## onTeardown

```
public void onTeardown(RTPSession rtspSession,  
                      com.wowza.wms.rtsp.RTSPRequestMessage req,  
                      com.wowza.wms.rtsp.RTSPResponseMessages resp)
```

## com.wowza.wms.pushpublish.protocol.rtp Class PushPublishSessionRTP

```
java.lang.Object
+- com.wowza.wms.pushpublish.model.PushPublishSessionBase
  +- com.wowza.wms.pushpublish.protocol.rtp.PushPublishSessionRTP
```

### All Implemented Interfaces:

[IPushPublishSession](#)

public class **PushPublishSessionRTP**  
 extends [PushPublishSessionBase](#)

Fields inherited from class [com.wowza.wms.pushpublish.model.PushPublishSessionBase](#)

[profile](#), [pushPublishImpl](#), [stream](#), [type](#)

Fields inherited from interface [com.wowza.wms.pushpublish.model.IPushPublishSession](#)

[PUSHPUBLISHSESSION\\_TYPE\\_RTMP](#), [PUSHPUBLISHSESSION\\_TYPE RTP](#),  
[PUSHPUBLISHSESSION\\_TYPE\\_UNKNOWN](#)

## Constructor Summary

public	<a href="#">PushPublishSessionRTP()</a>
--------	---

Methods inherited from class [com.wowza.wms.pushpublish.model.PushPublishSessionBase](#)

[getContextStr](#), [getProfile](#), [getPushPublishImpl](#), [getStream](#), [getType](#), [setProfile](#),  
[setPushPublishImpl](#), [setStream](#)

Methods inherited from class [java.lang.Object](#)

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#)

Methods inherited from interface [com.wowza.wms.pushpublish.model.IPushPublishSession](#)

[getContextStr](#), [getProfile](#), [getPushPublishImpl](#), [getStream](#), [getType](#), [setProfile](#),  
[setPushPublishImpl](#), [setStream](#)

## Constructors

### PushPublishSessionRTP

public [PushPublishSessionRTP\(\)](#)

---

Package

**com.wowza.wms.response**

## com.wowza.wms.response Class ResponseFunction

```
java.lang.Object
+-com.wowza.wms.response.ResponseFunction
```

```
public class ResponseFunction
extends Object
```

ResponseFunction: class for sending status messages to the client.

### Field Summary

public	<a href="#">doBreak</a>
--------	-------------------------

### Constructor Summary

public	<a href="#">ResponseFunction(IClient client)</a> Create an empty ResponseFunction
public	<a href="#">ResponseFunction(com.wowza.wms.netconnection.INetConnection netConnection)</a> Create an empty ResponseFunction
public	<a href="#">ResponseFunction(com.wowza.wms.netconnection.INetConnection netConnection, AMFObj amfObj)</a> Create an empty ResponseFunction
public	<a href="#">ResponseFunction(IClient client, AMFObj amfObj)</a> Create an empty ResponseFunction
public	<a href="#">ResponseFunction(IMediaStream stream, AMFObj amfObj)</a> Create an empty ResponseFunction

### Method Summary

void	<a href="#">addBody(AMFData body)</a> Add AMFData to the body of the message.
void	<a href="#">addBytes(byte[] inbytes)</a> Add raw AMFData bytes[] to message body
void	<a href="#">createBroadcastMessage(java.nio.ByteBuffer messageBytes)</a> Create broadcast message.
void	<a href="#">createConnectMessage(String inName, double inResultNum)</a> Create net connection connect message.
void	<a href="#">createDefaultMessage(String inName, double inResultNum)</a> Create a default message (onStatus type messages).
void	<a href="#">createEnhancedSeekMessage(byte[] messageBuffer, int src, int tc)</a>

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
--

### 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="#"><u>handleMessage</u></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(<u>RTPStream</u> stream)</a>
---------	--

### 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="#"><u>onRTPSessionCreate(RTPSession rtpSession)</u></a> Invoked when RTP session is created
void	<a href="#"><u>onRTPSessionDestroy(RTPSession rtpSession)</u></a> Invoked when RTP session is destroyed

### Methods

#### onRTPSessionCreate

public void **onRTPSessionCreate**([RTPSession rtpSession](#))

Invoked when RTP session is created

**Parameters:**

rtpSession - RTP session

#### onRTPSessionDestroy

public void **onRTPSessionDestroy**([RTPSession rtpSession](#))

Invoked when RTP session is destroyed

**Parameters:**

rtpSession - RTP session

## com.wowza.wms.rtp.model Interface IRTSPActionNotify

All Known Implementing Classes:

[PushPublishRTSPActionNotify](#)

public interface **IRTSPActionNotify**

extends

IRTSPActionNotify: listener interface for RTSP actions. See RTPSession.addActionListener(IRTSPActionNotify actionListener)

### Method Summary

void	<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	<code>onTeardown(RTPSession rtspSession, com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp)</code>
------	--

Invoked by TEARDOWN command

## Methods

### onDescribe

```
public void onDescribe(RTPSession rtspSession,  
                      com.wowza.wms.rtsp.RTSPRequestMessage req,  
                      com.wowza.wms.rtsp.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.rtsp.RTSPRequestMessage req,  
                      com.wowza.wms.rtsp.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.rtsp.RTSPRequestMessage req,  
                        com.wowza.wms.rtsp.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.rtsp.RTSPRequestMessage req,  
                        com.wowza.wms.rtsp.RTSPResponseMessages resp)
```

Invoked by GET\_PARAMETER command

**Parameters:**

- rtspSession - RTP session
- req - RTP request
- resp - RTP response

## onOptions

```
public void onOptions(RTPSession rtspSession,  
                      com.wowza.wms.rtsp.RTSPRequestMessage req,  
                      com.wowza.wms.rtsp.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.rtsp.RTSPRequestMessage req,  
                    com.wowza.wms.rtsp.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.rtsp.RTSPRequestMessage req,  
                   com.wowza.wms.rtsp.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.rtsp.RTSPRequestMessage req,  
                     com.wowza.wms.rtsp.RTSPResponseMessages resp)
```

Invoked by RECORD command

**Parameters:**

rtspSession  
req  
resp

---

## onRedirect

```
public void onRedirect(RTPSession rtspSession,  
                      com.wowza.wms.rtsp.RTSPRequestMessage req,  
                      com.wowza.wms.rtsp.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.rtsp.RTSPRequestMessage req,  
                    com.wowza.wms.rtsp.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.rtsp.RTSPRequestMessage req,  
                      com.wowza.wms.rtsp.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(IVHost vhost)</a>
	Constructor

### Method Summary

boolean	<a href="#">acquireSocketAddress(java.net.SocketAddress socketAddress)</a> Acquire UDP socket address
int	<a href="#">acquireUDPPortPair()</a> Acquire next available UDP port pair
int	<a href="#">acquireUDPPortPair(int port)</a> Acquire UDP port pair
void	<a href="#">cacheRTPStream(RTPStream stream)</a> Cache an RTP stream, Internal use.
void	<a href="#">doWatchdog()</a> Idle events for cleanup
boolean	<a href="#">existsRTSPTunnelingSession(String sessionId)</a> Return true if session id is valid RTSP/RTP tunneling session id
int[]	<a href="#">expandToPortPair(int port)</a> Expand a single port to a pair.
RTPPacketizerItem	<a href="#">getAudioPacketizerItem(IAplicationInstance appInstance, int codecId)</a> 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(RTPTrack rtpTrack)</a> 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>getRTSP TunnelingSession</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> (IAplicationInstance 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> (IAplicationInstance 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>putRTSP TunnelingSession</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>removeRTSP TunnelingSession</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> (RTPSession 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`

## 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.depacketizer.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="#"><u>setAudioHost</u></a> (String audioHost) Set audio host
void	<a href="#"><u>setAudioPort</u></a> (int audioPort) Set audio port
void	<a href="#"><u>setHost</u></a> (String host) Set host
void	<a href="#"><u>setHostType</u></a> (String hostType) Set host type (default IP4)
void	<a href="#"><u>setName</u></a> (String name) Set name
void	<a href="#"><u>setRTPWrapped</u></a> (boolean isRTPWrapped) Is stream wrapped in RTP (MPEG-TS in RTP)
void	<a href="#"><u>setStreamPort</u></a> (int streamPort) Set stream port
void	<a href="#"><u>setTTL</u></a> (int ttl) Set time to live
void	<a href="#"><u>setVideoHost</u></a> (String videoHost) Set video host
void	<a href="#"><u>setVideoPort</u></a> (int videoPort) Set video port
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

## 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

<code>public</code>	<code>RTPPort(String inIpAddress, int inPort, String outIpAddress, int outPort, boolean isMulticast)</code>
	Constructor

## Method Summary

<code>void</code>	<code><a href="#">bind(int direction)</a></code> Bind to port
<code>int</code>	<code><a href="#">getAddressCount()</a></code> Get address count
<code>IUDPMessageHandler</code>	<code><a href="#">getHandler()</a></code> Get the message handler
<code>String</code>	<code><a href="#">getInIpAddress()</a></code> Get in IP address
<code>int</code>	<code><a href="#">getInPort()</a></code> Get in port
<code>String</code>	<code><a href="#">getOutIpAddress()</a></code> Get out IP address
<code>int</code>	<code><a href="#">getOutPort()</a></code> Get out port
<code>RTPTrack</code>	<code><a href="#">getTrack()</a></code> Get RTP track
<code>int</code>	<code><a href="#">getTTL()</a></code> Get time to live (milliseconds)
<code>com.wowza.wms.rtp.transport.IUDPTransport</code>	<code><a href="#">getUDPTransport()</a></code> Get the UDP transport
<code>void</code>	<code><a href="#">handleMessage(java.net.SocketAddress socketAddr, Object message)</a></code>

boolean	<a href="#">isBlockUDPOut()</a>
boolean	<a href="#">isInMulticast()</a> Is in stream multicast
boolean	<a href="#">isMulticast()</a> Is multicast
boolean	<a href="#">isOutMulticast()</a> Is out stream multicast
void	<a href="#">sendMessage</a> (byte[] message, int offset, int len) Send a message out
void	<a href="#">sendResponse</a> (byte[] message) Send a response
void	<a href="#">sendResponse</a> (byte[] message, int offset, int len) Send response
void	<a href="#">sendResponse</a> (byte[] message, int offset, int len, java.net.SocketAddress destination) Send response to destination
void	<a href="#">sessionClosed</a> (com.wowza.wms.rtp.transport.IUDPTransportSession session)
void	<a href="#">sessionOpened</a> (com.wowza.wms.rtp.transport.IUDPTransportSession session)
void	<a href="#">setAddressCount</a> (int addressCount) Set address count
void	<a href="#">setBlockUDPOut</a> (boolean blockUDPOut)
void	<a href="#">setHandler</a> ( <a href="#">IRTPMessageHandler</a> handler) Set the message handler
void	<a href="#">setTrack</a> (RTPTrack track) Set RTP track
void	<a href="#">setTTL</a> (int ttl) Set time to live (milliseconds)
void	<a href="#">shutdown</a> () Shutdown port
void	<a href="#">unbind</a> () Unbind

**Methods inherited from class java.lang.Object**

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait`

**Methods inherited from interface com.wowza.wms.rtp.transport.IUDPMessagesHandler**

`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 IUDPTransport.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(RTPSession 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
--

### 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: 1
public static final	<a href="#">AUTHMETHOD_PUBLISH</a> Value: 2
public static final	<a href="#">AUTHMETHOD_UNKNOWN</a> Value: 0
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="#">ioPerformanceCounter</a>
protected	<a href="#">ioSession</a>
protected	<a href="#">ip</a>
protected	<a href="#">isAnnounce</a>
protected	<a href="#">isConnected</a>
protected	<a href="#">isDescribe</a>
protected	<a href="#">isSessionValid</a>
protected	<a href="#">lastAuthenticateMethod</a>
protected	<a href="#">lock</a>
protected	<a href="#">loggedConnect</a>
protected	<a href="#">properties</a>
protected	<a href="#">queryStr</a>
protected	<a href="#">redirectSession</a>
protected	<a href="#">redirectSessionCode</a>
protected	<a href="#">redirectSessionMessage</a>
protected	<a href="#">redirectSessionURL</a>
protected	<a href="#">referrer</a>
protected	<a href="#">rtpWriteListener</a>
protected	<a href="#">rtspPlayRangeStart</a>
protected	<a href="#">rtspPlayRangeStop</a>
protected	<a href="#">rtspTunnelingSessionId</a>
protected	<a href="#">serverIp</a>
protected	<a href="#">serverPort</a>
protected	<a href="#">sessionId</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(String sessionId)</a> Constructor
--------	---

## Method Summary

void	<a href="#">acceptSession()</a> Accept this session
void	<a href="#">addActionListener(IRTSPActionNotify actionListener)</a> Add action listener
void	<a href="#">addIOPerformance(IOPerformanceCounter totalIOPerformanceResult)</a> Internal use
void	<a href="#">addIOPerformance2(IOPerformanceCounter totalIOPerformanceResult)</a> Internal use
void	<a href="#">addRTSPStream(RTPStream stream)</a> 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="#"><code>getAudioPacketizerItem(RTPContext rtpContext, IApplicationInstance appInstance, int codecId)</code></a> Get audio packetizer for a given codec id.
IAuthenticateRTSP	<a href="#"><code>getAuthenticatePlayHandler()</code></a> Get the authentication play handler
IAuthenticateRTSP	<a href="#"><code>getAuthenticatePublishHandler()</code></a> Get the RTP authentication handler
ConnectionHolder	<a href="#"><code>getConnectionHolder()</code></a> Get connection holder, Internal use.
String	<a href="#"><code>getCookieStr()</code></a> Get cookie string
ElapsedTimer	<a href="#"><code>getElapsedTime()</code></a> Get the elapsed timer for this RTP session
int	<a href="#"><code>getIdleFrequency()</code></a> Get idle frequency (milliseconds)
RTPIdleHandler	<a href="#"><code>getIdleHandler()</code></a> Get idle handler
IOPerformanceCounter	<a href="#"><code>getIOPerformanceCounter()</code></a> Get IO performance counter
org.apache.mina.common.support.BaseIoSession	<a href="#"><code>getIoSession()</code></a>
String	<a href="#"><code>getIp()</code></a> Get remote IP address
int	<a href="#"><code>getLastAuthenticateMethod()</code></a> Get the last method received
WMSProperties	<a href="#"><code>getProperties()</code></a> Get properties
String	<a href="#"><code>getQueryStr()</code></a> Get query string
int	<a href="#"><code>getRedirectSessionCode()</code></a>
String	<a href="#"><code>getRedirectSessionMessage()</code></a>
String	<a href="#"><code>getRedirectSessionURL()</code></a>
String	<a href="#"><code>getReferrer()</code></a> Get referrer
RTPWriteListener	<a href="#"><code>getRTPWriteListener()</code></a> Get the RTP write listener for this session
double	<a href="#"><code>getRTSPPlayRangeStart()</code></a> Get play start range, Internal use.

double	<a href="#"><code>getRTSPPlayRangeStop()</code></a> Get play stop range, Internal use.
<a href="#"><u>RTPStream</u></a>	<a href="#"><code>getRTSPStream()</code></a> Get the default RTP Stream (all RTP sessions have a single RTP Stream)
<a href="#"><u>RTPStream</u></a>	<a href="#"><code>getRTSPStream(String streamId)</code></a> Get RTP Stream
String	<a href="#"><code>getRTSPTunnelingSessionId()</code></a> Get the RTSP/RTP tunneling session id
String	<a href="#"><code>getServerIp()</code></a> Get the server IP address
int	<a href="#"><code>getServerPort()</code></a> Get server port
String	<a href="#"><code>getSessionId()</code></a> Get session id
RTPPacketizerItem	<a href="#"><code>getStreamPacketizerItem(RTPContext rtpContext, IApplicationInstance appInstance, int codecId)</code></a> Get stream packetizer for a given codec id.
String	<a href="#"><code>getTimeRunning()</code></a> Get the time running for this RTP session
double	<a href="#"><code>getTimeRunningSeconds()</code></a> Get the number of second this RTP session has been running
String	<a href="#"><code>getUri()</code></a> Get URI
String	<a href="#"><code>getUserAgent()</code></a> Get user agent
<a href="#"><u>IVHost</u></a>	<a href="#"><code>getVHost()</code></a> Get vhost
RTPPacketizerItem	<a href="#"><code>getVideoPacketizerItem(RTPContext rtpContext, IApplicationInstance appInstance, int codecId)</code></a> Get video packetizer for a given codec id.
boolean	<a href="#"><code>isAnnounce()</code></a> Has ANNOUNCE command been called on this session
boolean	<a href="#"><code>isAnnounceOrDescribe()</code></a> Has announce or described been called
boolean	<a href="#"><code>isConnected()</code></a> Is session connection
boolean	<a href="#"><code>isDebugRTSPSession()</code></a> True if debugging RTSP session
boolean	<a href="#"><code>isDescribe()</code></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>(com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp) Invoked on ANNOUNCE command</a>
void	<a href="#"><u>onDescribe</u>(com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp) Invoked on DESCRIBE command</a>
void	<a href="#"><u>onGetParameter</u>(com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp) Invoked on GET_PARAMETER command</a>
void	<a href="#"><u>onOptions</u>(com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp) Invoked on OPTIONS command</a>
void	<a href="#"><u>onPause</u>(com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp) Invoked on PAUSE command</a>
void	<a href="#"><u>onPlay</u>(com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp) Invoked on PLAY command</a>
void	<a href="#"><u>onRecord</u>(com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp) Invoked on RECORD command</a>
void	<a href="#"><u>onRedirect</u>(com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp) Invoked on REDIRECT command</a>
void	<a href="#"><u>onSetParameter</u>(com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp) Invoked on SET_PARAMETER command</a>
void	<a href="#"><u>onSetup</u>(com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp) Invoked on SETUP command</a>
void	<a href="#"><u>onTeardown</u>(com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp) Invoked on TEARDOWN command</a>
void	<a href="#"><u>putAudioPacketizerItem</u>(int codecId, String classPath) Set the audio packetizer for a given codec id</a>
void	<a href="#"><u>putStreamPacketizerItem</u>(int codecId, String classPath) Set the stream packetizer for a given codec id</a>

void	<a href="#"><u>putVideoPacketizerItem(int codecId, String classPath)</u></a> Set the video packetizer for a given codec id
void	<a href="#"><u>redirectSession(String redirectSessionURL)</u></a>
void	<a href="#"><u>redirectSession(String redirectSessionURL, int redirectSessionCode)</u></a>
void	<a href="#"><u>rejectSession()</u></a> Reject this session
void	<a href="#"><u>removeActionListener(IRTSPActionNotify actionListener)</u></a> Remove action listener
RTPStream	<a href="#"><u>removeRTSPStream(String streamId)</u></a> Remove RTP stream
void	<a href="#"><u>setAnnounce(boolean isAnnounce)</u></a> Set ANNOUNCE command been called on this session
void	<a href="#"><u>setAppInstance(IApplicationInstance appInstance)</u></a> Set application instance
void	<a href="#"><u>setAuthenticatePlayHandler(IAuthenticateRTSP authenticatePlayHandler)</u></a> Set the authentication play handler
void	<a href="#"><u>setAuthenticatePublishHandler(IAuthenticateRTSP authenticatePublishHandler)</u></a> Set the RTP authentication handler
void	<a href="#"><u>setConnected(boolean isConnected)</u></a> Set session is connected
void	<a href="#"><u>setCookieStr(String cookieStr)</u></a> Set cooking string
void	<a href="#"><u>setDebugRTSPSession(boolean debugRTSPSession)</u></a> Set debugging RTSP session
void	<a href="#"><u>setDescribe(boolean isDescribe)</u></a> Set DESCRIBE command been called on this session
void	<a href="#"><u>setIdleFrequency(int idleFrequency)</u></a> Set idle frequency (milliseconds)
void	<a href="#"><u>setIdleHandler(RTPIdleHandler idleHandler)</u></a> Set idle handler
void	<a href="#"><u>setIOPerformanceCounter(IOPerformanceCounter ioPerformanceCounter)</u></a> Set IO performance counter
void	<a href="#"><u>setIoSession(org.apache.mina.common.support.BaseIoSession ioSession)</u></a>
void	<a href="#"><u>setIp(String ip)</u></a> Set remote IP address
void	<a href="#"><u>setLastAuthenticateMethod(int lastAuthenticateMethod)</u></a> 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 updateLoggingValues\(\)
    Update logging values, Internal use.
```

#### Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, 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.RTPIidleHandler 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(RTPStream stream)
```

Add RTP stream

**Parameters:**

stream - RTP stream

---

## removeRTSPStream

```
public RTPStream removeRTSPStream(String streamId)
```

Remove RTP stream

**Parameters:**

streamId - stream id

**Returns:**

RTP stream

---

## getRTSPStream

```
public RTPStream getRTSPStream(String streamId)
```

Get RTP Stream

**Parameters:**

streamId - stream id

**Returns:**

RTP Stream

---

## getRTSPStream

```
public RTPStream 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

---

## getRTSPPPlayRangeStart

```
public double getRTSPPPlayRangeStart()
```

Get play start range, Internal use.

**Returns:**

start range

---

## setRTSPPPlayRangeStart

```
public void setRTSPPPlayRangeStart(double rtspPlayRangeStart)
```

Set play start range, Internal use.

**Parameters:**

rtspPlayRangeStart - start range

---

## getRTSPPPlayRangeStop

```
public double getRTSPPPlayRangeStop()
```

Get play stop range, Internal use.

**Returns:**

stop range

---

## setRTSPPPlayRangeStop

```
public void setRTSPPPlayRangeStop(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(IVHost vhost)</a>
	Constructor

### Method Summary

<a href="#">RTPSession</a>	<a href="#">addSession(RTPSession session)</a> Add RTP session
void	<a href="#">addSessionListener(IRTPSessionNotify listener)</a> 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(String sessionId)</a> 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(IApplicationInstance appInstance, RTPSession rtpSession)</a> Notify session create
void	<a href="#">notifySessionCreate(RTPSession rtpSession)</a> Notify session create
void	<a href="#">notifySessionDestroy(IApplicationInstance appInstance, RTPSession rtpSession)</a> Notify session destroy
void	<a href="#">notifySessionDestroy(RTPSession rtpSession)</a> Notify session destroy
void	<a href="#">releaseSessionId(String sessionIdStr)</a> Release an RTP session id

<u>RTPSession</u>	<u>removeSession</u> ( <u>RTPSession</u> session) Remove RTP session by object
<u>RTPSession</u>	<u>removeSession</u> (String sessionId) Remove RTP session by session id
void	<u>removeSessionListener</u> ( <u>IRTPSessionNotify</u> listener) Remove an RTP session listener

**Methods inherited from class java.lang.Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, 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(IAplicationInstance appInstance,  
                               RTPSession rtpSession)
```

Notify session create

**Parameters:**

appInstance - application instance

rtpSession - RTP session

---

## notifySessionDestroy

```
public void notifySessionDestroy(IAplicationInstance 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="#">DEFAULT_RTPIGNOREPROFILELEVELID</a>
	Value: <b>false</b>
public static final	<a href="#">DEFAULT_RTPIGNORESPROPPARAMETERSETS</a>
	Value: <b>false</b>
public static final	<a href="#">DEFAULT_RTSPSESSIONNAME</a>
	Value:
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="#"><u>SDPLOCATION_STREAM</u></a> Value: <b>0</b>
public static final	<a href="#"><u>SDPLOCATION_VIDEO</u></a> Value: <b>2</b>
public static final	<a href="#"><u>STREAMINFO_SESSIONATTRIBUTES</u></a> Value: <b>attributes</b>
public static final	<a href="#"><u>STREAMINFO_SESSIONBANDWIDTH</u></a> Value: <b>bandwidth</b>
public static final	<a href="#"><u>STREAMINFO_SESSIONCONNECTIONDATA</u></a> Value: <b>connectiondata</b>
public static final	<a href="#"><u>STREAMINFO_SESSIONEMAILADDRESS</u></a> Value: <b>emailaddress</b>
public static final	<a href="#"><u>STREAMINFO_SESSIONINFORMATION</u></a> Value: <b>information</b>
public static final	<a href="#"><u>STREAMINFO_SESSIONNAME</u></a> Value: <b>name</b>
public static final	<a href="#"><u>STREAMINFO_SESSIONPHONENUMBER</u></a> Value: <b>phonenumber</b>
public static final	<a href="#"><u>STREAMINFO_SESSIONPORIGIN</u></a> Value: <b>origin</b>
public static final	<a href="#"><u>STREAMINFO_SESSIONPROTOCOLVERSION</u></a> Value: <b>protocolversion</b>
public static final	<a href="#"><u>STREAMINFO_SESSIONREPEATTIMES</u></a> Value: <b>repeattimes</b>
public static final	<a href="#"><u>STREAMINFO_SESSIONTIMEZONES</u></a> Value: <b>timezones</b>
public static final	<a href="#"><u>STREAMINFO_SESSIONTIMING</u></a> Value: <b>timing</b>
public static final	<a href="#"><u>STREAMINFO_SESSIONURI</u></a> Value: <b>uri</b>

## Constructor Summary

public	<a href="#"><u>RTPStream(RTPContext context, String streamId)</u></a> Constructor
--------	--

## Method Summary

void	<a href="#"><u>addStreamAttribute</u></a> (String key, String value) Add a name value pair to the stream attribute collection
void	<a href="#"><u>addStreamInfo</u></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> (RTSPSession rtspSession, int isStreamPacketizer, RTPRequestStatus status) Execute describe command
String	<a href="#"><u>describe</u></a> (RTSPSession 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>getAppName</u></a> () Get the application instance name

String	<a href="#"><code>getAppName()</code></a> Get the application name
RTPTrack	<a href="#"><code>getAudioTrack()</code></a> Get the most likely audio track
int	<a href="#"><code>getAutoAllocateInterleavePorts()</code></a>
int	<a href="#"><code>getAVSyncMethod()</code></a> Get the audio/video sync method.
double	<a href="#"><code>getDuration()</code></a> Get the duration of the stream if video on demand
String	<a href="#"><code>getHost()</code></a> Get host
<a href="#"><code>AMFPacket[]</code></a>	<a href="#"><code>getLastPacketsByType(IMediaReader localReader, double startTime)</code></a> Analyzes stream to get information, Internal use.
<a href="#"><code>AMFPacket[]</code></a>	<a href="#"><code>getLastPacketsByType(IMediaStream localStream)</code></a> Analyzes stream to get information, Internal use.
int	<a href="#"><code>getMaxRTCPWaitTime()</code></a> Get max time to wait for RTCP sender reports (milliseconds)
String	<a href="#"><code>getMediaCasterType()</code></a> Get the media caster stream type for this stream
<a href="#"><code>IMediaReader</code></a>	<a href="#"><code>getMediaReader()</code></a> Get media reader if video on demand stream
<code>byte[]</code>	<a href="#"><code>getMetadataPacket()</code></a> Get the onMetadata packet for this RTP stream
int	<a href="#"><code>getMode()</code></a> Get the current play/publish mode
int	<a href="#"><code>getMPEGTSAudioBitrate()</code></a>
String	<a href="#"><code>getMPEGTSAudioLanguage()</code></a>
int	<a href="#"><code>getMPEGTSAudioPID()</code></a> Get the audio PID id if MPEG-TS stream
int	<a href="#"><code>getMPEGTSProgramID()</code></a>
int	<a href="#"><code>getMPEGTSVideoBitrate()</code></a>
int	<a href="#"><code>getMPEGTSVideoPID()</code></a> Get the video PID id if MPEG-TS stream
long	<a href="#"><code>getNormalizedNTPTimecode(long timecode)</code></a> Turn a millisecond timcode into an NTP timecode
String	<a href="#"><code>getOutHost()</code></a> Get the out host

<a href="#">RTPContext</a>	<a href="#"><code>getRTPContext()</code></a> Get the RTP context
<a href="#">RTPDestination</a>	<a href="#"><code>getRTPDestination()</code></a> Get RTP destination
RTPStream.RTPInfo	<a href="#"><code>getRTPInfo(double startTime, int videoSeq, int audioSeq)</code></a> Get the RTP info
RTPStreamContext	<a href="#"><code>getRTPStreamContext()</code></a>
String	<a href="#"><code>getRTSPBindIpAddress()</code></a> Get the bind RTSP bind IP address
String	<a href="#"><code>getRTSPConnectionAddressType()</code></a> Get the connection address type
String	<a href="#"><code>getRTSPConnectionIpAddress()</code></a> Get the connection IP address
int	<a href="#"><code>getRTSPMaximumPendingWriteBytes()</code></a> Get the maximum number of waiting bytes allow for this RTSP session
String	<a href="#"><code>getRTSPOriginAddressType()</code></a> Get the origin address type
String	<a href="#"><code>getRTSPOriginIpAddress()</code></a> Get the origin IP address
String	<a href="#"><code>getRTSPSessionDescription()</code></a> Get session description
java.util.List	<a href="#"><code>getRTSPSessionExtraLines()</code></a> Get extra SDP lines
String	<a href="#"><code>getRTSPSessionName()</code></a> Get session name
int	<a href="#"><code>getRTSPSessionTimeout()</code></a> Get RTP session timeout (milliseconds)
String	<a href="#"><code>getSDPLang()</code></a> Get the SDP language
<a href="#">RTPSession</a>	<a href="#"><code>getSession()</code></a> Get the RTP session
<a href="#">IMediaStream</a>	<a href="#"><code>getStream()</code></a> Get the IMediaStream
java.util.Map	<a href="#"><code>getStreamAttributes()</code></a> Get all name/value pairs in the stream attributes collection
String	<a href="#"><code>getStreamExt()</code></a> Get the stream extension
String	<a href="#"><code>getStreamId()</code></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(String trackId)</u></a> Get track by id
String	<a href="#"><u>getTrackID(String seq)</u></a> 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.IUDPTTransport	<a href="#"><u>getUDPTransport(boolean isMulticast)</u></a> 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="#"><code>getVODLastTimeTC()</code></a> Get the last timecode (milliseconds) sent for video on demand
long	<a href="#"><code>getVODPlayLen()</code></a> Get the video on demand play duration (milliseconds)
long	<a href="#"><code>getVODStartTimeTC()</code></a> Get the video on demand start time (milliseconds)
void	<a href="#"><code>idle(org.apache.mina.common.IoSession session, RtmpResponseMessage resp)</code></a> Process idle event
void	<a href="#"><code>incrementMediaInBytes(long bytes)</code></a> Increment the media bytes in, Internal use.
void	<a href="#"><code>initProperties(IApplicationInstance appInstance)</code></a>
boolean	<a href="#"><code>isAVSyncNonSR()</code></a> Is sync method based on RTCP packets (sender report)
boolean	<a href="#"><code>isBlockUDPOut()</code></a>
boolean	<a href="#"><code>isCheckIpAddr()</code></a> Are we checking the ip address of each incoming RTP packet
boolean	<a href="#"><code>isCheckSSRC()</code></a> Are we checking the ssrc values of each incoming RTP packet
boolean	<a href="#"><code>isForceMPEGTSOut()</code></a>
boolean	<a href="#"><code>isForceRTSPInterleaved()</code></a> True if forcing RTSP interleaved
boolean	<a href="#"><code>isLive()</code></a> Is live stream
boolean	<a href="#"><code>isModePlay()</code></a> Is this a play stream
boolean	<a href="#"><code>isModePublish()</code></a> Is this a publish stream
boolean	<a href="#"><code>isModeUnknown()</code></a> Is the stream mode unknown (publish vs play)
boolean	<a href="#"><code>isMPEGTSOut()</code></a> Is MPEG-TS out
boolean	<a href="#"><code>isPaused()</code></a> Is stream paused
boolean	<a href="#"><code>isPublishStreamReady()</code></a> See if a publishing stream has enough data to start playback
boolean	<a href="#"><code>isRefuseIncomingUDPConnections()</code></a>

boolean	<a href="#"><u>isRefuseOutgoingUDPConnections()</u></a>
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>isRTPIgnoreSPropParameterSets()</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(long currTime, int timeout)</u></a> Is the stream timeout out
void	<a href="#"><u>lockRepeaterStreams(java.util.List streamNames)</u></a> Lock a list of live repeater stream names, Internal use.
void	<a href="#"><u>pause(RTPRequestStatus status)</u></a> Execute pause
RTPStreamPlayResult	<a href="#"><u>play(RTPRequestStatus status)</u></a> Execute play
RTPStreamPlayResult	<a href="#"><u>play(RTPRequestStatus status, double startTime, double stopTime)</u></a> Execute play
void	<a href="#"><u>putRTSPSessionExtraLine(int location, String line)</u></a> Add an extra line to the SDP data
RTPStreamPlayResult	<a href="#"><u>record(RTPRequestStatus status)</u></a> Execute record
RTPStreamPlayResult	<a href="#"><u>record(RTPRequestStatus status, double startTime, double stopTime)</u></a> Execute record
RTPTrack	<a href="#"><u>removeTrack(String trackId)</u></a> Remove a track by id
RTPTrack	<a href="#"><u>removeTrackInternal(String trackId)</u></a> Remove track by track id

void	<a href="#"><u>resetSentMetadataFlag()</u></a> Reset sendMetadata flag
void	<a href="#"><u>setAppInstanceName</u>(String appInstanceName)</a> Set the application instance name
void	<a href="#"><u>setAppName</u>(String appName)</a> Get the application name
void	<a href="#"><u>setAVSyncMethod</u>(int avSyncMethod)</a> Set the audio/video sync method.
void	<a href="#"><u>setBlockUDPOut</u>(boolean blockUDPOut)</a>
void	<a href="#"><u>setCheckIpAddr</u>(boolean checkIpAddr)</a> Are we checking the ip address of each incoming RTP packet
void	<a href="#"><u>setCheckSSRC</u>(boolean checkSSRC)</a> Are we checking the ssrc values of each incoming RTP packet
void	<a href="#"><u>setForceMPEGTSOut</u>(boolean isForceMPEGTSOut)</a>
void	<a href="#"><u>setForceRTSPInterleaved</u>(boolean isForceRTSPInterleaved)</a> True if forcing RTSP interleaved
void	<a href="#"><u>setHost</u>(String host)</a> Set host
void	<a href="#"><u>setLive</u>(boolean isLive)</a> Is live stream
void	<a href="#"><u>setMaxRTCPWaitTime</u>(int maxRTCPWaitTime)</a> Set max time to wait for RTCP sender reports (milliseconds)
void	<a href="#"><u>setMode</u>(int mode)</a> Set the play/publish mode
void	<a href="#"><u>setMPEGTSAudioBitrate</u>(int mpegtsAudioBitrate)</a>
void	<a href="#"><u>setMPEGTSAudioLanguage</u>(String mpegtsAudioLanguage)</a>
void	<a href="#"><u>setMPEGTSAudioPID</u>(int mpegtsAudioPID)</a> Set the audio PID id if MPEG-TS stream
void	<a href="#"><u>setMPEGTSOut</u>(boolean isMPEGTSOut)</a> Is MPEG-TS out
void	<a href="#"><u>setMPEGTSProgramID</u>(int mpegtsProgramId)</a>
void	<a href="#"><u>setMPEGTSVideoBitrate</u>(int mpegtsVideoBitrate)</a>
void	<a href="#"><u>setMPEGTSVideoPID</u>(int mpegtsVideoPID)</a> Set the video PID id if MPEG-TS stream
void	<a href="#"><u>setOutHost</u>(String outHost)</a> Set out host

void	<a href="#"><u>setRefuseIncomingUDPConnections(boolean refuseIncomingUDPConnections)</u></a>
void	<a href="#"><u>setRefuseOutgoingUDPConnections(boolean refuseOutgoingUDPConnections)</u></a>
void	<a href="#"><u>setResetPlayStream(boolean doResetPlayStream)</u></a> Set reset stream trigger
void	<a href="#"><u>setResyncAudioVideoOnSR(boolean resyncAudioVideoOnSR)</u></a> Reset audio/video sync on new RTCP packets (not just first packet)
void	<a href="#"><u>setRTPDestination(RTPDestination rtpDestination)</u></a> Set RTP destination
void	<a href="#"><u>setRTPIgnoreProfileLevelId(boolean rtpIgnoreProfileLevelId)</u></a>
void	<a href="#"><u>setRTPIgnoreSPropParameterSets(boolean rtpIgnoreSPropParameterSets)</u></a>
void	<a href="#"><u>setRTPStreamContext(RTPStreamContext rtpStreamContext)</u></a>
void	<a href="#"><u>setRTSP(boolean isRTSP)</u></a> Is this RTP Stream managed by RTSP session
void	<a href="#"><u>setRTSPAlwaysUseSDPPorts(boolean rtspAlwaysUseSDPPorts)</u></a> Force RTSP to use ports in SDP data
void	<a href="#"><u>setRTSPBindIpAddress(String rtspBindIpAddress)</u></a> Set the bind RTSP bind IP address
void	<a href="#"><u>setRTSPConnectionAddressType(String rtspConnectionAddressType)</u></a> Set the connection address type
void	<a href="#"><u>setRTSPConnectionIpAddress(String rtspConnectionIpAddress)</u></a> Set the connection IP address
void	<a href="#"><u>setRTSPMaximumPendingWriteBytes(int rtspMaximumPendingWriteBytes)</u></a> Set the maximum number of waiting bytes allow for this RTSP session
void	<a href="#"><u>setRTSPOriginAddressType(String rtspOriginAddressType)</u></a> Set the origin address type
void	<a href="#"><u>setRTSPOriginIpAddress(String rtspOriginIpAddress)</u></a> Set the origin IP address
void	<a href="#"><u>setRTSPPull(boolean isRTSPPull)</u></a> Is this RTP Stream managed by RTSP session
void	<a href="#"><u>setRTSPSessionDescription(String rtspSessionDescription)</u></a> Get session description
void	<a href="#"><u>setRTSPSessionName(String rtspSessionName)</u></a> Set session name
void	<a href="#"><u>setRTSPSessionTimeout(int rtspSessionTimeout)</u></a> Set RTP session timeout (milliseconds)
void	<a href="#"><u>setSDPLang(String sdpLang)</u></a> Set the SDP language

void	<a href="#"><u>setSendSDEEvents(boolean sendSDEEvents)</u></a> Send RTCP SDES events
void	<a href="#"><u>setSession(RTPSession session)</u></a> Set the RTP session
void	<a href="#"><u>setStreamExt(String streamExt)</u></a> Set the stream extension
void	<a href="#"><u>setStreamName(String streamName)</u></a> Set stream name
void	<a href="#"><u>setStreamNameLogging(String streamNameLogging)</u></a> Set the stream name used for logging
void	<a href="#"><u>setStreamQueryStr(String streamQueryStr)</u></a> Set the stream query string
void	<a href="#"><u>setStreamSessionId(long streamSessionId)</u></a> Set the stream session id
void	<a href="#"><u>setStreamSessionIp(String streamSessionIp)</u></a> Set the stream session ip
void	<a href="#"><u>setStreamSessionVersion(long streamSessionVersion)</u></a> Set the stream session version
void	<a href="#"><u>setStreamType(String streamType)</u></a> Set the stream type
void	<a href="#"><u>setTransportMode(String transportMode)</u></a> Set the transport mode
void	<a href="#"><u>setUDPManagedDeliveryCount(int udpManagedDeliveryCount)</u></a>
void	<a href="#"><u>setUDPManagedDeliveryDelay(int udpManagedDeliveryDelay)</u></a>
void	<a href="#"><u>setVODLastTimeTC(long vodLastTimeTC)</u></a> Set the last timecode (milliseconds) sent for video on demand
void	<a href="#"><u>setVODPlayLen(long vodPlayLen)</u></a> Set the video on demand play duration (milliseconds)
void	<a href="#"><u>setVODStartTimeTC(long vodStartTimeTC)</u></a> Set the video on demand start time (milliseconds)
void	<a href="#"><u>shutdown(RTPRequestStatus status)</u></a> shutdown RTP stream, Internal use.
RTPTrack	<a href="#"><u>sloppyGetTrack(String trackID)</u></a> 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(String transport)</u></a> 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`

**Fields****SDPLOCATION\_STREAM**

`public static final int SDPLOCATION_STREAM`

Constant value: **0**

**SDPLOCATION\_AUDIO**

`public static final int SDPLOCATION_AUDIO`

Constant value: **1**

**SDPLOCATION\_VIDEO**

`public static final int SDPLOCATION_VIDEO`

Constant value: **2**

**MODE\_UNKNOWN**

`public static final int MODE_UNKNOWN`

Constant value: **0**

**MODE\_PLAY**

`public static final int MODE_PLAY`

Constant value: **1**

**MODE\_PUBLISH**

`public static final int MODE_PUBLISH`

Constant value: **2**

## AVSYNCMETHODS\_UNKNOWN

```
public static final int AVSYNCMETHODS_UNKNOWN
```

Constant value: 0

---

## AVSYNCMETHODS\_SENDERREPORT

```
public static final int AVSYNCMETHODS_SENDERREPORT
```

Constant value: 1

---

## AVSYNCMETHODS\_SYSTEMCLOCK

```
public static final int AVSYNCMETHODS_SYSTEMCLOCK
```

Constant value: 2

---

## AVSYNCMETHODS\_RTPTIMECODE

```
public static final int AVSYNCMETHODS_RTPTIMECODE
```

Constant value: 3

---

## STREAMINFO\_SESSIONPROTOCOLVERSION

```
public static final java.lang.String STREAMINFO_SESSIONPROTOCOLVERSION
```

Constant value: **protocolversion**

---

## STREAMINFO\_SESSIONPORIGIN

```
public static final java.lang.String STREAMINFO_SESSIONPORIGIN
```

Constant value: **origin**

---

## STREAMINFO\_SESSIONNAME

```
public static final java.lang.String STREAMINFO_SESSIONNAME
```

Constant value: **name**

---

## STREAMINFO\_SESSIONINFORMATION

```
public static final java.lang.String STREAMINFO_SESSIONINFORMATION
```

Constant value: **information**

---

(continued from last page)

## STREAMINFO\_SESSIONURI

```
public static final java.lang.String STREAMINFO_SESSIONURI
```

Constant value: **uri**

## STREAMINFO\_SESSIONEMAILADDRESS

```
public static final java.lang.String STREAMINFO_SESSIONEMAILADDRESS
```

Constant value: **emailaddress**

## STREAMINFO\_SESSIONPHONENUMBER

```
public static final java.lang.String STREAMINFO_SESSIONPHONENUMBER
```

Constant value: **phonenumbers**

## STREAMINFO\_SESSIONCONNECTIONDATA

```
public static final java.lang.String STREAMINFO_SESSIONCONNECTIONDATA
```

Constant value: **connectiondata**

## STREAMINFO\_SESSIONBANDWIDTH

```
public static final java.lang.String STREAMINFO_SESSIONBANDWIDTH
```

Constant value: **bandwidth**

## STREAMINFO\_SESSIONTIMING

```
public static final java.lang.String STREAMINFO_SESSIONTIMING
```

Constant value: **timing**

## STREAMINFO\_SESSIONREPEATTIMES

```
public static final java.lang.String STREAMINFO_SESSIONREPEATTIMES
```

Constant value: **repeattimes**

## STREAMINFO\_SESSIONTIMEZONES

```
public static final java.lang.String STREAMINFO_SESSIONTIMEZONES
```

Constant value: **timezones**

## STREAMINFO\_SESSIONATTRIBUTES

```
public static final java.lang.String STREAMINFO_SESSIONATTRIBUTES
```

(continued from last page)

Constant value: **attributes**

---

## DEFAULT\_RTSPSESSIONNAME

```
public static final java.lang.String DEFAULT_RTSPSESSIONNAME
```

Constant value:

---

## DEFAULT\_RTPIGNOREPROFILELEVELID

```
public static final boolean DEFAULT_RTPIGNOREPROFILELEVELID
```

Constant value: **false**

---

## DEFAULT\_RTPIGNORERESPROPARAMETERSETS

```
public static final boolean DEFAULT_RTPIGNORERESPROPARAMETERSETS
```

Constant value: **false**

## Constructors

### RTPStream

```
public RTPStream(RTPContext context,  
String streamId)
```

Constructor

**Parameters:**

context - RTP context  
streamId - stream id

## Methods

### getRTPContext

```
public RTPContext getRTPContext()
```

Get the RTP context

**Returns:**

RTP context

---

### getStreamLock

```
public Object getStreamLock()
```

Get the synchronization lock for this stream

**Returns:**

synchronization lock for this stream

(continued from last page)

## **addTrack**

```
public void addTrack(RTPTrack track)
```

Add a track

**Parameters:**

track - RTP track

---

## **removeTrack**

```
public RTPTrack removeTrack(String trackId)
```

Remove a track by id

**Parameters:**

trackId - track id

**Returns:**

RTP track

---

## **addTrackInternal**

```
public void addTrackInternal(RTPTrack track)
```

Add RTP track

**Parameters:**

track - RTP track

---

## **removeTrackInternal**

```
public RTPTrack removeTrackInternal(String trackId)
```

Remove track by track id

**Parameters:**

trackId - track id

**Returns:**

RTP track that was removed

---

## **clearTracks**

```
public void clearTracks()
```

Clear all tracks

---

## **incrementMediaInBytes**

```
public void incrementMediaInBytes(long bytes)
```

Increment the media bytes in, Internal use.

**Parameters:**

bytes

(continued from last page)

## sloppyGetTrack

```
public RTPTrack sloppyGetTrack(String trackId)
```

Sloppy method for finding track by name, Internal use.

**Parameters:**

trackId - track name

**Returns:**

RTP track

---

## getTrack

```
public RTPTrack getTrack(String trackId)
```

Get track by id

**Parameters:**

trackId - track id

**Returns:**

RTP track

---

## addTrackId

```
public void addTrackId(String seq,  
                      String trackId)
```

Add track

**Parameters:**

seq - sequence number  
trackId - track id

---

## getTrackId

```
public String getTrackId(String seq)
```

Get track by sequence number

**Parameters:**

seq - sequence number

**Returns:**

RTP track

---

## getTrackNames

```
public java.util.List getTrackNames()
```

Get a list of track ids

**Returns:**

list of track ids

(continued from last page)

## getSession

```
public RTPSession getSession()
```

Get the RTP session

**Returns:**

RTP session

---

## setSession

```
public void setSession(RTPSession session)
```

Set the RTP session

**Parameters:**

session - RTP session

---

## getStreamId

```
public String getStreamId()
```

Get the stream id

**Returns:**

stream id

---

## getStream

```
public IMediaStream getStream()
```

Get the IMediaStream

**Returns:**

stream (IMediaStream) interface

---

## getAudioTrack

```
public RTPTrack getAudioTrack()
```

Get the most likely audio track

**Returns:**

RTP track

---

## getVideoTrack

```
public RTPTrack getVideoTrack()
```

Get the most likely video track

**Returns:**

RTP track

---

## getStreamTrack

```
public RTPTrack getStreamTrack()
```

(continued from last page)

Get the most likely stream track

**Returns:**

RTP track

---

**addStreamInfo**

```
public void addStreamInfo(String key,  
                           String value)
```

Add a name value pair to the stream info collection

**Parameters:**

key - name  
value - value

---

**getStreamInfo**

```
public String getStreamInfo(String key)
```

Get stream info by name

**Parameters:**

key - name

**Returns:**

value

---

**addStreamAttribute**

```
public void addStreamAttribute(String key,  
                               String value)
```

Add a name value pair to the stream attribute collection

**Parameters:**

key - name  
value - value

---

**getVHost**

```
public IVHost getVHost()
```

Get vhost

**Returns:**

vhost

---

**getHost**

```
public String getHost()
```

Get host

**Returns:**

host

(continued from last page)

## setHost

```
public void setHost(String host)
```

Set host

**Parameters:**

host - host

---

## streamExists

```
public boolean streamExists()
```

Return true if RTP stream contains a IMediaStream

**Returns:**

true if RTP stream contains a IMediaStream

---

## isPublishStreamReady

```
public boolean isPublishStreamReady()
```

See if a publishing stream has enough data to start playback

**Returns:**

true, if the stream is ready for playback

---

## lockRepeaterStreams

```
public void lockRepeaterStreams(java.util.List streamNames)
```

Lock a list of live repeater stream names, Internal use.

**Parameters:**

streamNames - stream names

---

## unlockRepeaterStreams

```
public void unlockRepeaterStreams()
```

Unlock live repeater streams

---

## getMediaCasterType

```
public String getMediaCasterType()
```

Get the media caster stream type for this stream

**Returns:**

media caster stream type

---

## createStream

```
public boolean createStream(RTPRequestStatus status)
```

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`

---

**initProperties**`public void initProperties(IApplicationInstance appInstance)`

---

**getAppInstance**`public IApplicationInstance getAppInstance()`

Get application instance

**Returns:**

application instance

---

**isPaused**`public boolean isPaused()`

Is stream paused

**Returns:**

true if stream is paused

---

**getDuration**`public double getDuration()`

Get the duration of the stream if video on demand

**Returns:**

duration in seconds

---

**isResetPlayStream**`public boolean isResetPlayStream()`

Is reset stream trigger, Internal use.

**Returns:**

true if reset trigger

(continued from last page)

## 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

(continued from last page)

startTime - playback start time  
stopTime - playback stop time

**Returns:**

status

---

**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:**

(continued from last page)

`resyncAudioVideoOnSR - true if resetting audio/video on new RTCP packets`

---

## getMetadataPacket

```
public byte[] getMetadataPacket()
```

Get the onMetadata packet for this RTP stream

**Returns:**

onMetadata packet

---

## resetSentMetadataFlag

```
public void resetSentMetadataFlag()
```

Reset sendMetadata flag

---

## checkSendMetadata

```
public void checkSendMetadata(long adjTimecode,  
                           RTPTrack rtpTrack)
```

Check to see if we have sent onMetadata event, if not send

**Parameters:**

adjTimecode - timecode (milliseconds)

---

## extractCodecConfigFromTrackInfo

```
public void extractCodecConfigFromTrackInfo()
```

Extract codec config information from SDP data

---

## getStreamType

```
public String getStreamType()
```

Get the stream type

**Returns:**

stream type

---

## setStreamType

```
public void setStreamType(String streamType)
```

Set the stream type

**Parameters:**

streamType - stream type

---

## getAppName

```
public String getAppName()
```

Get the application name

**Returns:**

(continued from last page)

---

application name

---

## setAppName

```
public void setAppName(String appName)
```

Get the application name

**Parameters:**

appName - application name

---

## getAppInstanceName

```
public String getAppInstanceName()
```

Get the application instance name

**Returns:**

application instance name

---

## setAppInstanceName

```
public void setAppInstanceName(String appInstanceName)
```

Set the application instance name

**Parameters:**

appInstanceName - application instance name

---

## getStreamExt

```
public String getStreamExt()
```

Get the stream extension

**Returns:**

stream extension

---

## setStreamExt

```
public void setStreamExt(String streamExt)
```

Set the stream extension

**Parameters:**

streamExt - stream extension

---

## 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

---

## 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)
```

(continued from last page)

Set the stream session version

**Parameters:**

streamSessionVersion - stream session version

---

**getStreamSessionIp**

```
public String getStreamSessionIp()
```

Get the stream session ip

**Returns:**

stream session ip

---

**setStreamSessionIp**

```
public void setStreamSessionIp(String streamSessionIp)
```

Set the stream session ip

**Parameters:**

streamSessionIp - stream session ip

---

**isCheckSSRC**

```
public boolean isCheckSSRC()
```

Are we checking the ssrc values of each incoming RTP packet

**Returns:**

true if checking

---

**setCheckSSRC**

```
public void setCheckSSRC(boolean checkSSRC)
```

Are we checking the ssrc values of each incoming RTP packet

**Parameters:**

checkSSRC - true if checking

---

**isCheckIpAddr**

```
public boolean isCheckIpAddr()
```

Are we checking the ip address of each incoming RTP packet

**Returns:**

true if checking ip address of each incoming RTP packet

---

**setCheckIpAddr**

```
public void setCheckIpAddr(boolean checkIpAddr)
```

Are we checking the ip address of each incoming RTP packet

**Parameters:**

checkIpAddr - true if checking ip address of each incoming RTP packet

## getUDPTransport

```
public com.wowza.wms.rtp.transport.IUDPTransport getUDPTransport(boolean isMulticast)
```

Get the UDP transport for this stream

**Parameters:**

isMulticast - true if multicast

**Returns:**

UDP transport

---

## getAVSyncMethod

```
public int getAVSyncMethod()
```

Get the audio/video sync method. See AVSYNCMETHODS\_\*

**Returns:**

audio/video sync method. See AVSYNCMETHODS\_\*

---

## setAVSyncMethod

```
public void setAVSyncMethod(int avSyncMethod)
```

Set the audio/video sync method. See AVSYNCMETHODS\_\*

**Parameters:**

avSyncMethod - audio/video sync method. See AVSYNCMETHODS\_\*

---

## isAVSyncNonSR

```
public boolean isAVSyncNonSR()
```

Is sync method based on RTCP packets (sender report)

**Returns:**

true if based in RTCP sender reports

---

## getMaxRTCPWaitTime

```
public int getMaxRTCPWaitTime()
```

Get max time to wait for RTCP sender reports (milliseconds)

**Returns:**

max time to wait for RTCP sender reports (milliseconds)

---

## setMaxRTCPWaitTime

```
public void setMaxRTCPWaitTime(int maxRTCPWaitTime)
```

Set max time to wait for RTCP sender reports (milliseconds)

**Parameters:**

maxRTCPWaitTime - max time to wait for RTCP sender reports (milliseconds)

---

**getStreamInfo**

```
public java.util.Map getStreamInfo()
```

Get all name/value pairs in the stream info collection

**Returns:**

map of name/value pairs

---

**getStreamAttributes**

```
public java.util.Map getStreamAttributes()
```

Get all name/value pairs in the stream attributes collection

**Returns:**

map of name/value pairs

---

**avSyncNameToId**

```
public static int avSyncNameToId(String avSyncName)
```

Get audio/video sync id from name

**Parameters:**

avSyncName - audio/video sync name

**Returns:**

audio/video sync id, see AVSYNCMETHODS\_\*

---

**getMPEGTSAudioLanguage**

```
public String getMPEGTSAudioLanguage()
```

---

**setMPEGTSAudioLanguage**

```
public void setMPEGTSAudioLanguage(String mpegtsAudioLanguage)
```

---

**getMPEGTSPID**

```
public int getMPEGTSPID()
```

---

**setMPEGTSPID**

```
public void setMPEGTSPID(int mpegtsPid)
```

---

**getMPEGTSPVideoPID**

```
public int getMPEGTSPVideoPID()
```

---

(continued from last page)

Get the video PID id if MPEG-TS stream

**Returns:**

video PID id if MPEG-TS stream

---

**setMPEGTSVideoPID**

```
public void setMPEGTSVideoPID(int mpegtsvideopID)
```

Set the video PID id if MPEG-TS stream

**Parameters:**

mpegtsvideopID - video PID id if MPEG-TS stream

---

**getMPEGTSAudioPID**

```
public int getMPEGTSAudioPID()
```

Get the audio PID id if MPEG-TS stream

**Returns:**

audio PID id if MPEG-TS stream

---

**setMPEGTSAudioPID**

```
public void setMPEGTSAudioPID(int mpegtsaudioPID)
```

Set the audio PID id if MPEG-TS stream

**Parameters:**

mpegtsaudioPID - audio PID id if MPEG-TS stream

---

**isRTSP**

```
public boolean isRTSP()
```

Is this RTP Stream managed by RTSP session

**Returns:**

true if managed by RTSP session

---

**setRTSP**

```
public void setRTSP(boolean isRTSP)
```

Is this RTP Stream managed by RTSP session

**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()
```

(continued from last page)

Get the out host

**Returns:**

out host

---

**setOutHost**

```
public void setOutHost(String outHost)
```

Set out host

**Parameters:**

outHost - out host

---

**idle**

```
public void idle(org.apache.mina.common.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(RTSPSession rtspSession,  
RTPRequestStatus status)
```

Execute describe command

(continued from last page)

**Parameters:**

rtspSession - RTP session  
status - RTP status

**Returns:**

describe response

---

**switchSetupToMPEGTS**

```
public void switchSetupToMPEGTS()
```

Switch a stream to MPEG-TS, Internal use.

---

**describe**

```
public String describe(RTPSession rtspSession,  
                      int isStreamPacketizer,  
                      RTPRequestStatus status)
```

Execute describe command

**Parameters:**

rtspSession - RTP session  
isStreamPacketizer - true is stream  
status - RTP status

**Returns:**

describe string

---

**getMode**

```
public int getMode()
```

Get the current play/publish mode

**Returns:**

play/publish mode

---

**setMode**

```
public void setMode(int mode)
```

Set the play/publish mode

**Parameters:**

mode - play/publish mode

---

**getTransportMode**

```
public String getTransportMode()
```

Get the transport mode

**Returns:**

transport mode

(continued from last page)

## setTransportMode

```
public void setTransportMode(String transportMode)
```

Set the transport mode

**Parameters:**

transportMode - transport mode

---

## 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

(continued from last page)

## formatRTPInfo

```
public String formatRTPInfo(long timecode,
    int videoSeq,
    int audioSeq)
```

Format RTP info

**Parameters:**

- timecode - timecode
- videoSeq - video sequence number
- audioSeq - audio sequence number

**Returns:**

RTP Info string

---

## getRTSPSessionName

```
public String getRTSPSessionName()
```

Get session name

**Returns:**

session name

---

## setRTSPSessionName

```
public void setRTSPSessionName(String rtspSessionName)
```

Set session name

**Parameters:**

- rtspSessionName - session name

---

## getRTSPSessionDescription

```
public String getRTSPSessionDescription()
```

Get session description

**Returns:**

session description

---

## setRTSPSessionDescription

```
public void setRTSPSessionDescription(String rtspSessionDescription)
```

Get session description

**Parameters:**

- rtspSessionDescription - session description

---

## isSendSDEEvents

```
public boolean isSendSDEEvents()
```

Send RTCP SDES events

**Returns:**

(continued from last page)

true if sending RTCP SDES events

---

### setSendSDESEvents

```
public void setSendSDESEvents(boolean sendSDESEvents)
```

Send RTCP SDES events

**Parameters:**

sendSDESEvents - true if sending RTCP SDES events

---

### isMPEGTSOut

```
public boolean isMPEGTSOut()
```

Is MPEG-TS out

**Returns:**

true if MPEG-TS out

---

### setMPEGTSOut

```
public void setMPEGTSOut(boolean isMPEGTSOut)
```

Is MPEG-TS out

**Parameters:**

isMPEGTSOut - true if MPEG-TS out

---

### transportFindBestMatch

```
public String transportFindBestMatch(String transport)
```

Based on a trasport string from SETUP command find best match

**Parameters:**

transport - trasport string

**Returns:**

best match

---

### getAutoAllocateInterleavePorts

```
public int getAutoAllocateInterleavePorts()
```

---

### isLive

```
public boolean isLive()
```

Is live stream

**Returns:**

true if live

(continued from last page)

## setLive

```
public void setLive(boolean isLive)
```

Is live stream

**Parameters:**

isLive - true if live

---

## getMediaReader

```
public IMediaReader getMediaReader()
```

Get media reader if video on demand stream

**Returns:**

media reader

---

## getRTSPSessionTimeout

```
public int getRTSPSessionTimeout()
```

Get RTP session timeout (milliseconds)

**Returns:**

RTP session timeout (milliseconds)

---

## setRTSPSessionTimeout

```
public void setRTSPSessionTimeout(int rtspSessionTimeout)
```

Set RTP session timeout (milliseconds)

**Parameters:**

rtspSessionTimeout - RTP session timeout (milliseconds)

---

## getRTSPMaximumPendingWriteBytes

```
public int getRTSPMaximumPendingWriteBytes()
```

Get the maximum number of waiting bytes allow for this RTSP session

**Returns:**

maximum number of waiting bytes allow for this RTSP session

---

## setRTSPMaximumPendingWriteBytes

```
public void setRTSPMaximumPendingWriteBytes(int rtspMaximumPendingWriteBytes)
```

Set the maximum number of waiting bytes allow for this RTSP session

**Parameters:**

rtspMaximumPendingWriteBytes - maximum number of waiting bytes allow for this RTSP session

---

## isTimeout

```
public boolean isTimeout(long currTime,  
                      int timeout)
```

(continued from last page)

Is the stream timeout out

**Parameters:**

currTime - current time  
timeout - timeout value

**Returns:**

true if timed out

---

**touch**

```
public void touch()
```

Touch the stream so that it does not timeout

---

**getRTSPBindIpAddress**

```
public String getRTSPBindIpAddress()
```

Get the bind RTSP bind IP address

**Returns:**

bind RTSP bind IP address

---

**setRTSPBindIpAddress**

```
public void setRTSPBindIpAddress(String rtspBindIpAddress)
```

Set the bind RTSP bind IP address

**Parameters:**

rtspBindIpAddress - bind RTSP bind IP address

---

**getRTSPConnectionIpAddress**

```
public String getRTSPConnectionIpAddress()
```

Get the connection IP address

**Returns:**

connection IP address

---

**setRTSPConnectionIpAddress**

```
public void setRTSPConnectionIpAddress(String rtspConnectionIpAddress)
```

Set the connection IP address

**Parameters:**

rtspConnectionIpAddress - connection IP address

---

**getRTSPConnectionAddressType**

```
public String getRTSPConnectionAddressType()
```

Get the connection address type

**Returns:**

(continued from last page)

connection address type

---

### setRTSPConnectionAddressType

```
public void setRTSPConnectionAddressType(String rtspConnectionAddressType)
```

Set the connection address type

**Parameters:**

rtspConnectionAddressType - connection address type

---

### getRTSPOriginIpAddress

```
public String getRTSPOriginIpAddress()
```

Get the origin IP address

**Returns:**

origin IP address

---

### setRTSPOriginIpAddress

```
public void setRTSPOriginIpAddress(String rtspOriginIpAddress)
```

Set the origin IP address

**Parameters:**

rtspOriginIpAddress - origin IP address

---

### getRTSPOriginAddressType

```
public String getRTSPOriginAddressType()
```

Get the origin address type

**Returns:**

origin address type

---

### setRTSPOriginAddressType

```
public void setRTSPOriginAddressType(String rtspOriginAddressType)
```

Set the origin address type

**Parameters:**

rtspOriginAddressType - origin address type

---

### getVODStartTimeTC

```
public long getVODStartTimeTC()
```

Get the video on demand start time (milliseconds)

**Returns:**

video on demand start time (milliseconds)

(continued from last page)

## setVODStartTimeTC

```
public void setVODStartTimeTC(long vodStartTimeTC)
```

Set the video on demand start time (milliseconds)

**Parameters:**

vodStartTimeTC - video on demand start time (milliseconds)

---

## getVODLastTimeTC

```
public long getVODLastTimeTC()
```

Get the last timecode (milliseconds) sent for video on demand

**Returns:**

last timecode (milliseconds) sent for video on demand

---

## setVODLastTimeTC

```
public void setVODLastTimeTC(long vodLastTimeTC)
```

Set the last timecode (milliseconds) sent for video on demand

**Parameters:**

vodLastTimeTC - last timecode (milliseconds) sent for video on demand

---

## getVODPlayLen

```
public long getVODPlayLen()
```

Get the video on demand play duration (milliseconds)

**Returns:**

video on demand play duration (milliseconds)

---

## setVODPlayLen

```
public void setVODPlayLen(long vodPlayLen)
```

Set the video on demand play duration (milliseconds)

**Parameters:**

vodPlayLen - video on demand play duration (milliseconds)

---

## isStreamStarted

```
public boolean isStreamStarted()
```

Is stream started

**Returns:**

true if stream has started playback

---

## isRTSPAlwaysUseSDPPorts

```
public boolean isRTSPAlwaysUseSDPPorts()
```

(continued from last page)

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

(continued from last page)

**getSDPLang**

```
public String getSDPLang()
```

Get the SDP language

**Returns:**

SDP language

**setSDPLang**

```
public void setSDPLang(String sdpLang)
```

Set the SDP language

**Parameters:**

sdpLang - SDP language

**isRTPIgnoreProfileLevelId**

```
public boolean isRTPIgnoreProfileLevelId()
```

**setRTPIgnoreProfileLevelId**

```
public void setRTPIgnoreProfileLevelId(boolean rtpIgnoreProfileLevelId)
```

**getUDPManagedDeliveryDelay**

```
public int getUDPManagedDeliveryDelay()
```

**setUDPManagedDeliveryDelay**

```
public void setUDPManagedDeliveryDelay(int udpManagedDeliveryDelay)
```

**getUDPManagedDeliveryCount**

```
public int getUDPManagedDeliveryCount()
```

**setUDPManagedDeliveryCount**

```
public void setUDPManagedDeliveryCount(int udpManagedDeliveryCount)
```

**isForceMPEGTSOut**

```
public boolean isForceMPEGTSOut()
```

(continued from last page)

---

**setForceMPEGTSOut**

```
public void setForceMPEGTSOut(boolean isForceMPEGTSOut)
```

---

**isBlockUDPOut**

```
public boolean isBlockUDPOut()
```

---

**setBlockUDPOut**

```
public void setBlockUDPOut(boolean blockUDPOut)
```

---

**getMPEGTSAudioBitrate**

```
public int getMPEGTSAudioBitrate()
```

---

**setMPEGTSAudioBitrate**

```
public void setMPEGTSAudioBitrate(int mpegtsAudioBitrate)
```

---

**getMPEGTSVideoBitrate**

```
public int getMPEGTSVideoBitrate()
```

---

**setMPEGTSVideoBitrate**

```
public void setMPEGTSVideoBitrate(int mpegtsVideoBitrate)
```

---

**isRTPIgnoreSPropParameterSets**

```
public boolean isRTPIgnoreSPropParameterSets()
```

---

**setRTPIgnoreSPropParameterSets**

```
public void setRTPIgnoreSPropParameterSets(boolean rtpIgnoreSPropParameterSets)
```

---

(continued from last page)

**getRTPStreamContext**

```
public RTPStreamContext getRTPStreamContext()
```

---

**setRTPStreamContext**

```
public void setRTPStreamContext(RTPStreamContext rtpStreamContext)
```

---

**isRefuseIncomingUDPConnections**

```
public boolean isRefuseIncomingUDPConnections()
```

---

**setRefuseIncomingUDPConnections**

```
public void setRefuseIncomingUDPConnections(boolean refuseIncomingUDPConnections)
```

---

**isRefuseOutgoingUDPConnections**

```
public boolean isRefuseOutgoingUDPConnections()
```

---

**setRefuseOutgoingUDPConnections**

```
public void setRefuseOutgoingUDPConnections(boolean refuseOutgoingUDPConnections)
```

---

Package

**com.wowza.wms.server**

## com.wowza.wms.server Interface ICommandInterfaceCommand

public interface ICommandInterfaceCommand  
extends

### Method Summary

boolean	<a href="#">canHandle</a> (CommandInterfaceRequestMessage req)
void	<a href="#">invoke</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 ICompilerDirectives

public interface ICompilerDirectives  
extends

### Field Summary

public static final	<a href="#">RTMPENABLED</a>
---------------------	-----------------------------

Value: **true**

### Fields

#### RTMPENABLED

public static final boolean **RTMPENABLED**

Constant value: **true**

## com.wowza.wms.server Interface IResponseListener

---

public interface **IResponseListener**  
extends

IResponseListener: Internal use.

---

### Method Summary

void	<a href="#">onResponseWriteStart</a> (RtmpResponseMessage response)
void	<a href="#">onResponseWriteStop</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="#"><code>addServerListener(IIServerNotify serverListener)</code></a> Add server listener
java.util.List	<a href="#"><code>getAdminInterfaceObjectList()</code></a> Get the list of objects exposed through JMX interface
RandomIdGenerator	<a href="#"><code>getClientIdGenerator()</code></a> Get the client id generator for the server
CommandInterfaceCommandHandler	<a href="#"><code>getCommandInterfaceCommandHandler()</code></a> Get the command interface command handler
<a href="#"><u>HostPort</u></a>	<a href="#"><code>getCommandInterfaceHostPort()</code></a> Get the definition of the command interface
<a href="#"><u>ConnectionCounter</u></a>	<a href="#"><code>getConnectionCounter()</code></a> Get the server connection counter.
ConnectionCounterSimple	<a href="#"><code>getConnectionCounter(int counterIndex)</code></a> Get the server connection counter for a specific technology (see IVHost.COUNTER_*)
long	<a href="#"><code>getConnectionsMaximum()</code></a> Returns maximum number of connections allowed to the server (-1 means unlimited)
int	<a href="#"><code>getCoreHandlerPoolSize()</code></a> Get the handler core thread pool size.
int	<a href="#"><code>getCoreTransportPoolSize()</code></a> Get the transport core thread pool size.
String	<a href="#"><code>getDateStarted()</code></a> Get the date and time the server was started.
String	<a href="#"><code>getDefaultStreamPrefix()</code></a> Get the default stream prefix used if the stream name does not include a prefix (default is mp4)
java.util.Properties	<a href="#"><code>getDynamicLogProperties()</code></a> Get the dynamic log properties defined at the server level in conf/log4j.properties
<a href="#"><u>ThreadPool</u></a>	<a href="#"><code>getHandlerThreadPool()</code></a> Get the server handler thread pool.

<a href="#">IOPerformanceCounter</a>	<a href="#"><code>getIoPerformanceCounter()</code></a> Get the server performance counter.
<a href="#">IOPerformanceCounter</a>	<a href="#"><code>getIoPerformanceCounter(int counterIndex)</code></a> Get the server performance counter for a specific technology (see IVHost.COUNTER_*)
long	<a href="#"><code>getLiveStreamTranscoderSessionCount()</code></a> Get the current number of live stream transcoder sessions.
<a href="#">WMSProperties</a>	<a href="#"><code>getManagerProperties()</code></a> Get server's manager's properties collection
<a href="#">WMSProperties</a>	<a href="#"><code>getProperties()</code></a> Get server level properties collection
long	<a href="#"><code>getPublishersMaximum()</code></a> Get the maximum number of streams that can be published to the server
<a href="#">ThreadPool</a>	<a href="#"><code>getThreadPool()</code></a> Get the server handler thread pool.
String	<a href="#"><code>getTimeRunning()</code></a> Get a formatted String of how long the server has been running.
double	<a href="#"><code>getTimeRunningSeconds()</code></a> Get time running in seconds
long	<a href="#"><code>getTranscoderLicenseInUse()</code></a> Get the total number of transcoder licenses in use.
long	<a href="#"><code>getTranscoderLicenseTotal()</code></a> Get the total number of transcoder licenses (-1 means unlimited - subscription)
<a href="#">ThreadPool</a>	<a href="#"><code>getTransportThreadPool()</code></a> Get the server transport thread pool.
<a href="#">ServerTuningHandler</a>	<a href="#"><code>getTuningHandler()</code></a> Get the handler that control automatic server tuning
<a href="#">com.wowza.wms.transport.udp.UDPPortManager</a>	<a href="#"><code>getUDPPortManager()</code></a> Get the UDP port manager which manages the allocation of incoming UDP port binding to be sure there are not port conflicts
<a href="#">com.wowza.wms.transport.udp.UDPPortSharingManager</a>	<a href="#"><code>getUDPPortSharingManager()</code></a> Get the UDP port sharing manager.
String[]	<a href="#"><code>getUserAgents()</code></a> Get a pipe " " delimited list of user agents that the server recognizes as RTMPT client.
String	<a href="#"><code>getVersion()</code></a> Get server version number.
<a href="#">VHostList</a>	<a href="#"><code>getVHostList()</code></a> Returns the interface to the VHostList for the server
boolean	<a href="#"><code>isAcceptWOWZConnections()</code></a> If true, WOWZ connections will use the WOWZ protocol.

boolean	<a href="#"><u>isDRMAvailable()</u></a> Returns true if DRM AddOn is properly licensed.
boolean	<a href="#"><u>isDRMInUse()</u></a> Returns true if DRM AddOn is in-use.
boolean	<a href="#"><u>isDVRAvailable()</u></a> Returns true if nDVR AddOn is properly licensed.
boolean	<a href="#"><u>isDVRInUse()</u></a> Returns true if nDVR AddOn is in-use.
boolean	<a href="#"><u>isDynamicLogContextLoaded(String logContext)</u></a> Returns true of the given dynamic log context is already loaded.
boolean	<a href="#"><u>isHTTPOriginAvailable()</u></a> Returns true if HTTP origin is properly licensed.
boolean	<a href="#"><u>isInitiateWOWZConnections()</u></a> If true, NetConnection attempts will use the WOWZ protocol when possible.
boolean	<a href="#"><u>isPushPublishAvailable()</u></a> Returns true if Push Publish is properly licensed.
boolean	<a href="#"><u>isRESTAPIAvailable()</u></a> Returns true if REST API is properly licensed.
boolean	<a href="#"><u>isRestServerEnabled()</u></a> Returns true if REST API is enabled
boolean	<a href="#"><u>isSuspended()</u></a> Is the server current suspended
boolean	<a href="#"><u>isTranscoderWatermark()</u></a> Return true if transcoder has a watermark
String	<a href="#"><u>readConfig(String sName)</u></a> Method to read xml config file..
void	<a href="#"><u>reloadVHostConfig()</u></a> Reload the VHosts.xml file.
void	<a href="#"><u>removeServerListener(IserverNotify serverListener)</u></a> Remove server listener
void	<a href="#"><u>setAcceptWOWZConnections(boolean acceptWOWZConnections)</u></a> If true, WOWZ connections will use the WOWZ protocol.
void	<a href="#"><u>setCommandInterfaceHostPort(HostPort commandInterfaceHostPort)</u></a> Set the definition for the command interface.
void	<a href="#"><u>setCoreHandlerPoolSize(int corePoolSize)</u></a> Set the handler core thread pool size.
void	<a href="#"><u>setCoreTransportPoolSize(int corePoolSize)</u></a> Set the transport core thread pool size.
void	<a href="#"><u>setDefaultStreamPrefix(String defaultStreamPrefix)</u></a> Set the default stream prefix used if the stream name does not include a prefix (default is mp4)

void	<a href="#"><u>setDynamicLogProperties</u></a> ( java.util.Properties dynamicLogProperties) Set the dynamic log properties set at the server level
void	<a href="#"><u>setInitiateWOWZConnections</u></a> (boolean initiateWOWZConnections) If true, NetConnection attempts will use the WOWZ protocol when possible.
void	<a href="#"><u>setRestServerEnabled</u></a> (boolean restServerEnabled) Sets if REST API server will be enabled
void	<a href="#"><u>setUserAgents</u></a> (String[] userAgents) Set a pipe " " delimited list of user agents that the server recognizes as RTMPT client.
void	<a href="#"><u>startCommandInterface</u></a> () Start the command interface as defined in Server.xml.
void	<a href="#"><u>startVHost</u></a> (String vhostName) Start a vHost by name.
void	<a href="#"><u>startVHosts</u></a> () Start all vHosts
void	<a href="#"><u>stopAdminAgent</u></a> () Stop the JMX interface
void	<a href="#"><u>stopCommandInterface</u></a> () Stop the command interface as defined in Server.xml.
void	<a href="#"><u>stopVHost</u></a> (String vhostName) Stop a vHost by name.
void	<a href="#"><u>stopVHosts</u></a> () Stop all vHosts
void	<a href="#"><u>suspendAllVHosts</u></a> () Suspend all virtual hosts (Calls IVHost.suspendAllHostPorts for each vhost)
void	<a href="#"><u>suspendServer</u></a> () Suspend all virtual hosts and the command interface
void	<a href="#"><u>unbindAllVHosts</u></a> () Unbind all virtual hosts (Calls IVHost.unbindAllHostPorts for each vhost)
boolean	<a href="#"><u>writeConfig</u></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()
```

(continued from last page)

Stop the command interface as defined in Server.xml. The command interface is used by shutdown.sh script to stop the server. It is also used by ant task to stop and start the server on build events.

---

## getVersion

```
public String getVersion()
```

Get server version number.

**Returns:**

server version number

---

## reloadVHostConfig

```
public void reloadVHostConfig()
```

Reload the VHosts.xml file. This method can be invoked through the JMX interface to manage vHosts while the server is running.

---

## stopVHost

```
public void stopVHost(String vhostName)
```

Stop a vHost by name.

**Parameters:**

vhostName - vHost name

---

## stopVHosts

```
public void stopVHosts()
```

Stop all vHosts

---

## startVHost

```
public void startVHost(String vhostName)
```

Start a vHost by name.

**Parameters:**

vhostName - vHost name

---

## startVHosts

```
public void startVHosts()
```

Start all vHosts

---

## getIoPerformanceCounter

```
public IOPerformanceCounter getIoPerformanceCounter()
```

Get the server performance counter.

**Returns:**

io performance counter

---

## getIoPerformanceCounter

```
public IOPerformanceCounter getIoPerformanceCounter(int counterIndex)
```

Get the server performance counter for a specific technology (see IVHost.COUNTER\_\*)

**Parameters:**

counterIndex - counter index (see IVHost.COUNTER\_\*)

**Returns:**

io performance counter

---

## getConnectionCounter

```
public ConnectionCounter getConnectionCounter()
```

Get the server connection counter.

**Returns:**

connection counter

---

## getPublishersMaximum

```
public long getPublishersMaximum()
```

Get the maximum number of streams that can be published to the server

**Returns:**

maximum number of streams that can be published to the server

---

## getConnectionCounter

```
public ConnectionCounterSimple getConnectionCounter(int counterIndex)
```

Get the server connection counter for a specific technology (see IVHost.COUNTER\_\*)

**Parameters:**

counterIndex - counter index (see IVHost.COUNTER\_\*)

**Returns:**

connection counter

---

## getDateStarted

```
public String getDateStarted()
```

Get the date and time the server was started.

**Returns:**

date and time the server was started

---

## getTimeRunning

```
public String getTimeRunning()
```

Get a formatted String of how long the server has been running.

(continued from last page)

**Returns:**

formatted String of how long the server has been running

---

**getTimeRunningSeconds****public double getTimeRunningSeconds( )**

Get time running in seconds

**Returns:**

time running in seconds

---

**getCommandInterfaceHostPort****public HostPort getCommandInterfaceHostPort( )**

Get the definition of the command interface

**Returns:**

host port definition of command interface

---

**setCommandInterfaceHostPort****public void setCommandInterfaceHostPort(HostPort commandInterfaceHostPort)**

Set the definition for the command interface.

**Parameters:**

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

(continued from last page)

## removeServerListener

```
public void removeServerListener(IServerNotify serverListener)
```

Remove server listener

**Parameters:**

serverListener - server listener

---

## getAdminInterfaceObjectList

```
public java.util.List getAdminInterfaceObjectList()
```

Get the list of objects exposed through JMX interface

**Returns:**

list of objects exposed through JMX interface

---

## getCoreTransportPoolSize

```
public int getCoreTransportPoolSize()
```

Get the transport core thread pool size.

**Returns:**

default core thread pool size

---

## 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()
```

(continued from last page)

Get the server handler thread pool. Same as getHandlerThreadPool.

**Returns:**

server handler thread pool

## getTransportThreadPool

```
public ThreadPool getTransportThreadPool()
```

Get the server transport thread pool. This thread pool is used to read/write data from the transports sockets.

**Returns:**

server transport thread pool

## getHandlerThreadPool

```
public ThreadPool getHandlerThreadPool()
```

Get the server handler thread pool. This thread pool is used to process the incoming events.

**Returns:**

server handler thread pool

## getProperties

```
public WMSProperties getProperties()
```

Get server level properties collection

**Returns:**

server level properties collection

## getManagerProperties

```
public WMSProperties getManagerProperties()
```

Get server's manager's properties collection

**Returns:**

manager's property collection

## getClientIdGenerator

```
public RandomIdGenerator getClientIdGenerator()
```

Get the client id generator for the server

**Returns:**

client id generator for the server

## getDynamicLogProperties

```
public java.util.Properties getDynamicLogProperties()
```

Get the dynamic log properties defined at the server level in conf/log4j.properties

**Returns:**

dynamic log properties defined at the server level

---

## setDynamicLogProperties

```
public void setDynamicLogProperties(java.util.Properties dynamicLogProperties)
```

Set the dynamic log properties set at the server level

**Parameters:**

dynamicLogProperties - dynamic log properties defined at the server level

---

## isDynamicLogContextLoaded

```
public boolean isDynamicLogContextLoaded(String logContext)
```

Returns true if the given dynamic log context is already loaded. If not loaded it will return false and add it to the a Set of loaded log context. Log context is [VHost].[Application].[AppInstance]

---

## getVHostList

```
public VHostList getVHostList()
```

Returns the interface to the VHostList for the server

**Returns:**

vhostList

---

## 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 currently suspended

---

## stopAdminAgent

```
public void stopAdminAgent()
```

Stop the JMX interface

---

(continued from last page)

## 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)
```

Method to write xml config file..

---

## getCommandInterfaceCommandHandler

```
public CommandInterfaceCommandHandler getCommandInterfaceCommandHandler()
```

Get the command interface command handler

**Returns:**

command interface command handler

---

## isAcceptWOWZConnections

```
public boolean isAcceptWOWZConnections()
```

If true, WOWZ connections will use the WOWZ protocol. If false, WOWZ connection attempts will fall back to RTMP.

**Returns:**

true, WOWZ connections will use the WOWZ protocol

---

## setAcceptWOWZConnections

```
public void setAcceptWOWZConnections(boolean acceptWOWZConnections)
```

If true, WOWZ connections will use the WOWZ protocol. If false, WOWZ connection attempts will fall back to RTMP.

**Parameters:**

acceptWOWZConnections - true, WOWZ connections will use the WOWZ protocol

---

## isInitiateWOWZConnections

```
public boolean isInitiateWOWZConnections()
```

If true, NetConnection attempts will use the WOWZ protocol when possible.

**Returns:**

true, NetConnection attempts will use the WOWZ protocol when possible.

---

## setInitiateWOWZConnections

```
public void setInitiateWOWZConnections(boolean initiateWOWZConnections)
```

If true, NetConnection attempts will use the WOWZ protocol when possible.

**Parameters:**

initiateWOWZConnections - true, NetConnection attempts will use the WOWZ protocol when possible.

---

## getLiveStreamTranscoderSessionCount

```
public long getLiveStreamTranscoderSessionCount()
```

Get the current number of live stream transcoder sessions.

**Returns:**

current number of live stream transcoder sessions

---

## getTuningHandler

```
public ServerTuningHandler getTuningHandler()
```

Get the handler that control automatic server tuning

**Returns:**

handler that control automatic server tuning

---

## getTranscoderLicenseInUse

```
public long getTranscoderLicenseInUse()
```

Get the total number of transcoder licenses in use.

**Returns:**

total number of transcoder licenses in use.

---

## getTranscoderLicenseTotal

```
public long getTranscoderLicenseTotal()
```

Get the total number of transcoder licenses (-1 means unlimited - subscription)

**Returns:**

total number of transcoder licenses

---

## isTranscoderWatermark

```
public boolean isTranscoderWatermark()
```

(continued from last page)

Return true if transcoder has a watermark

**Returns:**

true if transcoder has a watermark

---

**isDVRAvailable**

```
public boolean isDVRAvailable()
```

Returns true if nDVR AddOn is properly licensed.

**Returns:**

true if nDVR AddOn is properly licensed

---

**isDVRInUse**

```
public boolean isDVRInUse()
```

Returns true if nDVR AddOn is in-use.

**Returns:**

true if nDVR AddOn is in-use

---

**isDRMAvailable**

```
public boolean isDRMAvailable()
```

Returns true if DRM AddOn is properly licensed.

**Returns:**

true if DRM AddOn is properly licensed

---

**isDRMInUse**

```
public boolean isDRMInUse()
```

Returns true if DRM AddOn is in-use.

**Returns:**

true if DRM AddOn is in-use

---

**isHTTPOriginAvailable**

```
public boolean isHTTPOriginAvailable()
```

Returns true if HTTP origin is properly licensed.

**Returns:**

true if HTTP origin is properly licensed

---

**isRESTAPIAvailable**

```
public boolean isRESTAPIAvailable()
```

Returns true if REST API is properly licensed.

**Returns:**

true if REST API is properly licensed

---

## isPushPublishAvailable

```
public boolean isPushPublishAvailable()
```

Returns true if Push Publish is properly licensed.

**Returns:**

---

## setRestServerEnabled

```
public void setRestServerEnabled(boolean restServerEnabled)
```

Sets if REST API server will be enabled

**Parameters:**

restServerEnabled, - if the REST API server will be enabled

---

## isRestServerEnabled

```
public boolean isRestServerEnabled()
```

Returns true if REST API is enabled

**Returns:**

true if REST API is enabled

---

## getConnectionsMaximum

```
public long getConnectionsMaximum()
```

Returns maximum number of connections allowed to the server (-1 means unlimited)

**Returns:**

maximum number of connections allowed to the server (-1 means unlimited)

---

## getDefaultStreamPrefix

```
public String getDefaultStreamPrefix()
```

Get the default stream prefix used if the stream name does not include a prefix (default is mp4)

**Returns:**

default stream prefix

---

## setDefaultStreamPrefix

```
public void setDefaultStreamPrefix(String defaultStreamPrefix)
```

Set the default stream prefix used if the stream name does not include a prefix (default is mp4)

**Parameters:**

defaultStreamPrefix - default stream prefix

## com.wowza.wms.server Interface IServerNotify

All Subinterfaces:

[IServerNotify2](#), [IServerNotify3](#)

public interface **IServerNotify**

extends

IServerNotify: listener interface. Configured by adding class entries definitions to Server.xml.Startup order is: [constructor]; onServerConfigLoaded, onServerCreate, onServerInit Shutdown order is: onServerShutdownStart, onServerShutdownComplete, [exit]

### Method Summary

void	<a href="#">onServerCreate(<i>IServer</i> server)</a> Triggered when server object is first created.
void	<a href="#">onServerInit(<i>IServer</i> server)</a> Triggered when server initialization is complete and all VHosts have been started
void	<a href="#">onServerShutdownComplete(<i>IServer</i> server)</a> Triggered at the end of server shutdown
void	<a href="#">onServerShutdownStart(<i>IServer</i> server)</a> Triggered at the beginning of server shutdown

### Methods

#### onServerCreate

public void **onServerCreate(*IServer* server)**

Triggered when server object is first created.

**Parameters:**

server - server object

#### onServerInit

public void **onServerInit(*IServer* server)**

Triggered when server initialization is complete and all VHosts have been started

**Parameters:**

server - server object

#### onServerShutdownStart

public void **onServerShutdownStart(*IServer* server)**

Triggered at the beginning of server shutdown

(continued from last page)

**Parameters:**

server - server object

---

**onServerShutdownComplete**`public void onServerShutdownComplete(IServer server)`

Triggered at the end of server shutdown

**Parameters:**

server - server object

## com.wowza.wms.server Interface IServerNotify2

All Superinterfaces:

[IServerNotify](#)

All Subinterfaces:

[IServerNotify3](#)

---

public interface **IServerNotify2**  
extends [IServerNotify](#)

IServerNotify2: listener interface. Configured by adding class entries definitions to Server.xml.Startup order is: [constructor];  
onServerConfigLoaded, onServerCreate, onServerInit Shutdown order is: onServerShutdownStart, onServerShutdownComplete, [exit]

---

### Method Summary

void	<a href="#">onServerConfigLoaded(<a href="#">IServer</a> server)</a>
	Triggered when server configuration is loaded

Methods inherited from interface [com.wowza.wms.server.IServerNotify](#)

[onServerCreate](#), [onServerInit](#), [onServerShutdownComplete](#), [onServerShutdownStart](#)

---

### Methods

#### onServerConfigLoaded

public void **onServerConfigLoaded([IServer](#) server)**

Triggered when server configuration is loaded

**Parameters:**

server - server object

## com.wowza.wms.server Interface IServerNotify3

All Superinterfaces:

[IServerNotify2](#), [IServerNotify](#)

public interface **IServerNotify3**

extends [IServerNotify2](#)

### Method Summary

void	<a href="#">onServerAfterVHostStartup(<i>IServer</i> server)</a> Triggered just after virtual hosts are started.
void	<a href="#">onServerBeforeVHostStartup(<i>IServer</i> server)</a> Triggered just before virtual hosts are started.

Methods inherited from interface [com.wowza.wms.server.IServerNotify2](#)

[onServerConfigLoaded](#)

Methods inherited from interface [com.wowza.wms.server.IServerNotify](#)

[onServerCreate](#), [onServerInit](#), [onServerShutdownComplete](#), [onServerShutdownStart](#)

### Methods

#### **onServerBeforeVHostStartup**

public void **onServerBeforeVHostStartup(*IServer* server)**

Triggered just before virtual hosts are started.

**Parameters:**

server - server object

#### **onServerAfterVHostStartup**

public void **onServerAfterVHostStartup(*IServer* server)**

Triggered just after virtual hosts are started.

**Parameters:**

server - server object

## com.wowza.wms.server

### Class Server

```
java.lang.Object
+-com.wowza.wms.server.Server
```

All Implemented Interfaces:

[IServer](#)

```
public class Server
extends Object
implements IServer
```

### Nested Class Summary

class	<a href="#">Server.TranscoderPollingTracker</a> Server.TranscoderPollingTracker
-------	--

### Field Summary

public	<a href="#">debugWOWZProtocol</a>
public static	<a href="#">logNotifier</a>

### Constructor Summary

public	<a href="#">Server()</a>
--------	--------------------------

### Method Summary

void	<a href="#">addServerListener(<a href="#">IServerNotify</a> serverListener)</a>
static void	<a href="#">decodeS(String[] ins)</a> <i>Deprecated.</i>
static boolean	<a href="#">decodeSS(String in)</a> <i>Deprecated.</i>
String	<a href="#">decodeStorageDir(<a href="#">IVHost</a> vhost, String storageDir)</a>
void	<a href="#">doWatchdog()</a>
static String	<a href="#">fS(byte[] kiIn, String p)</a> <i>Deprecated.</i>
com.wowza.wms.admin.AdminAgent	<a href="#">getAdminAgent()</a>
String	<a href="#">getAdminGUID()</a>

<code>java.util.List</code>	<a href="#"><code>getAdminInterfaceObjectList()</code></a>
<code>RandomIdGenerator</code>	<a href="#"><code>getClientIdGenerator()</code></a>
<code>int</code>	<a href="#"><code>getClientIdGeneratorRecycleDelaySize()</code></a>
<code>int</code>	<a href="#"><code>getClientIdGeneratorRecycleSize()</code></a>
<code>long</code>	<a href="#"><code>getClientIdGeneratorTimeout()</code></a>
<code>Object</code>	<a href="#"><code>getCommandInterface()</code></a>
<code>CommandInterfaceComma ndHandler</code>	<a href="#"><code>getCommandInterfaceCommandHandler()</code></a>
<code>HostPort</code>	<a href="#"><code>getCommandInterfaceHostPort()</code></a>
<code>long</code>	<a href="#"><code>getCommittedVirtualMemory()</code></a>
<code>ConnectionCounter</code>	<a href="#"><code>getConnectionCounter()</code></a>
<code>ConnectionCounterSimp le</code>	<a href="#"><code>getConnectionCounter(int counterIndex)</code></a>
<code>long</code>	<a href="#"><code>getConnectionsMaximum()</code></a>
<code>IConnectionValidator</code>	<a href="#"><code>getConnectionValidator()</code></a>
<code>int</code>	<a href="#"><code>getCoreHandlerPoolSize()</code></a>
<code>int</code>	<a href="#"><code>getCoreTransportPoolSize()</code></a>
<code>int</code>	<a href="#"><code>getCryptoPoolActiveCount()</code></a>
<code>int</code>	<a href="#"><code>getCryptoPoolMaxSize()</code></a>
<code>long</code>	<a href="#"><code>getCurrentHeapSize()</code></a>
<code>String</code>	<a href="#"><code>getDateStarted()</code></a>
<code>String</code>	<a href="#"><code>getDefaultStreamPrefix()</code></a>
<code>java.util.Properties</code>	<a href="#"><code>getDynamicLogProperties()</code></a>
<code>String</code>	<a href="#"><code>getGUID()</code></a>
<code>ThreadPool</code>	<a href="#"><code>getHandlerThreadPool()</code></a>
<code>static Server</code>	<a href="#"><code>getInstance()</code></a>

<a href="#">IOPerformanceCounter</a>	<a href="#">getIOPerformanceCounter()</a>
<a href="#">IOPerformanceCounter</a>	<a href="#">getIOPerformanceCounter(int counterIndex)</a>
<a href="#">JMXRemoteConfig</a>	<a href="#">getJmxRemoteConfig()</a>
<a href="#">LicenseCounter</a>	<a href="#">getLicenseCounter(int index)</a>
<a href="#">Server.LicenseSessionTracker</a>	<a href="#">getLicenseTracker(int index)</a>
<a href="#">ILicenseValidator</a>	<a href="#">getLicenseValidator()</a>
long	<a href="#">getLiveStreamTranscoderSessionCount()</a>
long	<a href="#">getLiveThreads()</a>
<a href="#">WMSProperties</a>	<a href="#">getManagerProperties()</a>
long	<a href="#">getMaxHeapSize()</a>
long	<a href="#">getPeakThreads()</a>
<a href="#">WMSProperties</a>	<a href="#">getProperties()</a>
boolean[]	<a href="#">getProtocolUsage()</a>
<a href="#">Server.ProtocolUsageSessionTracker</a>	<a href="#">getProtocolUsageTracker()</a>
long	<a href="#">getPublishersMaximum()</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>
long	<a href="#">getTranscoderLicenseInUse()</a>
long	<a href="#">getTranscoderLicenseTotal()</a>
<a href="#">ThreadPool</a>	<a href="#">getTransportThreadPool()</a>
<a href="#">ServerTuningHandler</a>	<a href="#">getTuningHandler()</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>
VHostList	<a href="#">getVHostList()</a>
boolean	<a href="#">isAcceptWOWZConnections()</a>
boolean	<a href="#">isDRMAvailable()</a>
boolean	<a href="#">isDRMInUse()</a>
boolean	<a href="#">isDVRAvailable()</a>
boolean	<a href="#">isDVRInUse()</a>
boolean	<a href="#">isDynamicLogContextLoaded(String logContext)</a>
boolean	<a href="#">isHTTPOriginAvailable()</a>
boolean	<a href="#">isInitiateWOWZConnections()</a>
boolean	<a href="#">isPushPublishAvailable()</a>
boolean	<a href="#">isRESTAPIAvailable()</a>
boolean	<a href="#">isRestServerEnabled()</a>
boolean	<a href="#">isSuspended()</a>
boolean	<a href="#">isTranscoderWatermark()</a>
boolean	<a href="#">isVHostRunning(String name)</a>
static void	<a href="#">main(String[] args)</a>
void	<a href="#">onNewVHost(IVHost vhost)</a>
String	<a href="#">readConfig(String sName)</a>
static String	<a href="#">readXMLConfig(String sPath)</a>

void	<a href="#"><u>registerLiveStreamTranscoder(ILiveStreamTranscoder liveStreamTranscoder, byte[] license)</u></a>
void	<a href="#"><u>reloadVHostConfig()</u></a>
void	<a href="#"><u>removeServerListener(IServerNotify serverListener)</u></a>
void	<a href="#"><u>setAcceptWOWZConnections(boolean acceptWOWZConnections)</u></a>
void	<a href="#"><u>setCommandInterface(Object commandInterface)</u></a>
void	<a href="#"><u>setCommandInterfaceHostPort(HostPort commandInterfaceHostPort)</u></a>
void	<a href="#"><u>setCoreHandlerPoolSize(int corePoolSize)</u></a>
void	<a href="#"><u>setCoreTransportPoolSize(int corePoolSize)</u></a>
void	<a href="#"><u>setDefaultStreamPrefix(String defaultStreamPrefix)</u></a>
void	<a href="#"><u>setDynamicLogProperties(java.util.Properties dynamicLogProperties)</u></a>
void	<a href="#"><u>setInitiateWOWZConnections(boolean initiateWOWZConnections)</u></a>
void	<a href="#"><u>setIoPerformanceCounter(IOPerformanceCounter ioPerformanceCounter)</u></a>
void	<a href="#"><u>setRestServerEnabled(boolean restServerEnabled)</u></a>
void	<a href="#"><u>setUserAgents(String[] userAgents)</u></a>
static void	<a href="#"><u>start()</u></a>
void	<a href="#"><u>startCommandInterface()</u></a>
void	<a href="#"><u>startServer()</u></a>
void	<a href="#"><u>startStats()</u></a>
void	<a href="#"><u>startVHost(String vhostName)</u></a>
void	<a href="#"><u>startVHosts()</u></a>
void	<a href="#"><u>stopAdminAgent()</u></a>
void	<a href="#"><u>stopCommandInterface()</u></a>
void	<a href="#"><u>stopServer()</u></a>
void	<a href="#"><u>stopStats()</u></a>

void	<a href="#">stopVHost(String vhostName)</a>
void	<a href="#">stopVHosts()</a>
void	<a href="#">suspendAllVHosts()</a>
void	<a href="#">suspendCommandInterface()</a>
void	<a href="#">suspendServer()</a>
void	<a href="#">unbindAllVHosts()</a>
void	<a href="#">unregisterLiveStreamTranscoder(ILiveStreamTranscoder LiveStreamTranscoder, byte[] license)</a>
<a href="#">IOPerformanceCounter</a>	<a href="#">updateIOPerformance()</a>
void	<a href="#">updateLoggingDuration()</a>
boolean	<a href="#">writeConfig(String sName, String data)</a>
static boolean	<a href="#">writeXMLConfig(String sPath, String data)</a>

**Methods inherited from class java.lang.Object**

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait`

**Methods inherited from interface com.wowza.wms.server.IServer**

[addServerListener](#), [getAdminInterfaceObjectList](#), [getClientIdGenerator](#),  
[getCommandInterfaceCommandHandler](#), [getCommandInterfaceHostPort](#), [getConnectionCounter](#),  
[getConnectionCounter](#), [getConnectionsMaximum](#), [getCoreHandlerPoolSize](#),  
[getCoreTransportPoolSize](#), [getDateStarted](#), [getDefaultStreamPrefix](#),  
[getDynamicLogProperties](#), [getHandlerThreadPool](#), [getIoPerformanceCounter](#),  
[getIoPerformanceCounter](#), [getLiveStreamTranscoderSessionCount](#), [getManagerProperties](#),  
[getProperties](#), [getPublishersMaximum](#), [getThreadPool](#), [getTimeRunning](#),  
[getTimeRunningSeconds](#), [getTranscoderLicenseInUse](#), [getTranscoderLicenseTotal](#),  
[getTransportThreadPool](#), [getTuningHandler](#), [getUDPPortManager](#),  
[getUDPPortSharingManager](#), [getUserAgents](#), [getVersion](#), [getVHostList](#),  
[isAcceptWOWZConnections](#), [isDRMAvailable](#), [isDRMInUse](#), [isDVRAvailable](#), [isDVRInUse](#),  
[isDynamicLogContextLoaded](#), [isHTTPOriginAvailable](#), [isInitiateWOWZConnections](#),  
[isPushPublishAvailable](#), [isRESTAPIAvailable](#), [isRestServerEnabled](#), [isSuspended](#),  
[isTranscoderWatermark](#), [readConfig](#), [reloadVHostConfig](#), [removeServerListener](#),  
[setAcceptWOWZConnections](#), [setCommandInterfaceHostPort](#), [setCoreHandlerPoolSize](#),  
[setCoreTransportPoolSize](#), [setDefaultStreamPrefix](#), [setDynamicLogProperties](#),  
[setInitiateWOWZConnections](#), [setRestServerEnabled](#),  [setUserAgents](#),  
[startCommandInterface](#), [startVHost](#), [startVHosts](#), [stopAdminAgent](#), [stopCommandInterface](#),  
[stopVHost](#), [stopVHosts](#), [suspendAllVHosts](#), [suspendServer](#), [unbindAllVHosts](#), [writeConfig](#)

**Fields**

(continued from last page)

**logNotifier**

```
public static com.wowza.wms.logging.ILogNotify logNotifier
```

**debugWOWZProtocol**

```
public boolean debugWOWZProtocol
```

## Constructors

**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()
```

---

---

### **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()
```

---

(continued from last page)

**decodeStorageDir**

```
public String decodeStorageDir(IVHost vhost,  
    String storageDir)
```

---

**startVHost**

```
public void startVHost(String vhostName)
```

---

**isSuspended**

```
public boolean isSuspended()
```

---

**suspendServer**

```
public void suspendServer()
```

---

**suspendAllVHosts**

```
public void suspendAllVHosts()
```

---

**unbindAllVHosts**

```
public void unbindAllVHosts()
```

---

**startVHosts**

```
public void startVHosts()
```

---

**doWatchdog**

```
public void doWatchdog()
```

---

**updateIOPerformance**

```
public IOPerformanceCounter updateIOPerformance()
```

---

**getIoPerformanceCounter**

```
public IOPerformanceCounter getIoPerformanceCounter()
```

(continued from last page)

---

**getIoPerformanceCounter**

```
public IOPerformanceCounter getIoPerformanceCounter(int counterIndex)
```

---

**setIoPerformanceCounter**

```
public void setIoPerformanceCounter(IOPerformanceCounter ioPerformanceCounter)
```

---

**getConnectionCounter**

```
public ConnectionCounter getConnectionCounter()
```

---

**getConnectionCounter**

```
public ConnectionCounterSimple getConnectionCounter(int counterIndex)
```

---

**getConnectionValidator**

```
public IConnectionValidator getConnectionValidator()
```

---

**getLicenseValidator**

```
public ILicenseValidator getLicenseValidator()
```

---

**onNewVHost**

```
public void onNewVHost(IVHost vhost)
```

---

**getDateStarted**

```
public String getDateStarted()
```

---

**getTimeRunning**

```
public String getTimeRunning()
```

---

(continued from last page)

**getTimeRunningSeconds**

```
public double getTimeRunningSeconds()
```

---

**getCommandInterfaceHostPort**

```
public HostPort getCommandInterfaceHostPort()
```

---

**setCommandInterfaceHostPort**

```
public void setCommandInterfaceHostPort(HostPort commandInterfaceHostPort)
```

---

**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(I ServerNotify serverListener)
```

---

**removeServerListener**

```
public void removeServerListener(I ServerNotify serverListener)
```

(continued from last page)

---

**getJmxRemoteConfig**

```
public JMXRemoteConfig getJmxRemoteConfig()
```

---

**getAdminInterfaceObjectList**

```
public java.util.List getAdminInterfaceObjectList()
```

---

**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()
```

---

(continued from last page)

**getProperties**

```
public WMSProperties getProperties()
```

---

**getManagerProperties**

```
public WMSProperties getManagerProperties()
```

---

**getClientIdGenerator**

```
public RandomIdGenerator getClientIdGenerator()
```

---

**getDynamicLogProperties**

```
public java.util.Properties getDynamicLogProperties()
```

---

**setDynamicLogProperties**

```
public void setDynamicLogProperties(java.util.Properties dynamicLogProperties)
```

---

**isDynamicLogContextLoaded**

```
public boolean isDynamicLogContextLoaded(String logContext)
```

---

**getVHostList**

```
public VHostList getVHostList()
```

---

**getSessionGUID**

```
public String getSessionGUID()
```

---

**getServerGUID**

```
public String getServerGUID()
```

---

**getGUID**

```
public String getGUID()
```

---

(continued from last page)

---

**getAdminGUID**

```
public String getAdminGUID()
```

---

**getUDPPortSharingManager**

```
public com.wowza.wms.transport.udp.UDPPortSharingManager getUDPPortSharingManager()
```

---

**getUDPPortManager**

```
public com.wowza.wms.transport.udp.UDPPortManager getUDPPortManager()
```

---

**getCryptoPoolMaxSize**

```
public int getCryptoPoolMaxSize()
```

---

**getCryptoPoolActiveCount**

```
public int getCryptoPoolActiveCount()
```

---

**getLiveThreads**

```
public long getLiveThreads()
```

---

**getPeakThreads**

```
public long getPeakThreads()
```

---

**getCurrentHeapSize**

```
public long getCurrentHeapSize()
```

---

**getMaxHeapSize**

```
public long getMaxHeapSize()
```

(continued from last page)

**getCommittedVirtualMemory**

```
public long getCommittedVirtualMemory()
```

**readConfig**

```
public String readConfig(String sName)
```

**writeConfig**

```
public boolean writeConfig(String sName,  
                           String data)
```

**readXMLConfig**

```
public static String readXMLConfig(String sPath)
```

**writeXMLConfig**

```
public static boolean writeXMLConfig(String sPath,  
                                     String data)
```

**getClientIdGeneratorTimeout**

```
public long getClientIdGeneratorTimeout()
```

**getClientIdGeneratorRecycleSize**

```
public int getClientIdGeneratorRecycleSize()
```

**getClientIdGeneratorRecycleDelaySize**

```
public int getClientIdGeneratorRecycleDelaySize()
```

**getCommandInterfaceCommandHandler**

```
public CommandInterfaceCommandHandler getCommandInterfaceCommandHandler()
```

(continued from last page)

**getLicenseCounter**

```
public LicenseCounter getLicenseCounter(int index)
```

---

**getLicenseTracker**

```
public Server.LicenseSessionTracker getLicenseTracker(int index)
```

---

**decodeS**

```
public static void decodeS(String[] ins)
```

Deprecated.

---

**decodeSS**

```
public static boolean decodeSS(String in)
```

Deprecated.

---

**fS**

```
public static String fS(byte[] kiIn,  
String p)
```

Deprecated.

---

**registerLiveStreamTranscoder**

```
public void registerLiveStreamTranscoder(ILiveStreamTranscoder liveStreamTranscoder,  
byte[] license)
```

---

**unregisterLiveStreamTranscoder**

```
public void unregisterLiveStreamTranscoder(ILiveStreamTranscoder liveStreamTranscoder,  
byte[] license)
```

---

**getProtocolUsage**

```
public boolean[] getProtocolUsage()
```

(continued from last page)

**getProtocolUsageTracker**

```
public Server.ProtocolUsageSessionTracker getProtocolUsageTracker()
```

---

**isAcceptWOWZConnections**

```
public boolean isAcceptWOWZConnections()
```

---

**setAcceptWOWZConnections**

```
public void setAcceptWOWZConnections(boolean acceptWOWZConnections)
```

---

**isInitiateWOWZConnections**

```
public boolean isInitiateWOWZConnections()
```

---

**setInitiateWOWZConnections**

```
public void setInitiateWOWZConnections(boolean initiateWOWZConnections)
```

---

**setRestServerEnabled**

```
public void setRestServerEnabled(boolean restServerEnabled)
```

---

**isRestServerEnabled**

```
public boolean isRestServerEnabled()
```

---

**getLiveStreamTranscoderSessionCount**

```
public long getLiveStreamTranscoderSessionCount()
```

---

**getTuningHandler**

```
public ServerTuningHandler getTuningHandler()
```

---

**getTranscoderLicenseInUse**

```
public long getTranscoderLicenseInUse()
```

(continued from last page)

---

**getTranscoderLicenseTotal**

```
public long getTranscoderLicenseTotal()
```

---

**isTranscoderWatermark**

```
public boolean isTranscoderWatermark()
```

---

**isDVRAvailable**

```
public boolean isDVRAvailable()
```

---

**isDVRInUse**

```
public boolean isDVRInUse()
```

---

**isHTTPOriginAvailable**

```
public boolean isHTTPOriginAvailable()
```

---

**isPushPublishAvailable**

```
public boolean isPushPublishAvailable()
```

---

**isRESTAPIAvailable**

```
public boolean isRESTAPIAvailable()
```

---

**isDRMInUse**

```
public boolean isDRMInUse()
```

---

---

(continued from last page)

**getConnectionsMaximum**

```
public long getConnectionsMaximum()
```

---

**getPublishersMaximum**

```
public long getPublishersMaximum()
```

---

**startStats**

```
public void startStats()
```

---

**stopStats**

```
public void stopStats()
```

---

**getDefaultStreamPrefix**

```
public String getDefaultStreamPrefix()
```

---

**setDefaultStreamPrefix**

```
public void setDefaultStreamPrefix(String defaultStreamPrefix)
```

---

**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
```

---

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="#">SHAREDOBJECT_CMD_CONNECT</a> shared object command: connect Value: <b>1</b>
public static final	<a href="#">SHAREDOBJECT_CMD_CONNECTSUCCESS</a> shared object command: clear Value: <b>11</b>
public static final	<a href="#">SHAREDOBJECT_CMD_DELETE</a> shared object command: delete Value: <b>10</b>
public static final	<a href="#">SHAREDOBJECT_CMD_DISCONNECT</a> shared object command: disconnect Value: <b>2</b>
public static final	<a href="#">SHAREDOBJECT_CMD_ERROR</a> shared object command: error Value: <b>7</b>
public static final	<a href="#">SHAREDOBJECT_CMD_SEND</a> shared object command: send Value: <b>6</b>
public static final	<a href="#">SHAREDOBJECT_CMD_SETVALUE</a> shared object command: setvalue Value: <b>3</b>
public static final	<a href="#">SHAREDOBJECT_STATUS_CHANGE</a> shared object status: change Value: <b>4</b>
public static final	<a href="#">SHAREDOBJECT_STATUS_CLEAR</a> shared object status: clear Value: <b>8</b>

<code>public static final</code>	<a href="#"><u>SHAREDOBJECT_STATUS_DELETE</u></a> shared object status: delete Value: <b>9</b>
<code>public static final</code>	<a href="#"><u>SHAREDOBJECT_STATUS_SUCCESS</u></a> shared object status: success Value: <b>5</b>

## Method Summary

<code>void</code>	<a href="#"><u>acquire()</u></a> Increment the reference count to this shared object.
<code>void</code>	<a href="#"><u>addClient(IClient client)</u></a> Add a client to this shared object.
<code>void</code>	<a href="#"><u>addSlotListener(ISharedObjectSlotNotify slotListener)</u></a> Add a slot listener.
<code>void</code>	<a href="#"><u>clear()</u></a> Clear all properties from a shared object
<code>void</code>	<a href="#"><u>close()</u></a> Force close this shared object (not implemented)
<code>boolean</code>	<a href="#"><u>containsProperty(String slotName)</u></a> Returns true if slot/property name exists
<code>boolean</code>	<a href="#"><u>containsSlot(String slotName)</u></a> Returns true if slot/property name exists
<code>void</code>	<a href="#"><u>deleteSlot(IClient client, String slotName)</u></a> Remove a slot (property)
<code>void</code>	<a href="#"><u>deleteSlot(String slotName)</u></a> Remove slot (property)
<code>void</code>	<a href="#"><u>disconnect(IClient client)</u></a> Disconnect client from shared object.
<code>void</code>	<a href="#"><u>flush()</u></a> Flush (write to disk) shared object
<code>java.util.List</code>	<a href="#"><u>getClients()</u></a> Get a list of clients that are connected to this shared object.
<code>String</code>	<a href="#"><u>getName()</u></a> Get shared object name
<code>ISharedObjects</code>	<a href="#"><u>getParent()</u></a> Get the shared object container to which this shared object belongs.
<code>AMFData</code>	<a href="#"><u>getProperty(String slotName)</u></a> Get slot (property) value.
<code>int</code>	<a href="#"><u>getRefCount()</u></a> Get the current reference (clients) connected to this shared object.

<a href="#"><code>ISharedObjectSlot</code></a>	<a href="#"><code>getSlot(String name)</code></a> Get ISharedObjectSlot interface to a slot (property) by name
<code>java.util.List</code>	<a href="#"><code>getSlotNames()</code></a> Get a list of slot (property) names
<code>java.util.List</code>	<a href="#"><code>getSlots()</code></a> Get a list of active slots
<code>String</code>	<a href="#"><code>getStorageDir()</code></a> Get path used to store shared object.
<code>int</code>	<a href="#"><code>getVersion()</code></a> Get the internal version number.
<code>boolean</code>	<a href="#"><code>isClient(IClient client)</code></a> Is this client connected to shared object
<code>boolean</code>	<a href="#"><code>isPersistent()</code></a> Is this shared object being persisted.
<code>void</code>	<a href="#"><code>lock()</code></a> Lock a shared object for write access
<code>int</code>	<a href="#"><code>purge(int version)</code></a> Purge all deleted properties older than the version number
<code>void</code>	<a href="#"><code>putSlot(String name, ISharedObjectSlot slot)</code></a> Add a new slot (property) to a shared object.
<code>void</code>	<a href="#"><code>release()</code></a> Decrement the reference count to this shared object.
<code>void</code>	<a href="#"><code>removeClient(IClient client)</code></a> Remove a client from this shared object.
<code>void</code>	<a href="#"><code>removeSlotListener(ISharedObjectSlotNotify slotListener)</code></a> Remove slot listener
<code>void</code>	<a href="#"><code>send(String handlerName)</code></a> Call client side handler attached to shared object (no parameters).
<code>void</code>	<a href="#"><code>send(String handlerName, Object[] params)</code></a> Call client side handler attached to shared object.
<code>void</code>	<a href="#"><code>setName(String name)</code></a> Set shared object name
<code>void</code>	<a href="#"><code>setPersistent(boolean isPersistent)</code></a> Set is shared object persisted.
<code>void</code>	<a href="#"><code>setProperty(String slotName, AMFData data)</code></a> Set slot (property) value as AMFData object.
<code>void</code>	<a href="#"><code>setProperty(String slotName, boolean value)</code></a> Set slot (property) value as a boolean value (will be wrapped in an AMFDataItem object)
<code>void</code>	<a href="#"><code>setProperty(String slotName, java.util.Date value)</code></a> Set slot (property) value as a date value (will be wrapped in an AMFDataItem object)

void	<a href="#"><u>setProperty</u></a> (String slotName, double value) Set slot (property) value as a double value (will be wrapped in an AMFDataItem object)
void	<a href="#"><u>setProperty</u></a> (String slotName, int value) Set slot (property) value as a int value (will be wrapped in an AMFDataItem object)
void	<a href="#"><u>setProperty</u></a> (String slotName, long value) Set slot (property) value as a long value (will be wrapped in an AMFDataItem object)
void	<a href="#"><u>setProperty</u></a> (String slotName, String value) Set slot (property) value as a string value (will be wrapped in an AMFDataItem object)
void	<a href="#"><u>setStorageDir</u></a> (String storageDir) Set path used to store shared object.
void	<a href="#"><u>setVersion</u></a> (int version) Set the internal version number.
int	<a href="#"><u>size</u></a> () Get the number of active slot (properties).
void	<a href="#"><u>unlock</u></a> () Unlock a shared object for write access
void	<a href="#"><u>writeDeleteError</u></a> (IClient client, String soName, boolean isPersistent, String slotName, String errorMsg) Write an delete error message back to the client
void	<a href="#"><u>writeSetValueError</u></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: **rso**

### SHAREDOBJECT\_CMD\_CONNECT

```
public static final byte SHAREDOBJECT_CMD_CONNECT
```

shared object command: connect  
Constant value: **1**

### SHAREDOBJECT\_CMD\_DISCONNECT

```
public static final byte SHAREDOBJECT_CMD_DISCONNECT
```

shared object command: disconnect  
Constant value: **2**

(continued from last page)

## **SHAREDOBJECT\_CMD\_SETVALUE**

```
public static final byte SHAREDOBJECT_CMD_SETVALUE
```

shared object command: setvalue  
Constant value: **3**

---

## **SHAREDOBJECT\_CMD\_SEND**

```
public static final byte SHAREDOBJECT_CMD_SEND
```

shared object command: send  
Constant value: **6**

---

## **SHAREDOBJECT\_CMD\_ERROR**

```
public static final byte SHAREDOBJECT_CMD_ERROR
```

shared object command: error  
Constant value: **7**

---

## **SHAREDOBJECT\_CMD\_DELETE**

```
public static final byte SHAREDOBJECT_CMD_DELETE
```

shared object command: delete  
Constant value: **10**

---

## **SHAREDOBJECT\_CMD\_CONNECTSUCCESS**

```
public static final byte SHAREDOBJECT_CMD_CONNECTSUCCESS
```

shared object command: clear  
Constant value: **11**

---

## **SHAREDOBJECT\_STATUS\_CHANGE**

```
public static final byte SHAREDOBJECT_STATUS_CHANGE
```

shared object status: change  
Constant value: **4**

---

## **SHAREDOBJECT\_STATUS\_SUCCESS**

```
public static final byte SHAREDOBJECT_STATUS_SUCCESS
```

shared object status: success  
Constant value: **5**

---

## **SHAREDOBJECT\_STATUS\_CLEAR**

```
public static final byte SHAREDOBJECT_STATUS_CLEAR
```

shared object status: clear  
Constant value: **8**

---

## **SHAREDOBJECT\_STATUS\_DELETE**

```
public static final byte SHAREDOBJECT_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 internal version number.

**Returns:**

internal 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 if slot/property name exists

**Parameters:**

slotName - slot/property name

**Returns:**

true if slot/property name exists

---

## containsSlot

```
public boolean containsSlot(String slotName)
```

Returns true if slot/property name exists

**Parameters:**

slotName - slot/property name

**Returns:**

true if 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="#"><code>addSharedObjectListener(ISharedObjectNotify sharedObjectListener)</code></a> Add a shared object listener.
void	<a href="#"><code>disconnect(IClient client)</code></a> Disconnect client from all shared objects in list.
boolean	<a href="#"><code>exists(ISharedObject sharedObject)</code></a> Is sharedObject in this list (by shared object reference).
boolean	<a href="#"><code>exists(String objectName)</code></a> Is sharedObject in this list (by name).
void	<a href="#"><code>flush()</code></a> Flush all persistent shared objects to disk.
<a href="#"><code>ISharedObject</code></a>	<a href="#"><code>get(String name)</code></a> Get shared object by name.
<code>java.util.List</code>	<a href="#"><code>getObjectNames()</code></a> Get a list of shared object names.
<a href="#"><code>ISharedObject</code></a>	<a href="#"><code>getOrCreate(String name)</code></a> Get shared object by name if it does not exist create a new shared object with the given name.
String	<a href="#"><code>getStorageDir()</code></a> Get the storage directory for all shared objects in list.
boolean	<a href="#"><code>isPersistent()</code></a> Are shared objects in list persistent.
void	<a href="#"><code>load()</code></a> Load persistent shared objects from file system.
void	<a href="#"><code>put(String objectName, ISharedObject sharedObject)</code></a> Add or replace a shared object.
void	<a href="#"><code>remove(String objectName)</code></a> Remove a shared object.
void	<a href="#"><code>removeClient(IClient client)</code></a> Remove a client from any shared object that it is connected to in this list.
void	<a href="#"><code>removeSharedObjectListener(ISharedObjectNotify sharedObjectListener)</code></a> Remove a shared object listener.

void	<a href="#"><u>setPersistent</u></a> (boolean isPersistent) Set is shared object in list persistent
void	<a href="#"><u>setStorageDir</u></a> (String storageDir) Set the storage directory for all shared objects in list.
int	<a href="#"><u>size()</u></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 is shared object in list persistent

(continued from last page)

**Parameters:**

isPersistent - true is shared objects in list are persistent

---

**removeClient****public void removeClient([IClient](#) client)**

Remove a client from any shared object that it is connected to in this list.

**Parameters:**

client - client

---

**flush****public void flush()**

Flush all persistent shared objects to disk.

---

**disconnect****public void disconnect([IClient](#) client)**

Disconnect client from all shared objects in list.

**Parameters:**

client - client

---

**addSharedObjectListener****public void addSharedObjectListener([ISharedObjectNotify](#) sharedObjectListener)**

Add a shared object listener. The listener will receive the following events: onSharedObjectCreate, onSharedObjectDestroy, onSharedObjectConnect, onSharedObjectDisconnect.

**Parameters:**

sharedObjectListener

---

**removeSharedObjectListener****public void removeSharedObjectListener([ISharedObjectNotify](#) sharedObjectListener)**

Remove a shared object listener.

**Parameters:**

sharedObjectListener

---

**getStorageDir****public String getStorageDir()**

Get the storage directory for all shared objects in list.

**Returns:**

storage dir

---

**setStorageDir****public void setStorageDir(String storageDir)**

(continued from last page)

Set the storage directory for all shared objects in list.

**Parameters:**

storageDir

---

**load**

**public void load()**

Load persistent shared objects from file system.

## com.wowza.wms.sharedobject Interface ISharedObjectSlot

public interface ISharedObjectSlot  
extends

ISharedObjectSlot: public interface to SharedObjectSlot class.

### Method Summary

<a href="#">AMFData</a>	<a href="#"><code>getData()</code></a> Get slot data as AMFData object.
int	<a href="#"><code>getLastClientId()</code></a> Get client id of client that performed last operation on slot.
int	<a href="#"><code>getLastOperation()</code></a> Get last slot (property) operation.
String	<a href="#"><code>getName()</code></a> Get slot (property) name.
int	<a href="#"><code>getSlotVersion()</code></a> Get slot version
int	<a href="#"><code>getSoVersion()</code></a> Get parent shared object version
void	<a href="#"><code>incSlotVersion()</code></a> Increment slot version by 1.
void	<a href="#"><code>init(String name, AMFData data, int slotVersion)</code></a> Initialize shared object slot
void	<a href="#"><code>setData(AMFData data)</code></a> Set slot data as AMFData object.
void	<a href="#"><code>setData(byte[] data)</code></a> Set slot data as byte[].
void	<a href="#"><code>setData(byte[] data, AMFDataContextDeserialize context)</code></a> Set slot data as byte[].
void	<a href="#"><code>setLastClientId(int lastClientId)</code></a> Set client id of client that performed last operation on slot.
void	<a href="#"><code>setLastOperation(int lastOperation)</code></a> Set last slot (property) operation.
void	<a href="#"><code>setName(String name)</code></a> Set slot (property) name
void	<a href="#"><code>setSlotVersion(int slotVersion)</code></a> Set slot version

void	<a href="#"><u>setSoVersion</u></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="#"><u>onSlotDelete</u></a> ( <a href="#"><u>ISharedObject</u></a> sharedObject, <a href="#"><u>ISharedObjectSlot</u></a> slot) Triggered when sharedObject slot value deleted
void	<a href="#"><u>onSlotSetValue</u></a> ( <a href="#"><u>ISharedObject</u></a> sharedObject, <a href="#"><u>ISharedObjectSlot</u></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: 0
public static final	<a href="#">WRITEACCESS</a>
	Value: 1

#### Fields inherited from interface [com.wowza.wms.sharedobject.ISharedObject](#)

```
FILEEXTENSION, SHAREDOBJECT_CMD_CONNECT, SHAREDOBJECT_CMD_CONNECTSUCCESS,
SHAREDOBJECT_CMD_DELETE, SHAREDOBJECT_CMD_DISCONNECT, SHAREDOBJECT_CMD_ERROR,
SHAREDOBJECT_CMD_SEND, SHAREDOBJECT_CMD_SETVALUE, SHAREDOBJECT_STATUS_CHANGE,
SHAREDOBJECT_STATUS_CLEAR, SHAREDOBJECT_STATUS_DELETE, SHAREDOBJECT_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="#">containsProperty(String slotName)</a>
boolean	<a href="#">containsSlot(String slotName)</a>
void	<a href="#">deleteSlot(IClient client, String slotName)</a>
void	<a href="#">deleteSlot(String slotName)</a>
void	<a href="#">disconnect(IClient client)</a>
void	<a href="#">flush()</a>
static boolean[]	<a href="#">getAccess(IClient client, String soName)</a>
java.util.List	<a href="#">getClients()</a>
void	<a href="#">getClientUpdates(IClient client)</a>
String	<a href="#">getName()</a>
<a href="#">ISharedObjects</a>	<a href="#">getParent()</a>
<a href="#">AMFData</a>	<a href="#">getProperty(String slotName)</a>
int	<a href="#">getRefCount()</a>
<a href="#">ISharedObjectSlot</a>	<a href="#">getSlot(String name)</a>
java.util.List	<a href="#">getSlotNames()</a>
java.util.List	<a href="#">getSlots()</a>
String	<a href="#">getStorageDir()</a>
int	<a href="#">getVersion()</a>
boolean	<a href="#">isClient(IClient client)</a>
boolean	<a href="#">isPersistent()</a>
void	<a href="#">load()</a>
void	<a href="#">lock()</a>
void	<a href="#">notifySlotDelete(ISharedObjectSlot slot)</a>
void	<a href="#">notifySlotSetValue(ISharedObjectSlot slot)</a>

int	<a href="#">purge(int version)</a>
void	<a href="#">putSlot(String name, ISharedObjectSlot slot)</a>
void	<a href="#">release()</a>
void	<a href="#">removeClient(IClient client)</a>
void	<a href="#">removeSlotListener(ISharedObjectSlotNotify slotListener)</a>
void	<a href="#">send(String handlerName)</a>
void	<a href="#">send(String handlerName, Object[] params)</a>
void	<a href="#">sendInternal(IClient client, String handlerName, byte[] msgBytes)</a>
void	<a href="#">sendInternal(IClient client, String handlerName, byte[] msgBytes, AMFDataContextDeserialize context)</a>
void	<a href="#">sendInternal(String handlerName, AMFData[] params)</a>
void	<a href="#">setName(String name)</a>
void	<a href="#">setParent(ISharedObjects parent)</a>
void	<a href="#">setPersistent(boolean isPersistent)</a>
void	<a href="#">setProperty(String slotName, AMFData data)</a>
void	<a href="#">setProperty(String slotName, boolean value)</a>
void	<a href="#">setProperty(String slotName, java.util.Date value)</a>
void	<a href="#">setProperty(String slotName, double value)</a>
void	<a href="#">setProperty(String slotName, int value)</a>
void	<a href="#">setProperty(String slotName, long value)</a>
void	<a href="#">setProperty(String slotName, String value)</a>
void	<a href="#">setSlotValue(IClient client, String slotName, AMFData amfData)</a>
void	<a href="#">setSlotValue(IClient client, String slotName, byte[] byteData)</a>
void	<a href="#">setSlotValue(IClient client, String slotName, byte[] byteData, AMFData amfData)</a>

void	<a href="#">setSlotValue</a> (IClient client, String slotName, byte[] byteData, <a href="#">AMFData</a> , <a href="#">AMFDataContextDeserialize</a> context)
void	<a href="#">setSlotValue</a> (IClient client, String slotName, byte[] byteData, <a href="#">AMFDataContextDeserialize</a> context)
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</a> (IClient client, String soName, boolean isPersistent, String slotName, String errorMsg)
static void	<a href="#">writeError</a> (IClient client, String soName, boolean isPersistent, String errorMsg, boolean isConnect)
void	<a href="#">writeSetValueError</a> (IClient client, String soName, boolean isPersistent, String slotName, String errorMsg)

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

#### Methods inherited from interface com.wowza.wms.sharedobject.ISharedObject

[acquire](#), [addClient](#), [addSlotListener](#), [clear](#), [close](#), [containsProperty](#), [containsSlot](#), [deleteSlot](#), [deleteSlot](#), [disconnect](#), [flush](#), [getClients](#), [getName](#), [getParent](#), [getProperty](#), [getRefCount](#), [getSlot](#), [getSlotNames](#), [getSlots](#), [getStorageDir](#), [getVersion](#), [isClient](#), [isPersistent](#), [lock](#), [purge](#), [putSlot](#), [release](#), [removeClient](#), [removeSlotListener](#), [send](#), [send](#), [setName](#), [setPersistent](#), [setProperty](#), [setProperty](#), [setProperty](#), [setProperty](#), [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="#">FastPlaySettings()</a> Create empty object
public	<a href="#">FastPlaySettings(double multiplier, int fps, int direction)</a> Create object

### Method Summary

int	<a href="#">getDirection()</a> Get direction (1 forward, -1 reverse)
int	<a href="#">getFps()</a> Get frames per second
double	<a href="#">getMultiplier()</a> Get speed of fast play
long	<a href="#">getStartTC()</a> Get the timecode (milliseconds) where this fast play started
long	<a href="#">getStartTCOffset()</a> Get the timecode (milliseconds) where this fast play started (not sure why we have both values)
void	<a href="#">setDirection(int direction)</a> Set direction (1 forward, -1 reverse)
void	<a href="#">setFps(int fps)</a> Set frames per second
void	<a href="#">setMultiplier(double multiplier)</a> Set speed of fast play
void	<a href="#">setStartTC(long startTC)</a> Set the timecode (milliseconds) where this fast play started
void	<a href="#">setStartTCOffset(long startTCoffset)</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,
```

## 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="#"><b>clone()</b></a> Make a clone copy of media index
String	<a href="#"><b>getChecksum()</b></a> Get checksum for media index
int	<a href="#"><b>getExtent()</b></a> Get the duration (milliseconds) of media index
long	<a href="#"><b>getLastAccessed()</b></a> Get last time media index was accessed (milliseconds)
boolean	<a href="#"><b>isHitEnd()</b></a> Is media index complete
void	<a href="#"><b>setChecksum(<i>IRandomAccessReader</i> file)</b></a> Set media index checksum.
void	<a href="#"><b>setChecksum(String checksum)</b></a> Set media index checksum
void	<a href="#"><b>setLastAccessed(long time)</b></a> Set last time media index accessed (milliseconds)
void	<a href="#"><b>touch()</b></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="#"><u>MediaList</u></a>	<a href="#"><u>resolveMediaList</u></a> ( <a href="#"><u>IMediaListReader</u></a> mediaListReader, <a href="#"><u>IMediaStream</u></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="#"><u>close()</u></a> close file
<a href="#"><u>IHTTPStreamerSession</u></a>	<a href="#"><u>getHTTPStreamerSession()</u></a> Get the HTTPStreamerSession associated with this media list reader
String	<a href="#"><u>getMediaExtension()</u></a> Get media extension
<a href="#"><u>MediaList</u></a>	<a href="#"><u>getMediaList()</u></a> Get the MediaList object
String	<a href="#"><u>getPath()</u></a> Get abstract path to the media item
void	<a href="#"><u>init(IApplicationInstance appInstance, IMediaStream stream, String mediaReadType, String basePath, String mediaName, IHTTPStreamerSession httpStreamerSession)</u></a> Initialize mediaReader
boolean	<a href="#"><u>isOpen()</u></a> is file open
void	<a href="#"><u>open(String basePath, String name)</u></a> Open the file for reading
void	<a href="#"><u>setHTTPStreamerSession(IHTTPStreamerSession httpStreamerSession)</u></a> Set the HTTPStreamerSession associated with this media list reader
void	<a href="#"><u>setMediaReaderItem(MediaReaderItem mediaReaderItem)</u></a> Set the mediaReader item definition
void	<a href="#"><u>setProperties(WMSProperties properties)</u></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(WMSProperties 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="#">CONTENTTYPE_MEDIA</a>
	Value: <b>1</b>
public static final	<a href="#">CONTENTTYPE_MEDIALIST</a>
	Value: <b>2</b>
public static final	<a href="#">DEFAULT_RANDOMACCESSREADER</a>
	Value: <b>com.wowza.io.DirectRandomAccessReader</b>
public static final	<a href="#">DEFAULT_RANDOMACCESSREADEROPTIMIZER</a>
	Value: <b>com.wowza.io.RandomAccessReadOptimizer</b>
public static final	<a href="#">PLAYEVENT_AFTERBUFFERFILL</a>
	Value: <b>5</b>
public static final	<a href="#">PLAYEVENT_AFTERMETADATA</a>
	Value: <b>3</b>
public static final	<a href="#">PLAYEVENT_BEFOREBUFFERFILL</a>
	Value: <b>4</b>
public static final	<a href="#">PLAYEVENT_BEFOREMETADATA</a>
	Value: <b>2</b>
public static final	<a href="#">PLAYEVENT_STARTPLAYBACK</a>
	Value: <b>1</b>
public static final	<a href="#">SEEK_EXACT</a>
	Seek direction: closest frame (audio, video) (key, no-key)
	Value: <b>4</b>
public static final	<a href="#">SEEK_KEYCLOSE</a>
	Seek direction: closest key frame
	Value: <b>3</b>
public static final	<a href="#">SEEK_KEYDOWN</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 href="#">IApplicationInstance</a> appInstance, <a href="#">IMediaStream</a> stream, String mediaReadType, String basePath, String mediaName)</a> Initialize mediaReader
boolean	<a href="#">isOpen()</a> is file open
void	<a href="#">open(String basePath, String name)</a> Open the file for reading
void	<a href="#">rewind()</a> rewind file to start
PlaylistSeekResult	<a href="#">seek(long timecode, int seektype)</a> seek to timecode in file
int	<a href="#">sendZeroLengthPacket(int type, int timecode, boolean isAbsolute, <a href="#">IMediaStream</a> stream, java.io.OutputStream out, <a href="#">AMFObj</a> wmsObj, <a href="#">PlaylistCursor</a> flvCursor, PlaylistWriteControl control, PlaylistReaderWriteResults results, long[] sizes, boolean isForceTCZero)</a>

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
void	<a href="#"><code>setStreamPosition(IMediaReaderStreamPosition pos)</code></a> 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(IMediaStream stream, java.io.OutputStream out, AMFObj wmsObjAudio, AMFObj wmsObjVideo, AMFObj wmsObjData, PlaylistCursor flvCursor, PlaylistWriteControl control, PlaylistReaderWriteResults results, long[] sizes, FastPlaySettings fastPlaySettings, boolean isForceTCZero)</code></a> From current location in file generate a key frame (enhanced seek) and write it to out
int	<a href="#"><code>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)</code></a> From current location in file write packets to output
int	<a href="#"><code>writePackets(java.util.List packetList, PlaylistCursor flvCursor, PlaylistWriteControl control, PlaylistReaderWriteResults results, long[] sizes, FastPlaySettings fastPlaySettings)</code></a> Write packets to the packetList as AMFPackets

## Fields

### CONTENTTYPE\_MEDIA

```
public static final int CONTENTTYPE_MEDIA
```

Constant value: 1

### CONTENTTYPE\_MEDIALIST

```
public static final int CONTENTTYPE_MEDIALIST
```

Constant value: 2

### DEFAULT\_RANDOMACCESSREADER

```
public static final java.lang.String DEFAULT_RANDOMACCESSREADER
```

Constant value: `com.wowza.io.DirectRandomAccessReader`

### DEFAULT\_RANDOMACCESSREADEROPTIMIZER

```
public static final java.lang.String DEFAULT_RANDOMACCESSREADEROPTIMIZER
```

Constant value: `com.wowza.io.RandomAccessReadOptimizer`

**SEEKTARGET\_VIDEOKEYFRAME**

```
public static final int SEEKTARGET_VIDEOKEYFRAME
```

Constant value: 1

---

**SEEKTARGET\_AUDIO**

```
public static final int SEEKTARGET_AUDIO
```

Constant value: 3

---

**SEEKTARGET\_ENHANCED**

```
public static final int SEEKTARGET_ENHANCED
```

Constant value: 4

---

**PLAYEVENT\_STARTPLAYBACK**

```
public static final int PLAYEVENT_STARTPLAYBACK
```

Constant value: 1

---

**PLAYEVENT\_BEFOREMETADATA**

```
public static final int PLAYEVENT_BEFOREMETADATA
```

Constant value: 2

---

**PLAYEVENT\_AFTERMETADATA**

```
public static final int PLAYEVENT_AFTERMETADATA
```

Constant value: 3

---

**PLAYEVENT\_BEFOREBUFFERFILL**

```
public static final int PLAYEVENT_BEFOREBUFFERFILL
```

Constant value: 4

---

**PLAYEVENT\_AFTERBUFFERFILL**

```
public static final int PLAYEVENT_AFTERBUFFERFILL
```

Constant value: 5

---

(continued from last page)

## SEEK\_KEYUP

```
public static final int SEEK_KEYUP
```

Seek direction: up to closets key frame  
Constant value: **1**

## SEEK\_KEYDOWN

```
public static final int SEEK_KEYDOWN
```

Seek direction: down to closets key frame  
Constant value: **2**

## SEEK\_KEYCLOSE

```
public static final int SEEK_KEYCLOSE
```

Seek direction: closest key frame  
Constant value: **3**

## SEEK\_EXACT

```
public static final int SEEK_EXACT
```

Seek direction: closest frame (audio, video) (key, no-key)  
Constant value: **4**

## Methods

### init

```
public void init(IApplicationInstance appInstance,  
    IMediaStream stream,  
    String mediaReadType,  
    String basePath,  
    String mediaName)
```

Initialize mediaReader

#### Parameters:

stream - parent stream  
mediaReadType - media reader type id. Example: flv  
basePath - Base path for application instance  
mediaName - name of the media item

### setMediaReaderItem

```
public void setMediaReaderItem(MediaReaderItem mediaReaderItem)
```

Set the mediaReader item definition

#### Parameters:

mediaReaderItem

### open

```
public void open(String basePath,  
    String name)
```

(continued from last page)

Open the file for reading

**Parameters:**

basePath - base path to file  
name - file name without extension

**getMetadata**

```
public java.util.List getMetadata()
```

Get a collection of metadata packets in ByteBuffers for this file. You can use 'new AMFDataList(data)' to convert to AMF objects.

**Returns:**

collection of metadata packets in ByteBuffers

**rewind**

```
public void rewind()
```

rewind file to start

**close**

```
public void close()
```

close file

**isOpen**

```
public boolean isOpen()
```

is file open

**Returns:**

is file open

**seek**

```
public PlaylistSeekResult seek(long timecode,  
                           int seektype)
```

seek to timecode in file

**Parameters:**

timecode - timecode (milliseconds) to seek to  
seektype - seek type IMediaReader.SEEK\_\*

**Returns:**

detailed results object of seek or null if failure

(continued from last page)

## writePackets

```
public int writePackets(IMediaStream stream,
                      java.io.OutputStream out,
                      AMFObj wmsObjAudio,
                      AMFObj wmsObjVideo,
                      AMFObj wmsObjData,
                      PlaylistCursor flvCursor,
                      PlaylistWriteControl control,
                      PlaylistReaderWriteResults results,
                      long[] sizes,
                      FastPlaySettings fastPlaySettings,
                      boolean isForceTCZero)
```

From current location in file write packets to output

**Parameters:**

stream - parent mediaStream  
 out - output stream  
 wmsObjAudio - audio channel amfObj  
 wmsObjVideo - video channel amfObj  
 wmsObjData - data channel amfObj  
 flvCursor - current cursor that keeps track of timecodes and flags indicating position in media file  
 control - write control flags. Mainly used to control when it should stop writing  
 results - write results returned from this routine  
 sizes - array of values that keep track of number of bytes written and number of packets for each data type (audio, video, data). See IMediaStreamPlay.PLAYSIZES\_\*  
 fastPlaySettings - current fastPlay settings for the stream  
 isForceTCZero - are we forcing all timecodes to zero

**Returns:**

number of bytes written

---

## writePackets

```
public int writePackets(java.util.List packetList,
                       PlaylistCursor flvCursor,
                       PlaylistWriteControl control,
                       PlaylistReaderWriteResults results,
                       long[] sizes,
                       FastPlaySettings fastPlaySettings)
```

Write packets to the packetList as AMFPackets

**Parameters:**

packetList - list to which to write packets  
 flvCursor - current cursor that keeps track of timecodes and flags indicating position in media file  
 control - write control flags. Mainly used to control when it should stop writing  
 results - write results returned from this routine  
 sizes - array of values that keep track of number of bytes written and number of packets for each data type (audio, video, data). See IMediaStreamPlay.PLAYSIZES\_\*  
 fastPlaySettings - current fastPlay settings for the stream

**Returns:**

number of bytes written

---

(continued from last page)

## writeGeneratedKeyFrame

```
public int writeGeneratedKeyFrame(IMediaStream stream,
    java.io.OutputStream out,
    AMFObj wmsObjAudio,
    AMFObj wmsObjVideo,
    AMFObj wmsObjData,
    PlaylistCursor flvCursor,
    PlaylistWriteControl control,
    PlaylistReaderWriteResults results,
    long[] sizes,
    FastPlaySettings fastPlaySettings,
    boolean isForceTCZero)
```

From current location in file generate a key frame (enhanced seek) and write it to out

**Parameters:**

stream - parent mediaStream  
 out - output stream  
 wmsObjAudio - audio channel amfObj  
 wmsObjVideo - video channel amfObj  
 wmsObjData - data channel amfObj  
 flvCursor - current cursor that keeps track of timecodes and flags indicating position in media file  
 control - write control flags. Mainly used to control when it should stop writing  
 results - write results returned from this routine  
 sizes - array of values that keep track of number of bytes written and number of packets for each data type (audio, video, data). See IMediaStreamPlay.PLAYSIZES\_\*  
 fastPlaySettings - current fastPlay settings for the stream  
 isForceTCZero - are we forcing all timecodes to zero

**Returns:**

number of bytes written

## sendZeroLengthPacket

```
public int sendZeroLengthPacket(int type,
    int timecode,
    boolean isAbsolute,
    IMediaStream stream,
    java.io.OutputStream out,
    AMFObj wmsObj,
    PlaylistCursor flvCursor,
    PlaylistWriteControl control,
    PlaylistReaderWriteResults results,
    long[] sizes,
    boolean isForceTCZero)
```

## startPlayback

```
public void startPlayback()
```

Called each time the player begins playback (before the buffer is filled)

## getDuration

```
public long getDuration()
```

Get duration or time (milliseconds) of the media file

**Returns:**

(continued from last page)

duration or time (milliseconds) of the media file

---

## getLength

```
public long getLength()
```

Get the stream length in bytes

**Returns:**

stream length in bytes

---

## getPath

```
public String getPath()
```

Get abstract path to the media item

**Returns:**

abstract path to the media item

---

## getMediaExtension

```
public String getMediaExtension()
```

Get media extension

**Returns:**

media extension

---

## getStreamPosition

```
public IMediaReaderStreamPosition getStreamPosition()
```

Get a reference to the current stream position

**Returns:**

stream position

---

## setStreamPosition

```
public void setStreamPosition(IMediaReaderStreamPosition pos)
```

Set the file position within the media file

**Parameters:**

pos - stream position

---

## setProperties

```
public void setProperties(WMSProperties properties)
```

Set the properties for this media reader

**Parameters:**

properties - properties

## com.wowza.wms.stream Interface IMediaReaderActionNotify

public interface **IMediaReaderActionNotify**  
extends

IMediaReaderActionNotify: listener interface for IMediaReader actions: See  
IAplicationInstance.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 extracted 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 IMediaReaderEncInfo**

---

public interface **IMediaReaderEncInfo**  
extends

---

## Method Summary

<a href="#">MediaReaderEncInfo</a>	<a href="#">getEncInfo()</a>
------------------------------------	------------------------------

---

## Methods

### getEncInfo

public [MediaReaderEncInfo](#) **getEncInfo()**

## com.wowza.wms.stream Interface IMediaReaderStreamPosition

public interface **IMediaReaderStreamPosition**  
extends

IMediaReaderStreamPosition: Internal use

### Method Summary

boolean	<a href="#"><u>isValid()</u></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="#">addAudioExtraData</a> (java.util.Map extraData) Add extra data to the current audio packet
void	<a href="#">addCallbackListener</a> ( <a href="#">IMediaStreamCallback</a> callbackListener) Add a callback listener to listen for all stream metadata
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="#"><u>addDataExtraData</u></a> (java.util.Map extraData) Add extra data to the current data packet
void	<a href="#"><u>addLivePacketListener</u></a> (IMediaStreamLivePacketNotify listener) Add a listener to receiver notification of new live AMF packets
void	<a href="#"><u>addVideoCodecConfigPacket</u></a> (long timecode, <u>AMFPacket</u> packet) Set video codec configuration packet (needed for H.264/AAC playback)
void	<a href="#"><u>addVideoData</u></a> (byte[] data, int offset, int size) Add data to current video packet
void	<a href="#"><u>addVideoExtraData</u></a> (java.util.Map extraData) Add extra data to the current video packet
void	<a href="#"><u>addVideoH264SEIListener</u></a> (IMediaStreamH264SEINotify 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> (IClient client, String name) Get the read/write access to this stream for this client
<u>AMFPacket</u>	<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
<u>IClient</u>	<a href="#"><u>getClient</u></a> () Get parent client connection

int	<a href="#">getclientId()</a> Get parent client connection (id)
String	<a href="#">getContextStr()</a> Returns the stream context string in the form [application]/[appInstance]/[streamName].
int	<a href="#">getDataMissing()</a> Get number of data bytes missing from current audio packet
int	<a href="#">getDataSize()</a> Get the size of the current data packet that is being streamed from the client to the server
long	<a href="#">getDataTC()</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="#">getExtension()</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="#">getMaxTimecode()</a> Get the timecode of the latest received packet
<a href="#">IOPerformanceCounter</a>	<a href="#">getMediaIOPerformance()</a> Get IO performance counter
<a href="#">IMediaStreamMetaDataProvider</a>	<a href="#">getMetaDataProvider()</a> Get the metaData provider
String	<a href="#">getName()</a> Get stream name
com.wowza.wms.netconnection.INetConnection	<a href="#">getNetConnection()</a> Get parent netConnection (future server to server communication)
<a href="#">IMediaStreamPlay</a>	<a href="#">getPlayer()</a> Get underlying player (IMediaStreamPlay) object
java.util.List	<a href="#">getPlayPackets()</a> Get all available live packets
<a href="#">WMSProperties</a>	<a href="#">getProperties()</a> Get mediaStream properties
int	<a href="#">getPublishAudioCodecId()</a> Get the codec id of the most recently published audio packet
int	<a href="#">getPublishVideoCodecId()</a> Get the codec id of the most recently published video packet
String	<a href="#">getQueryStr()</a> Get play/publish name query string.
int	<a href="#">getReceiveVideoFPS()</a> Set frame per seconds for video (not currently implemented)
<a href="#">AMFObj</a>	<a href="#">getRespAMFAudioObj()</a> Get audio response channel object
<a href="#">AMFObj</a>	<a href="#">getRespAMFDataObj()</a> Get data response channel object

<a href="#">AMFObj</a>	<a href="#"><code>getRespAMFVideoObj()</code></a> Get video response channel object
<a href="#">RTPStream</a>	<a href="#"><code>getRTPStream()</code></a> Get the RTP based stream this stream is associated with
int	<a href="#"><code>getSrc()</code></a> Get the stream id
<a href="#">java.io.File</a>	<a href="#"><code>getStreamFileForRead()</code></a> Get the File object to read from a stream (get stream name, ext and query from stream object)
<a href="#">java.io.File</a>	<a href="#"><code>getStreamFileForRead(String name, String ext, String query)</code></a> Get the File object to read from a stream (specify name, ext and query)
<a href="#">java.io.File</a>	<a href="#"><code>getStreamFileForWrite()</code></a> Get the File object to write to a stream (get stream name, ext and query from stream object)
<a href="#">java.io.File</a>	<a href="#"><code>getStreamFileForWrite(String name, String ext, String query)</code></a> Get the File object to write to a stream (specify name, ext and query)
<a href="#">MediaStreamMap</a>	<a href="#"><code>getStreams()</code></a> Get parent mediaStreamMap (owned by applicationInstance)
<a href="#">String</a>	<a href="#"><code>getStreamType()</code></a> Get mediaStream streamType
<a href="#">String</a>	<a href="#"><code>getUniqueStreamIdStr()</code></a> Get a string that uniquely identifies this stream
<a href="#">AMFPacket</a>	<a href="#"><code>getVideoCodecConfigPacket(long timecode)</code></a> Get video codec configuration packet (needed for H.264/AAC playback)
int	<a href="#"><code>getVideoMissing()</code></a> Get number of video bytes missing from current audio packet
int	<a href="#"><code>getVideoSize()</code></a> Get the size of the current video packet that is being streamed from the client to the server
long	<a href="#"><code>getVideoTC()</code></a> Get last absolute video timecode (milliseconds) sent to mediaStream
void	<a href="#"><code>handleCallback(com.wowza.wms.request.RequestFunction function)</code></a> Routes request function to callback handler onStatus, onPlayStatus or [method/handler]
boolean	<a href="#"><code>idle()</code></a> Allow the mediaStream to perform idle work (not currently in use)
long	<a href="#"><code>incrementMediaInBytes(long increment)</code></a> Increment the number of mediaStream bytes received
long	<a href="#"><code>incrementMediaLossBytes(long bytes, long count)</code></a> Increment the number of mediaStream loss bytes sent and number of packets sent
long	<a href="#"><code>incrementMediaOutBytes(long bytes, long count)</code></a> Increment the number of mediaStream bytes sent and number of packets sent
void	<a href="#"><code>init(MediaStreamMap parent, int src, WMSProperties properties)</code></a> 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>isPublisherStream</u></a> () Returns true if stream was created by stream publisher API.
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> (AMFPacket 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>notifyLivePacket</u></a> (AMFPacket packet) Notify new live AMF packet
void	<a href="#"><u>notifyVideoH264Packet</u></a> (AMFPacket packet, com.wowza.wms.media.h264.H264SEIMessages 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="#"><code>registerOnStatus(IMediaStreamCallback callback)</code></a> Register onStatus handler
void	<a href="#"><code>removeCalbackListener(IMediaStreamCallback callbackListener)</code></a> Remove a callback listener to listen for all stream metadata
void	<a href="#"><code>removeClientListener(IMediaStreamActionNotify actionListener)</code></a> Remove client listener.
void	<a href="#"><code>removeClientListener(IMediaStreamActionNotify2 actionListener)</code></a> Remove client listener.
void	<a href="#"><code>removeClientListener(IMediaStreamActionNotify3 actionListener)</code></a> Remove client listener.
<a href="#"><u>ILiveStreamDvrRecorder</u></a>	<a href="#"><code>removeDvrRecorder(String name)</code></a> Remove a live stream dvr by name
void	<a href="#"><code>removeLivePacketListener(IMediaStreamLivePacketNotify listener)</code></a> Remove a listener to receiver notification of new live AMF packets
<a href="#"><u>ILiveStreamTranscoder</u></a>	<a href="#"><code>removeLiveStreamTranscoder(String name)</code></a> Remove a live stream transcoder by name
void	<a href="#"><code>removeVideoH264SEIListener(IMediaStreamH264SEINotify h264SEIListener)</code></a> Remove an H.264 SEI listener.
void	<a href="#"><code>send(String handlerName)</code></a> Call client side NetStream method/handler with no parameters
void	<a href="#"><code>send(String handlerName, Object[] params)</code></a> Call client side NetStream method/handler
void	<a href="#"><code>sendAMF3(String handlerName)</code></a> Call client side NetStream method/handler with no parameters.
void	<a href="#"><code>sendAMF3(String handlerName, Object[] params)</code></a> Call client side NetStream method/handler.
int	<a href="#"><code>sendControlBytes(int controlType, java.io.OutputStream out)</code></a> Send playback control bytes.
void	<a href="#"><code>sendDirect(String handlerName)</code></a> Call client side NetStream method/handler and send event to underlying stream (will record event)
void	<a href="#"><code>sendDirect(String handlerName, Object[] params)</code></a> Call client side NetStream method/handler and send event to underlying stream (will record event)
void	<a href="#"><code>sendDirectAMF3(String handlerName)</code></a> Call client side NetStream method/handler and send event to underlying stream (will record event).
void	<a href="#"><code>sendDirectAMF3(String handlerName, Object[] params)</code></a> Call client side NetStream method/handler and send event to underlying stream (will record event).
int	<a href="#"><code>sendLivePlaySeek(java.io.OutputStream out, String name, long timecode)</code></a> Send onStatus(NetStream.Seek.Notify) event

int	<code>sendLivePlayStart(java.io.OutputStream out, String name, long timecode, long timecodeOffset)</code> Send onStatus(NetStream.Play.Start) event
int	<code>sendLivePlaySwitch(java.io.OutputStream out, String name, long timecode)</code> Send onStatus(NetStream.Play.Transition) event
int	<code>sendPauseNotify(long timecode, String name)</code> Send onStatus(NetStream.Pause.Notify) event
int	<code>sendPauseNotify(java.io.OutputStream out, long timecode, String name)</code> Send onStatus(NetStream.Pause.Notify) event.
int	<code>sendPlayReset(java.io.OutputStream out, String name)</code> Send onStatus(NetStream.Play.Reset) event.
int	<code>sendPlayReset(String name)</code> Send onStatus(NetStream.Play.Reset) event
int	<code>sendPlaySeek(long location, long seekLocation, String name)</code> Send onStatus(NetStream.Seek.Notify) event.
int	<code>sendPlaySeek(java.io.OutputStream out, long location, long seekLocation, String name)</code> Send onStatus(NetStream.Seek.Notify) event.
int	<code>sendPlaySeek(java.io.OutputStream out, long location, long seekLocation, String name, java.util.List seekTypes)</code> Send onStatus(NetStream.Seek.Notify) event.
int	<code>sendPlayStart(java.io.OutputStream out, String name, boolean isSwitch, boolean isLive, long timecode, java.util.List seekTypes)</code> Send onStatus(NetStream.Play.Start) event (it is not a typo, this also send play start but with different control codes for media switch).
int	<code>sendPlayStart(java.io.OutputStream out, String name, boolean isSwitch, long timecode)</code> Send onStatus(NetStream.Play.Start) event (it is not a typo, this also send play start but with different control codes for media switch).
int	<code>sendPlayStart(java.io.OutputStream out, String name, boolean isSwitch, long timecode, java.util.List seekTypes)</code> Send onStatus(NetStream.Play.Start) event (it is not a typo, this also send play start but with different control codes for media switch).
int	<code>sendPlayStart(String name, long timecode)</code> Send onStatus(NetStream.Play.Start) event
int	<code>sendPlayStatus(long timecode, int statusType, double duration, double bytesSent)</code> Send onPlayStatus(NetStream.Play.Switch, NetStream.Play.Complete, NetStream.Play.Stop) event
int	<code>sendPlayStatus(java.io.OutputStream out, long timecode, int statusType, double duration, double bytesSent)</code> Send onPlayStatus(NetStream.Play.Switch, NetStream.Play.Complete, NetStream.Play.Stop) event.
int	<code>sendPlayStop(long location, String name)</code> Send onStatus(NetStream.Play.Stop) event

int	<a href="#"><u>sendPlayStop</u></a> (java.io.OutputStream out, long location, String name) Send onStatus(NetStream.Play.Stop) event.
int	<a href="#"><u>sendPlaySwitch</u></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="#"><u>sendPlaySwitch</u></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="#"><u>sendStreamNotFound</u></a> (java.io.OutputStream out, String name) Send onStatus(NetStream.Play.StreamNotFound) event.
int	<a href="#"><u>sendStreamNotFound</u></a> (String name) Send onStatus(NetStream.Play.StreamNotFound) event
int	<a href="#"><u>sendUnpauseNotify</u></a> (long location, String name) Send onStatus(NetStream.Unpause.Notify) event
int	<a href="#"><u>sendUnpauseNotify</u></a> (java.io.OutputStream out, long location, String name) Send onStatus(NetStream.Unpause.Notify) event.
int	<a href="#"><u>sendUnpauseNotify</u></a> (java.io.OutputStream out, long location, String name, java.util.List seekTypes) Send onStatus(NetStream.Unpause.Notify) event
int	<a href="#"><u>sendVODPlaySwitch</u></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> (IClient 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> (FastPlaySettings 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> (IHTTPStreamerSession 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 onMetdata event data.
void	<a href="#"><u>setMetaDataProvider</u></a> (IMediaStreamMetaDataProvider 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> (IMediaStreamPlay 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>setPublisherStream</u></a> (boolean isPublisherStream) Set stream was created by stream publisher.
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> (RTPStream 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="#"><u>switchName</u></a> (String name, String oldName, String ext, String queryStr, double playStart, double playLen, int playTransition) Switch to stream name
void	<a href="#"><u>trim</u></a> () Trim mediaStream.
void	<a href="#"><u>unregisterCallback</u></a> (String handlerName) Unregister a callback handler
void	<a href="#"><u>unregisterOnPlayStatus</u></a> ( <a href="#">IMediaStreamCallback</a> callback) Unregister onPlayStatus handler
void	<a href="#"><u>unregisterOnStatus</u></a> ( <a href="#">IMediaStreamCallback</a> callback) Unregister onStatus handler
void	<a href="#"><u>updateLoggingDuration</u></a> () Update logging.MDC with mediaStream logging information
void	<a href="#"><u>updateLoggingValues</u></a> () Update logging.MDC with mediaStream logging information

## Fields

(continued from last page)

## READACCESS

```
public static final int READACCESS
```

Constant value: 0

## WRITEACCESS

```
public static final int WRITEACCESS
```

Constant value: 1

## AUDIOSAMPLEACCESS

```
public static final int AUDIOSAMPLEACCESS
```

Constant value: 2

## VIDEOSAMPLEACCESS

```
public static final int VIDEOSAMPLEACCESS
```

Constant value: 3

## Methods

### init

```
public void init(MediaStreamMap parent,  
                 int src,  
                 WMSProperties properties)
```

Initialize the mediaStream object after creation. Creation and initialization are separated due to the order of the creation events.

#### Parameters:

parent - parent mediaStreamMap of the parent applicationInstance  
src - stream id for this stream  
properties - initial properties as defined in Streams.xml

### getBufferTime

```
public int getBufferTime()
```

Get buffer time for mediaStream (milliseconds)

#### Returns:

buffer time (milliseconds)

### setBufferTime

```
public void setBufferTime(int bufferTime)
```

Set buffer time for mediaStream (milliseconds)

(continued from last page)

**Parameters:**

bufferTime - buffer time (milliseconds)

**isPlaying****public boolean.isPlaying()**

Is mediaStream playing (or paused - false)

**Returns:**

is playing

**setIsPlaying****public void.setIsPlaying(boolean.isPlaying)**

Set is mediaStream playing

**Parameters:**

isPlaying

**getAudioSize****public int.getAudioSize()**

Get the size of the current audio packet that is being streamed from the client to the server

**Returns:**

size

**getVideoSize****public int.getVideoSize()**

Get the size of the current video packet that is being streamed from the client to the server

**Returns:**

size

**getDataSize****public int.getDataSize()**

Get the size of the current data packet that is being streamed from the client to the server

**Returns:**

size

**setAudioSize****public void.setAudioSize(int.audioSize)**

Set the size of the current audio packet that is being streamed from the client to the server

**Parameters:**

audioSize

(continued from last page)

## setVideoSize

```
public void setVideoSize(int videoSize)
```

Set the size of the current video packet that is being streamed from the client to the server

**Parameters:**

videoSize

---

## setDataSize

```
public void setDataSize(int dataSize)
```

Set the size of the current data packet that is being streamed from the client to the server

**Parameters:**

dataSize

---

## addAudioExtraData

```
public void addAudioExtraData(java.util.Map extraData)
```

Add extra data to the current audio packet

**Parameters:**

extraData - extra data

---

## addVideoExtraData

```
public void addVideoExtraData(java.util.Map extraData)
```

Add extra data to the current video packet

**Parameters:**

extraData - extra data

---

## addDataExtraData

```
public void addDataExtraData(java.util.Map extraData)
```

Add extra data to the current data packet

**Parameters:**

extraData - extra data

---

## getSrc

```
public int getSrc()
```

Get the stream id

**Returns:**

stream id

---

## setSrc

```
public void setSrc(int src)
```

(continued from last page)

Set stream id

**Parameters:**

src - stream id

---

**getName**

```
public String getName()
```

Get stream name

**Returns:**

stream name

---

**setName**

```
public void setName(String name)
```

Set stream name

**Parameters:**

name - stream name

---

**setName**

```
public void setName(String name,  
                   String ext)
```

Set stream name and extension. Example: if play command sent flv:test name=test, ext=flv.

**Parameters:**

name - stream name

ext - stream extension

---

**setName**

```
public void setName(String name,  
                   String oldName,  
                   String ext,  
                   String queryStr,  
                   double playStart,  
                   double playLen,  
                   int playTransition)
```

Set stream name, extension, query string, play start, play len and play reset from play command.

**Parameters:**

name - stream name

oldName - old stream name

ext - stream extension

queryStr - query string

playStart - play start

playLen - play len

playTransition - play transition (see MediaBase.PLAYTRANSITION\_\*)

---

(continued from last page)

## switchName

```
public void switchName(String name,
                      String oldName,
                      String ext,
                      String queryStr,
                      double playStart,
                      double playLen,
                      int playTransition)
```

Switch to stream name

### Parameters:

name - stream name  
oldName - old stream name  
ext - stream extension  
queryStr - query string  
playStart - play start  
playLen - play len  
playTransition - play transition (see MediaBase.PLAYTRANSITION\_\*)

---

## stopName

```
public void stopName(String name,
                     String oldName,
                     String ext,
                     String queryStr,
                     double playStart,
                     double playLen,
                     int playTransition)
```

Stop stream name

### Parameters:

name - stream name  
oldName - old stream name  
ext - stream extension  
queryStr - query string  
playStart - play start  
playLen - play len  
playTransition - play transition (see MediaBase.PLAYTRANSITION\_\*)

---

## shutdown

```
public void shutdown()
```

shutdown or close this mediaStream

---

## getAudioTC

```
public long getAudioTC()
```

Get last absolute audio timecode (milliseconds) sent to mediaStream

### Returns:

last absolute audio timecode (milliseconds)

(continued from last page)

## setAudioTC

```
public void setAudioTC(long audioTC,  
                      boolean isAbsolute)
```

Set last absolute audio timecode (milliseconds) sent to mediaStream

**Parameters:**

audioTC - timecode (milliseconds)  
isAbsolute - is the timecode value relative to last timecode or absolute

---

## setAudioTC

```
public void setAudioTC(long audioTC)
```

Set last absolute audio timecode (milliseconds) sent to mediaStream

**Parameters:**

audioTC - timecode (milliseconds)

---

## getVideoTC

```
public long getVideoTC()
```

Get last absolute video timecode (milliseconds) sent to mediaStream

**Returns:**

last absolute video timecode (milliseconds)

---

## setVideoTC

```
public void setVideoTC(long videoTC,  
                      boolean isAbsolute)
```

Set last absolute video timecode (milliseconds) sent to mediaStream

**Parameters:**

videoTC - timecode (milliseconds)  
isAbsolute - is the timecode value relative to last timecode or absolute

---

## setVideoTC

```
public void setVideoTC(long videoTC)
```

Set last absolute video timecode (milliseconds) sent to mediaStream

**Parameters:**

videoTC - timecode (milliseconds)

---

## getDataTC

```
public long getDataTC()
```

Get last absolute data timecode (milliseconds) sent to mediaStream

**Returns:**

last absolute data timecode (milliseconds)

(continued from last page)

## setDataTC

```
public void setDataTC(long dataTC,  
                      boolean isAbsolute)
```

Set last absolute data timecode (milliseconds) sent to mediaStream

**Parameters:**

dataTC - timecode (milliseconds)

isAbsolute - is the timecode value relative to last timecode or absolute

---

## getDataType

```
public int getDataType()
```

Get the data packet type: (IVHost.CONTENTTYPE\_DATA0 or IVHost.CONTENTTYPE\_DATA3)

**Returns:**

data packet type

---

## setDataType

```
public void setDataType(int dataType)
```

Set the data packet type: (IVHost.CONTENTTYPE\_DATA0 or IVHost.CONTENTTYPE\_DATA3)

**Parameters:**

dataType - data packet type

---

## setDataTC

```
public void setDataTC(long dataTC)
```

Set last absolute data timecode (milliseconds) sent to mediaStream

**Parameters:**

dataTC - timecode (milliseconds)

---

## isRecord

```
public boolean isRecord()
```

Is this stream being recorded to a file

**Returns:**

is stream being recorded

---

## setRecord

```
public void setRecord(boolean isRecord)
```

Set is the stream being recorded

**Parameters:**

isRecord

(continued from last page)

## isPlay

```
public boolean isPlay()
```

Is the stream a play stream (vs a publish stream)

**Returns:**

is play stream (has nothing to do with if its playing just that its play vs publish)

---

## setPlay

```
public void setPlay(boolean isPlay)
```

Set is the stream a play stream (vs a publish stream)

**Parameters:**

*isPlay* - is play stream (has nothing to do with if its playing just that its play vs publish)

---

## idle

```
public boolean idle()
```

Allow the mediaStream to perform idle work (not currently in use)

**Returns:**

true if it did some work

---

## getClientId

```
public int getClientId()
```

Get parent client connection (id)

**Returns:**

parent client connection (id)

---

## getClient

```
public IClient getClient()
```

Get parent client connection

**Returns:**

parent client connection

---

## getNetConnection

```
public com.wowza.wms.netconnection.INetConnection getNetConnection()
```

Get parent netConnection (future server to server communication)

**Returns:**

parent netConnection connection

---

## setNetConnection

```
public void setNetConnection(com.wowza.wms.netconnection.INetConnection netConnection)
```

(continued from last page)

Set parent netConnection (future server to server communication)

**Parameters:**

netConnection - netConnection connection

---

**setClient**

```
public void setClient(IClient client)
```

Set parent client connection

**Parameters:**

client - parent client connection

---

**getStreams**

```
public MediaStreamMap getStreams()
```

Get parent mediaStreamMap (owned by applicationInstance)

**Returns:**

parent mediaStreamMap

---

**packetComplete**

```
public void packetComplete()
```

Invoked by requestAdapter when at the end of a set of packets

---

**sendStreamNotFound**

```
public int sendStreamNotFound(String name)
```

Send onStatus(NetStream.Play.StreamNotFound) event

**Parameters:**

name - stream name

**Returns:**

bytes sent to client

---

**sendStreamNotFound**

```
public int sendStreamNotFound(java.io.OutputStream out,  
                           String name)
```

Send onStatus(NetStream.Play.StreamNotFound) event. Send directly to OutputStream.

**Parameters:**

out - OutputStream  
name - stream name

**Returns:**

bytes sent to client

(continued from last page)

## sendLivePlayStart

```
public int sendLivePlayStart(java.io.OutputStream out,
    String name,
    long timecode,
    long timecodeOffset)
```

Send onStatus(NetStream.Play.Start) event

**Parameters:**

- out - OutputStream
- name - stream name
- timecode - timecode
- timecodeOffset - timecode offset

**Returns:**

bytes sent to client

---

## sendVODPlaySwitch

```
public int sendVODPlaySwitch(java.io.OutputStream out,
    String name,
    long timecode)
```

Send onStatus(NetStream.Play.Transition) event

**Parameters:**

- out - OutputStream
- name - stream name
- timecode - timecode

**Returns:**

bytes sent to client

---

## sendLivePlaySwitch

```
public int sendLivePlaySwitch(java.io.OutputStream out,
    String name,
    long timecode)
```

Send onStatus(NetStream.Play.Transition) event

**Parameters:**

- out - OutputStream
- name - stream name
- timecode - timecode

**Returns:**

bytes sent to client

---

## sendLivePlaySeek

```
public int sendLivePlaySeek(java.io.OutputStream out,
    String name,
    long timecode)
```

Send onStatus(NetStream.Seek.Notify) event

**Parameters:**

- out - OutputStream

(continued from last page)

name - stream name  
timecode - timecode

**Returns:**

bytes sent to client

---

**sendPlayReset**

```
public int sendPlayReset(String name)
```

Send onStatus(NetStream.Play.Reset) event

**Parameters:**

name - stream name

**Returns:**

bytes sent to client

---

**sendPlayReset**

```
public int sendPlayReset(java.io.OutputStream out,  
String name)
```

Send onStatus(NetStream.Play.Reset) event. Send directly to OutputStream.

**Parameters:**

out - OutputStream  
name - stream name

**Returns:**

bytes sent to client

---

**sendPlayStop**

```
public int sendPlayStop(long location,  
String name)
```

Send onStatus(NetStream.Play.Stop) event

**Parameters:**

location - timecode where play stopped  
name - stream name

**Returns:**

bytes sent to client

---

**sendPlayStop**

```
public int sendPlayStop(java.io.OutputStream out,  
long location,  
String name)
```

Send onStatus(NetStream.Play.Stop) event. Send directly to OutputStream

**Parameters:**

out - OutputStream  
location - timecode where play stopped  
name - stream name

(continued from last page)

**Returns:**

bytes sent to client

---

**sendPlaySeek**

```
public int sendPlaySeek(java.io.OutputStream out,
    long location,
    long seekLocation,
    String name,
    java.util.List seekTypes)
```

Send onStatus(NetStream.Seek.Notify) event.

**Parameters:**

out - OutputStream  
location - request location of seek  
seekLocation - result location of seek  
name - stream name  
seekTypes - list of commands to respond to (seek, unpause, play)

**Returns:**

bytes sent to client

---

**sendPlaySeek**

```
public int sendPlaySeek(long location,
    long seekLocation,
    String name)
```

Send onStatus(NetStream.Seek.Notify) event.

**Parameters:**

location - request location of seek  
seekLocation - result location of seek  
name - stream name

**Returns:**

bytes sent to client

---

**sendPlaySeek**

```
public int sendPlaySeek(java.io.OutputStream out,
    long location,
    long seekLocation,
    String name)
```

Send onStatus(NetStream.Seek.Notify) event. Send directly to OutputStream

**Parameters:**

out - OutputStream  
location - request location of seek  
seekLocation - result location of seek  
name - stream name

**Returns:**

bytes sent to client

(continued from last page)

## sendPlayStart

```
public int sendPlayStart(String name,  
                        long timecode)
```

Send onStatus(NetStream.Play.Start) event

**Parameters:**

name - stream name

**Returns:**

bytes sent to client

---

## sendPlaySwitch

```
public int sendPlaySwitch(String name,  
                         long timecode)
```

Send onStatus(NetStream.Play.Start) event (it is not a typo, this also send play start but with different control codes for media switch).

**Parameters:**

name - stream name

timecode - timecode of event

**Returns:**

bytes sent to client

---

## sendPlaySwitch

```
public int sendPlaySwitch(java.io.OutputStream out,  
                         String name,  
                         boolean isSwitch,  
                         long timecode)
```

Send onStatus(NetStream.Play.Start) event (it is not a typo, this also send play start but with different control codes for media switch). Send directly to OutputStream

**Parameters:**

out - OutputStream

name - stream name

isSwitch - is this a switch or a start

timecode - timecode of event

**Returns:**

bytes sent to client

---

## sendPlayStart

```
public int sendPlayStart(java.io.OutputStream out,  
                         String name,  
                         boolean isSwitch,  
                         boolean isLive,  
                         long timecode,  
                         java.util.List seekTypes)
```

Send onStatus(NetStream.Play.Start) event (it is not a typo, this also send play start but with different control codes for media switch). Send directly to OutputStream

**Parameters:**

out - OutputStream

(continued from last page)

name - stream name  
 isSwitch - is this a switch or a start  
 isLive - is the stream live  
 timecode - timecode of event  
 seekTypes - commands to respond to (seek, unpause, play)

## sendPlayStart

```
public int sendPlayStart(java.io.OutputStream out,
                      String name,
                      boolean isSwitch,
                      long timecode,
                      java.util.List seekTypes)
```

Send onStatus(NetStream.Play.Start) event (it is not a typo, this also send play start but with different control codes for media switch). Send directly to OutputStream

**Parameters:**

out - OutputStream  
 name - stream name  
 isSwitch - is this a switch or a start  
 timecode - timecode of event  
 seekTypes - commands to respond to (seek, unpause, play)

**Returns:**

bytes sent to client

## sendPlayStart

```
public int sendPlayStart(java.io.OutputStream out,
                      String name,
                      boolean isSwitch,
                      long timecode)
```

Send onStatus(NetStream.Play.Start) event (it is not a typo, this also send play start but with different control codes for media switch). Send directly to OutputStream

**Parameters:**

out - OutputStream  
 name - stream name  
 isSwitch - is this a switch or a start  
 timecode - timecode of event

**Returns:**

bytes sent to client

## sendPlayStatus

```
public int sendPlayStatus(long timecode,
                        int statusType,
                        double duration,
                        double bytesSent)
```

Send onPlayStatus(NetStream.Play.Switch, NetStream.Play.Complete, NetStream.Play.Stop) event

**Parameters:**

timecode - timecode of event  
 statusType - status type IMediaStreamPlay.PLAYSTATUSTYPE\_\*

duration - (not used)  
 bytesSent - (not used)

(continued from last page)

**Returns:**

bytes sent to client

## sendPlayStatus

```
public int sendPlayStatus(java.io.OutputStream out,
    long timecode,
    int statusType,
    double duration,
    double bytesSent)
```

Send onPlayStatus(NetStream.Play.Switch, NetStream.Play.Complete, NetStream.Play.Stop) event. Send directly to OutputStream.

**Parameters:**

out - OutputStream  
timecode - timecode of event  
statusType - status type IMediaStreamPlay.PLAYSTATUSTYPE\_\*

duration - (not used)  
bytesSent - (not used)

**Returns:**

bytes sent to client

## sendPauseNotify

```
public int sendPauseNotify(long timecode,
    String name)
```

Send onStatus(NetStream.Pause.Notify) event

**Parameters:**

timecode - timecode of event  
name - stream name

**Returns:**

bytes sent to client

## sendPauseNotify

```
public int sendPauseNotify(java.io.OutputStream out,
    long timecode,
    String name)
```

Send onStatus(NetStream.Pause.Notify) event. Send directly to OutputStream.

**Parameters:**

out - OutputStream  
timecode - timecode of event  
name - stream name

**Returns:**

bytes sent to client

## sendUnpauseNotify

```
public int sendUnpauseNotify(java.io.OutputStream out,
    long location,
    String name,
    java.util.List seekTypes)
```

(continued from last page)

Send onStatus(NetStream.Unpause.Notify) event

**Parameters:**

out - OutputStream  
location - timecode of event  
name - stream name  
seekTypes - list of operations to respond to (seek, play, unpause)

**Returns:**

bytes sent to client

---

**sendUnpauseNotify**

```
public int sendUnpauseNotify(long location,  
                           String name)
```

Send onStatus(NetStream.Unpause.Notify) event

**Parameters:**

location - timecode of event  
name - stream name

**Returns:**

bytes sent to client

---

**sendUnpauseNotify**

```
public int sendUnpauseNotify(java.io.OutputStream out,  
                           long location,  
                           String name)
```

Send onStatus(NetStream.Unpause.Notify) event. Send directly to OutputStream.

**Parameters:**

out - OutputStream  
location - timecode of event  
name - stream name

**Returns:**

bytes sent to client

---

**isAppend**

```
public boolean isAppend()
```

Is append to media file (only valid if isRecord)

**Returns:**

is appending or rewriting media file

---

**setAppend**

```
public void setAppend(boolean isAppend)
```

Set is append to media file (only valid if isRecord)

**Parameters:**

isAppend - is appending or rewriting media file

---

## setPlayer

```
public void setPlayer(IMediaStreamPlay player)
```

Set underlying player (IMediaStreamPlay) object

**Parameters:**

player - underlying player (IMediaStreamPlay) object

---

## getPlayer

```
public IMediaStreamPlay getPlayer()
```

Get underlying player (IMediaStreamPlay) object

**Returns:**

underlying player (IMediaStreamPlay) object

---

## getAudioMissing

```
public int getAudioMissing()
```

Get number of audio bytes missing from current audio packet

**Returns:**

number of bytes missing

---

## getVideoMissing

```
public int getVideoMissing()
```

Get number of video bytes missing from current audio packet

**Returns:**

number of bytes missing

---

## getDataMissing

```
public int getDataMissing()
```

Get number of data bytes missing from current audio packet

**Returns:**

number of bytes missing

---

## addVideoData

```
public void addVideoData(byte[] data,  
                      int offset,  
                      int size)
```

Add data to current video packet

**Parameters:**

data - byte array

offset - offset in byte array

size - size of data to add

## addAudioData

```
public void addAudioData(byte[] data,  
                         int offset,  
                         int size)
```

Add data to current audio packet

**Parameters:**

data - byte array  
offset - offset in byte array  
size - size of data to add

---

## addDataData

```
public void addDataData(byte[] data,  
                         int offset,  
                         int size)
```

Add data to current data packet

**Parameters:**

data - byte array  
offset - offset in byte array  
size - size of data to add

---

## getProperties

```
public WMSProperties getProperties()
```

Get mediaStream properties

**Returns:**

properties

---

## getPlayPackets

```
public java.util.List getPlayPackets()
```

Get all available live packets

**Returns:**

play packets

---

## getMaxTimecode

```
public long getMaxTimecode()
```

Get the timecode of the latest received packet

**Returns:**

timecode of the latest received packet

---

## getLastKeyFrame

```
public AMFPacket getLastKeyFrame()
```

Get most recent video key frame

(continued from last page)

**Returns:**

play packets

---

**getLastPacket**

```
public AMFPacket getLastPacket()
```

Get most recent live packet

**Returns:**

most recent live packet or null or no live packets

---

**getStreamType**

```
public String getStreamType()
```

Get mediaStream streamType

**Returns:**

streamType

---

**setStreamType**

```
public void setStreamType(String streamType)
```

Set mediaStream streamType. This method will not change the type of the current stream.

**Parameters:**

streamType

---

**sendDirectAMF3**

```
public void sendDirectAMF3(String handlerName,  
Object[] params)
```

Call client side NetStream method/handler and send event to underlying stream (will record event). Force AMF3 encoding.

**Parameters:**

handlerName - handler name

params - variable list of parameters. All parameters will be wrapped in AMFDataObj.

---

**sendDirect**

```
public void sendDirect(String handlerName,  
Object[] params)
```

Call client side NetStream method/handler and send event to underlying stream (will record event)

**Parameters:**

handlerName - handler name

params - variable list of parameters. All parameters will be wrapped in AMFDataObj.

---

**sendDirectAMF3**

```
public void sendDirectAMF3(String handlerName)
```

Call client side NetStream method/handler and send event to underlying stream (will record event). Force AMF3 encoding.

(continued from last page)

**Parameters:**

handlerName - handler name

**sendDirect**

```
public void sendDirect(String handlerName)
```

Call client side NetStream method/handler and send event to underlying stream (will record event)

**Parameters:**

handlerName - handler name

**sendAMF3**

```
public void sendAMF3(String handlerName,  
Object[] params)
```

Call client side NetStream method/handler. Force AMF3 encoding.

**Parameters:**

handlerName - handler name

params - variable list of parameters. All parameters will be wrapped in AMFDataObj.

**send**

```
public void send(String handlerName,  
Object[] params)
```

Call client side NetStream method/handler

**Parameters:**

handlerName - handler name

params - variable list of parameters. All parameters will be wrapped in AMFDataObj.

**sendAMF3**

```
public void sendAMF3(String handlerName)
```

Call client side NetStream method/handler with no parameters. Force AMF3 encoding.

**Parameters:**

handlerName - handler name

**send**

```
public void send(String handlerName)
```

Call client side NetStream method/handler with no parameters

**Parameters:**

handlerName - handler name

**close**

```
public void close()
```

Close mediaStream

(continued from last page)

## isOpen

```
public boolean isOpen()
```

Is mediaStream open

**Returns:**

is mediaStream open

---

## setOpen

```
public void setOpen(boolean isOpen)
```

Set mediaStream open

**Parameters:**

isOpen - mediaStream open

---

## getFastPlaySettings

```
public FastPlaySettings getFastPlaySettings()
```

Get current fastPlay settings

**Returns:**

fastPlay settings

---

## setFastPlaySettings

```
public void setFastPlaySettings(FastPlaySettings fastPlaySettings)
```

Set fastPlay settings

**Parameters:**

fastPlaySettings - fastPlay settings

---

## clearFastPlaySettings

```
public void clearFastPlaySettings()
```

Clear fastPlay settings

---

## isReceiveAudio

```
public boolean isReceiveAudio()
```

Is client currently receiving audio. Controlled by client side call receiveAudio.

**Returns:**

receive audio

---

## setReceiveAudio

```
public void setReceiveAudio(boolean receiveAudio)
```

Set receive audio

**Parameters:**

(continued from last page)

`receiveAudio` - receive audio

## isReceiveVideo

`public boolean isReceiveVideo()`Is client currently receiving video. Controlled by client side call `receiveVideo`**Returns:**`receive video`

## setReceiveVideo

`public void setReceiveVideo(boolean receiveVideo)`

Set receive video

**Parameters:**`receiveVideo` - receive video

## getReceiveVideoFPS

`public int getReceiveVideoFPS()`

Set frame per seconds for video (not currently implemented)

**Returns:**`video frames per second`

## setReceiveVideoFPS

`public void setReceiveVideoFPS(int receiveVideoFPS)`

Set frame per second for video (not currently implemented)

**Parameters:**`receiveVideoFPS` - video frames per second

## getMediaIOPerformance

`public IOPerformanceCounter getMediaIOPerformance()`

Get IO performance counter

**Returns:**`IO performance counter`

## incrementMediaOutBytes

`public long incrementMediaOutBytes(long bytes, long count)`

Increment the number of mediaStream bytes sent and number of packets sent

**Parameters:**`bytes` - number of bytes sent`count` - number of packets sent**Returns:**

(continued from last page)

total number of bytes sent (after increment)

---

## incrementMediaLossBytes

```
public long incrementMediaLossBytes(long bytes,  
                                long count)
```

Increment the number of mediaStream loss bytes sent and number of packets sent

**Parameters:**

bytes - number of bytes sent  
count - number of packets sent

**Returns:**

total number of bytes sent (after increment)

---

## incrementMediaInBytes

```
public long incrementMediaInBytes(long increment)
```

Increment the number of mediaStream bytes received

**Parameters:**

increment - number of byte received

**Returns:**

total number of bytes received (after increment)

---

## publish

```
public void publish()
```

Publish mediaStream

---

## trim

```
public void trim()
```

Trim mediaStream. This method will remove live packets that are older than the live buffer size.

---

## handleCallback

```
public void handleCallback(com.wowza.wms.request.RequestFunction function)
```

Routes request function to callback handler onStatus, onPlayStatus or [method/handler]

**Parameters:**

function - request function

---

## unregisterCallback

```
public void unregisterCallback(String handlerName)
```

Unregister a callback handler

**Parameters:**

handlerName - handler name

---

---

## registerCallback

```
public void registerCallback(String handlerName,  
    IMediaStreamCallback callback)
```

Register a callback handler

**Parameters:**

handlerName - handler name  
callback - callback object

---

## unregisterOnStatus

```
public void unregisterOnStatus(IMediaStreamCallback callback)
```

Unregister onStatus handler

**Parameters:**

callback

---

## registerOnStatus

```
public void registerOnstatus(IMediaStreamCallback callback)
```

Register onStatus handler

**Parameters:**

callback - callback object

---

## unregisterOnPlayStatus

```
public void unregisterOnPlayStatus(IMediaStreamCallback callback)
```

Unregister onPlayStatus handler

**Parameters:**

callback - callback object

---

## registerOnPlayStatus

```
public void registerOnPlayStatus(IMediaStreamCallback callback)
```

Register onPlayStatus handler

**Parameters:**

callback - callback object

---

## addClientListener

```
public void addClientListener(IMediaStreamActionNotify actionListener)
```

Add client listener. Listens for (onPlay, onPublish, onPause, onSeek, onStop)

**Parameters:**

actionListener - listener

---

(continued from last page)

## addClientListener

```
public void addClientListener(IMediaStreamActionNotify2 actionListener)
```

Add client listener. Listens for (onPlay, onPublish, onPause, onSeek, onStop)

**Parameters:**

actionListener - listener

---

## addClientListener

```
public void addClientListener(IMediaStreamActionNotify3 actionListener)
```

Add client listener. Listens for (onPlay, onPublish, onPause, onSeek, onStop)

**Parameters:**

actionListener - listener

---

## removeClientListener

```
public void removeClientListener(IMediaStreamActionNotify actionListener)
```

Remove client listener. Listens for (onPlay, onPublish, onPause, onSeek, onStop)

**Parameters:**

actionListener - listener

---

## removeClientListener

```
public void removeClientListener(IMediaStreamActionNotify2 actionListener)
```

Remove client listener. Listens for (onPlay, onPublish, onPause, onSeek, onStop)

**Parameters:**

actionListener - listener

---

## removeClientListener

```
public void removeClientListener(IMediaStreamActionNotify3 actionListener)
```

Remove client listener. Listens for (onPlay, onPublish, onPause, onSeek, onStop)

**Parameters:**

actionListener - listener

---

## notifyActionPlay

```
public void notifyActionPlay(String streamName,  
                           double playStart,  
                           double playLen,  
                           int playReset)
```

Notify client listeners of play action

**Parameters:**

streamName - stream name

playStart - play start

playLen - play length

playReset - play reset

## notifyActionPauseRaw

```
public void notifyActionPauseRaw(boolean isPause,  
                                long location)
```

Notify client listeners of pauseRaw action

**Parameters:**

isPause - is pause or unpause  
location - timecode (milliseconds) of action

---

## notifyActionPause

```
public void notifyActionPause(boolean isPause,  
                             long location)
```

Notify client listeners of pause action

**Parameters:**

isPause - is pause or unpause  
location - timecode (milliseconds) of action

---

## notifyActionSeek

```
public void notifyActionSeek(double location)
```

Notify client listeners of seek action

**Parameters:**

location - timecode (milliseconds) of seek request

---

## notifyActionPublish

```
public void notifyActionPublish(String streamName,  
                           boolean isRecord,  
                           boolean isAppend)
```

Notify client listeners of publish action

**Parameters:**

streamName - stream name  
isRecord - is record or live  
isAppend - is append if isRecord is true

---

## notifyActionUnPublish

```
public void notifyActionUnPublish(String streamName,  
                               boolean isRecord,  
                               boolean isAppend)
```

Notify client listeners of unpublish action

**Parameters:**

streamName - stream name  
isRecord - is record or live  
isAppend - is append if isRecord is true

---

(continued from last page)

## notifyActionOnMetaData

```
public void notifyActionOnMetaData(AMFPacket metaDataPacket)
```

Notify client listeners of onMetaData change

**Parameters:**

metaDataPacket - metaDataPacket

---

## notifyActionOnCodecInfoVideo

```
public void notifyActionOnCodecInfoVideo(com.wowza.wms.media.model.MediaCodecInfoVideo codecInfoVideo)
```

Notify client listeners of video codec information change

**Parameters:**

codecInfoVideo - video codec information

---

## notifyActionOnCodecInfoAudio

```
public void notifyActionOnCodecInfoAudio(com.wowza.wms.media.model.MediaCodecInfoAudio codecInfoAudio)
```

Notify client listeners of audio codec information change

**Parameters:**

codecInfoAudio - audio codec information

---

## notifyActionStop

```
public void notifyActionStop()
```

Notify client listeners of stop action

---

## isClustered

```
public boolean isClustered()
```

not used

**Returns:**

isClustered

---

## setClustered

```
public void setClustered(boolean isClustered)
```

not used

**Parameters:**

isClustered

---

## getCacheName

```
public String getCacheName()
```

not used

(continued from last page)

**Returns:**

cache name

**startPublishing**

```
public void startPublishing()
```

Start publishing live stream

**stopPublishing**

```
public void stopPublishing()
```

Stop publishing live stream

**getStreamFileForWrite**

```
public java.io.File getStreamFileForWrite()
```

Get the File object to write to a stream (get stream name, ext and query from stream object)

**Returns:**

resultant File object

**getStreamFileForWrite**

```
public java.io.File getStreamFileForWrite(String name,
String ext,
String query)
```

Get the File object to write to a stream (specify name, ext and query)

**Parameters:**

name - stream name

ext - stream prefix (Ex. mp4:)

query - query part of stream name (Ex. mystream?param1=value1)

**Returns:**

resultant File object

**getStreamFileForRead**

```
public java.io.File getStreamFileForRead()
```

Get the File object to read from a stream (get stream name, ext and query from stream object)

**Returns:**

resultant File object

**getStreamFileForRead**

```
public java.io.File getStreamFileForRead(String name,
String ext,
String query)
```

Get the File object to read from a stream (specify name, ext and query)

**Parameters:**

name - stream name

(continued from last page)

`ext` - stream prefix (Ex. mp4:)

`query` - query part of stream name (Ex. mystream?param1=value1)

**Returns:**

resultant File object

---

## sendControlBytes

```
public int sendControlBytes(int controlType,  
                         java.io.OutputStream out)
```

Send playback control bytes. Valid values are (0, 1, 4)

**Parameters:**

`controlType` - control types (0, 1, 4)

`out` - OutputStream

**Returns:**

bytes sent to client

---

## getBurstStartStop

```
public byte[] getBurstStartStop(boolean isStart)
```

Get the dynamic streaming burst start/stop AMF packet

**Parameters:**

`isStart` - is start

**Returns:**

byte array with AMF packet

---

## getRespAMFAudioObj

```
public AMFObj getRespAMFAudioObj()
```

Get audio response channel object

**Returns:**

audio response channel object

---

## getRespAMFVideoObj

```
public AMFObj getRespAMFVideoObj()
```

Get video response channel object

**Returns:**

video response channel object

---

## getRespAMFDataObj

```
public AMFObj getRespAMFDataObj()
```

Get data response channel object

**Returns:**

data response channel object

## getQueryStr

```
public String getQueryStr()
```

Get play/publish name query string. Example: if play name is flv:test?param1=data1&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 onMetdata event data. If false, replace.

**Parameters:**

mergeOnMetadata - true, merge incoming onMetadata events with the current onMetdata event data. If false, replace.

---

## getDvrRecorder

```
public String getDvrRecorder()
```

Get the DVR Recorder for this stream

**Returns:**

DVR Recorder

---

## getDvrRepeater

```
public String getDvrRepeater()
```

Get the DVR repeater name for this stream

**Returns:**

repeater name

---

## setDvrRecorder

```
public void setDvrRecorder(String recorderName)
```

Set the DVR Recorder that this stream is using

**Parameters:**

recorderName - DVR Recorder

---

## removeDvrRecorder

```
public ILiveStreamDvrRecorder removeDvrRecorder(String name)
```

Remove a live stream dvr by name

**Parameters:**

name - dvr name

**Returns:**

live stream dvr

---

## putDvrRecorder

```
public void putDvrRecorder(String name,  
                           ILiveStreamDvrRecorder dvr)
```

Add a live stream dvr to this stream

**Parameters:**

name - dvr name

dvr - live stream dvr

## getLiveStreamTranscoders

```
public java.util.Map getLiveStreamTranscoders()
```

Get the list of transcoders for this stream.

**Returns:**

list of transcoders for this stream

---

## getLiveStreamTranscoder

```
public ILiveStreamTranscoder getLiveStreamTranscoder(String name)
```

Get a live stream transcoder for this stream by name

**Parameters:**

name - transcoder name

**Returns:**

live stream transcoder

---

## removeLiveStreamTranscoder

```
public ILiveStreamTranscoder removeLiveStreamTranscoder(String name)
```

Remove a live stream transcoder by name

**Parameters:**

name - transcoder name

**Returns:**

live stream transcoder

---

## putLiveStreamTranscoder

```
public void putLiveStreamTranscoder(String name,  
                                    ILiveStreamTranscoder liveStreamTranscoder)
```

Add a live stream transcoder to this stream

**Parameters:**

name - transcoder name

liveStreamTranscoder - live stream transcoder

---

## isTranscodeResult

```
public boolean isTranscodeResult()
```

Is this stream the result of a transcode operation.

**Returns:**

true if stream the result of a transcode operation

---

## setTranscodeResult

```
public void setTranscodeResult(boolean isTranscodeResult)
```

Is this stream the result of a transcode operation.

(continued from last page)

**Parameters:**

isTranscodeResult - true if stream the result of a transcode operation

---

**addVideoH264SEIListener**

```
public void addVideoH264SEIListener(IMediaStreamH264SEINotify h264SEIListener)
```

Add an H.264 SEI listener. This listener will be notified of all incoming H.264 video packets and has the ability to read and/or modify SEI section of the video frame.

**Parameters:**

h264SEIListener - H.264 SEI listener

---

**removeVideoH264SEIListener**

```
public void removeVideoH264SEIListener(IMediaStreamH264SEINotify h264SEIListener)
```

Remove an H.264 SEI listener. This listener will be notified of all incoming H.264 video packets and has the ability to read and/or modify SEI section of the video frame.

**Parameters:**

h264SEIListener - H.264 SEI listener

---

**isVideoH264SEIListenerEmpty**

```
public boolean isVideoH264SEIListenerEmpty()
```

Is H.264 SEI listener list empty. This listener will be notified of all incoming H.264 video packets and has the ability to read and/or modify SEI section of the video frame.

**Returns:**

true if H.264 SEI listener list empty

---

**notifyVideoH264Packet**

```
public void notifyVideoH264Packet(AMFPacket packet,  
com.wowza.wms.media.h264.H264SEIMessages seiMessages)
```

Notify H.264 SEI listener.

**Parameters:**

packet - AMF Packet

seiMessages - seiMessages

---

**addCallbackListener**

```
public void addCallbackListener(IMediaStreamCallback callbackListener)
```

Add a callback listener to listen for all stream metadata

**Parameters:**

callbackListener

---

**removeCallbackListener**

```
public void removeCallbackListener(IMediaStreamCallback callbackListener)
```

Remove a callback listener to listen for all stream metadata

(continued from last page)

**Parameters:**

callbackListener

**addLivePacketListener**

```
public void addLivePacketListener(IMediaStreamLivePacketNotify listener)
```

Add a listener to receiver notification of new live AMF packets

**Parameters:**

listener - listener

**removeLivePacketListener**

```
public void removeLivePacketListener(IMediaStreamLivePacketNotify listener)
```

Remove a listener to receiver notification of new live AMF packets

**Parameters:**

listener - listener

**notifyLivePacket**

```
public void notifyLivePacket(AMFPacket packet)
```

Notify new live AMF packet

**Parameters:**

packet - new live AMF packet

**isPublisherStream**

```
public boolean isPublisherStream()
```

Returns true if stream was created by stream publisher API.

**Returns:**

true if stream was created by stream publisher API

**setPublisherStream**

```
public void setPublisherStream(boolean isPublisherStream)
```

Set stream was created by stream publisher.

**Parameters:**

isPublisherStream - true if stream was created by stream publisher API

## com.wowza.wms.stream

### Interface IMediaStreamActionNotify

All Subinterfaces:

[IMediaStreamActionNotify2](#), [IMediaStreamActionNotify3](#)

public interface **IMediaStreamActionNotify**

extends

IMediaStreamActionNotify: listener interface used by IMediaStream addClientListener

### Method Summary

void	<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

<pre>void</pre>	<b><a href="#">onCodecInfoAudio</a></b> ( <a href="#">IMediaStream</a> stream, <a href="#">com.wowza.wms.media.model.MediaCodecInfoAudio</a> codecInfoAudio)
	Triggered when publishing stream receives codec information.

  

<pre>void</pre>	<b><a href="#">onCodecInfoVideo</a></b> ( <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	<b>onCallback</b> ( <a href="#">IMediaStream</a> stream, com.wowza.wms.request.RequestFunction function, <a href="#">AMFDataList</a> params)
	Triggered on callback event

### 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	<b><a href="#">streamToFileForRead</a></b> ( <a href="#">IMediaStream</a> stream) Get the File object to read from a stream (get stream name, ext and query from stream object)
java.io.File	<b><a href="#">streamToFileForRead</a></b> ( <a href="#">IMediaStream</a> stream, String name, String ext, String query) Get the File object to read from a stream (specify name, ext and query)
java.io.File	<b><a href="#">streamToFileForWrite</a></b> ( <a href="#">IMediaStream</a> stream) Get the File object to write to a stream (get stream name, ext and query from stream object)
java.io.File	<b><a href="#">streamToFileForWrite</a></b> ( <a href="#">IMediaStream</a> stream, String name, String ext, String query) 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 IMediaStreamLivePacketNotify**

public interface **IMediaStreamLivePacketNotify**  
extends

---

**Method Summary**

void	<a href="#">onLivePacket</a> ( <a href="#">IMediaStream</a> stream, <a href="#">AMFPacket</a> packet)
------	---

---

**Methods****onLivePacket**

public void **onLivePacket**([IMediaStream](#) stream,  
[AMFPacket](#) packet)

**com.wowza.wms.stream**  
**Interface IMediaStreamMediaCaster**

---

public interface **IMediaStreamMediaCaster**  
extends

IMediaStreamMediaCaster: Internal use

---

**Method Summary**

<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="#"><u>onStreamStart</u></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  
IApplication.setStreamNameAliasProvider(IMediaStreamNameAliasProvider streamNameAliasProvider). See updated interface  
IMediaStreamNameAliasProvider2 which provides more complete context information.

### Method Summary

String	<a href="#">resolvePlayAlias</a> ( <a href="#">IAplicationInstance</a> appInstance, String name) Called to resolve a play alias
String	<a href="#">resolveStreamAlias</a> ( <a href="#">IAplicationInstance</a> appInstance, String name) Called to resolve a stream alias

### Methods

#### resolvePlayAlias

```
public String resolvePlayAlias(IAplicationInstance 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(IAplicationInstance 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  
IAplication.setStreamNameAliasProvider(IMediaStreamNameAliasProvider streamNameAliasProvider).

### Method Summary

String	<a href="#">resolvePlayAlias</a> ( <a href="#">IAppliationInstance</a> appInstance, String name, <a href="#">IClient</a> client) Resolve play alias for RTMP streaming
String	<a href="#">resolvePlayAlias</a> ( <a href="#">IAppliationInstance</a> appInstance, String name, <a href="#">IHTTPStreamerSession</a> httpSession) Resolve play alias for HTTP streaming.
String	<a href="#">resolvePlayAlias</a> ( <a href="#">IAppliationInstance</a> appInstance, String name, <a href="#">ILiveStreamPacketizer</a> liveStreamPacketizer) Resolve play alias for live stream packetizer
String	<a href="#">resolvePlayAlias</a> ( <a href="#">IAppliationInstance</a> appInstance, String name, <a href="#">RTSPSession</a> rtpSession) Resolve play alias for RTSP/RTP streaming
String	<a href="#">resolveStreamAlias</a> ( <a href="#">IAppliationInstance</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(IAppliationInstance 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(IAplicationInstance 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(IAplicationInstance 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(IAplicationInstance 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(IAplicationInstance 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 mediaStreamPlayer 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 href="#">IMediaStream</a> parent, <a href="#">MediaStreamMap</a> streams)</a> Initialize mediaStreamPlayer
void	<a href="#">initLiveStreamRepeating(String liveStreamPacketizer, String liveStreamRepeater)</a> 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(int pauseType, long timecode)</a> pause mediaPlayStream
void	<a href="#">pauseRaw(int pauseType, long timecode)</a> pauseRaw mediaPlayStream
boolean	<a href="#">play()</a> Return true if there are packets to play
int	<a href="#">play(java.io.OutputStream out, <a href="#">AMFObj</a> wmsObjAudio, <a href="#">AMFObj</a> wmsObjVideo, <a href="#">AMFObj</a> wmsObjData, long[] sizes)</a> Write new packets or play packets
void	<a href="#">reset(boolean isReset)</a> Reset mediaStreamPlayer
void	<a href="#">resetNoLookup()</a> Reset but do not lookup current position in live stream
void	<a href="#">seek(int location)</a> seek mediaPlayStream

void	<a href="#"><u>setBufferTime(int bufferTime)</u></a> Set buffer time
void	<a href="#"><u>setName(String name, String oldName, String ext, String queryStr, double playStart, double playLen, int playTransition)</u></a> Set mediaPlayStream name, extension, query string, play start, play len, play reset
void	<a href="#"><u>shutdown()</u></a> Shutdown mediaStreamPlayer
long	<a href="#"><u>size()</u></a> Get stream media file size
void	<a href="#"><u>startPlay()</u></a> Start playing stream
void	<a href="#"><u>stopName(String name, String oldName, String ext, String queryStr, double playStart, double playLen, int playTransition)</u></a> Stop stream name
void	<a href="#"><u>switchName(String name, String oldName, String ext, String queryStr, double playStart, double playLen, int playTransition)</u></a> Switch to stream name
void	<a href="#"><u>switchPlay()</u></a>
void	<a href="#"><u>updateLoggingValues()</u></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="#"><u>resetTimecodes()</u></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 uses 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="#"><u>getDuration()</u></a> Get the recorded duration of the file in seconds
boolean	<a href="#"><u>isVersionFile()</u></a> Return true if the old file is to be versioned
boolean	<a href="#"><u>isWaitForVideoKeyFrame( )</u></a> get wait for key frame
void	<a href="#"><u>putMetaData(String name, AMFData value)</u></a> Add metadata to the metadata packet.
void	<a href="#"><u>setMediaWriterItem(MediaWriterItem mediaWriterItem)</u></a> Set the media write definition
void	<a href="#"><u>setParent(IMediaStream parent)</u></a> Set the parent stream for this media write object
void	<a href="#"><u>setVersionFile(boolean versionFile)</u></a> Set to true if the old file is to be versioned
void	<a href="#"><u>setWaitForVideoKeyFrame(boolean waitForVideoKeyFrame)</u></a> Set to true if you want the recorder to skip opening frames until it hits a key frame
void	<a href="#"><u>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)</u></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  
IAplicationInstance.addMediaWriterListener(IMediaWriterActionNotify listener)

### Method Summary

void	<a href="#"><b>onFLVAddMetadata</b></a> ( <a href="#">IMediaStream</a> stream, java.util.Map extraMetadata) Called just before metadata is written to the file (FLV only)
void	<a href="#"><b>onWriteComplete</b></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 MediaReaderEncInfo

```
java.lang.Object
+-com.wowza.wms.stream.MediaReaderEncInfo
```

---

```
public class MediaReaderEncInfo
extends Object
```

### Field Summary

public static final	<a href="#">METHOD_NONE</a>
	Value: 0
public static final	<a href="#">METHOD_SAMPLE_PLAYREADY</a>
	Value: 1

### Constructor Summary

public	<a href="#">MediaReaderEncInfo()</a>
--------	--------------------------------------

### Method Summary

void	<a href="#">addPlayReadyDecryptorKey(com.wowza.wms.drm.playready.PlayReadyKeyInfo decryptorKey)</a>
int	<a href="#">getEncMethod()</a>
com.wowza.wms.drm.playready.PlayReadyKeyInfo	<a href="#">getPlayReadyDecryptorKey()</a>
com.wowza.wms.drm.playready.PlayReadyKeyInfo	<a href="#">getPlayReadyDecryptorKey(String keyId)</a>
void	<a href="#">setEncMethod(int encMethod)</a>

#### Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait,
```

### Fields

#### METHOD\_NONE

```
public static final int METHOD_NONE
```

(continued from last page)

Constant value: 0

---

**METHOD\_SAMPLE\_PLAYREADY**

```
public static final int METHOD_SAMPLE_PLAYREADY
```

Constant value: 1

**Constructors****MediaReaderEncInfo**

```
public MediaReaderEncInfo()
```

**Methods****addPlayReadyDecryptorKey**

```
public void addPlayReadyDecryptorKey(com.wowza.wms.drm.playready.PlayReadyKeyInfo  
decryptorKey)
```

---

**getPlayReadyDecryptorKey**

```
public com.wowza.wms.drm.playready.PlayReadyKeyInfo getPlayReadyDecryptorKey()
```

---

**getPlayReadyDecryptorKey**

```
public com.wowza.wms.drm.playready.PlayReadyKeyInfo getPlayReadyDecryptorKey(String  
keyId)
```

---

**getEncMethod**

```
public int getEncMethod()
```

---

**setEncMethod**

```
public void setEncMethod(int encMethod)
```

## com.wowza.wms.stream Class MediaStreamBase

```
java.lang.Object
+-com.wowza.wms.stream.MediaStream
+-com.wowza.wms.stream.MediaStreamBase
```

All Implemented Interfaces:

[IMediaStream](#)

```
public class MediaStreamBase
extends MediaStream
```

### Field Summary

public static	<a href="#">p</a> Deprecated.
public static	<a href="#">sinfo</a> Deprecated.

### Fields inherited from class com.wowza.wms.stream.MediaStream

```
actionListeners, AMLST_STREAM_EXT, audioBytes, audioSize, audioTC, BASE_STREAM_EXT,
bufferTime, callbackListeners, callbacks, client, dataBytes, dataSize, dataTC, dataType,
dvrLock, dvrMap, dvrPlayer, dvrRecorder, dvrRecorderList, dvrRepeater, elapsedTime, ext,
fastPlaySettings, FLV_STREAM_EXT, h264SEIListeners, headerSize, httpStreamerSession,
ID3_STREAM_EXT, isAppend, isClustered, isMediaCasterPlay, isOpen, isPlay, isPlaying,
isPublisherStream, isRecord, isTranscodeResult, livePacketListeners, liveStreamPacketizer,
liveStreamPacketizerList, liveStreamRepeater, liveStreamTranscoderList, lock,
mediaIOPerformance, mergeOnMetadata, metaDataProvider, MP3_STREAM_EXT, MP4_STREAM_EXT, name,
netConnection, NGRP_STREAM_EXT, parent, player, playLen, playStart, playTransition, properties,
queryStr, receiveAudio, receiveVideo, receiveVideoFPS, rtpStream, SEND_CONTROL0,
SEND_CONTROL1, SEND_CONTROL3, SEND_CONTROL4, sendDirectLock, sendDirectMessages,
sendPlayStopLogEvent, sendPublishStopLogEvent, sendRecordStopLogEvent, SMIL_STREAM_EXT,
src, STREAM_DEFAULTNAME, streamType, transcoderLock, transcoderMap, tss, videoBytes, videoSize,
videoTC
```

### Fields inherited from interface com.wowza.wms.stream.IMediaStream

[AUDIOSAMPLEACCESS](#), [READACCESS](#), [VIDEOSAMPLEACCESS](#), [WRITEACCESS](#)

### Constructor Summary

public	<a href="#">MediaStreamBase()</a>
--------	-----------------------------------

### Method Summary

void	<a href="#">init(MediaStreamMap parent, int src, WMSProperties properties)</a>
------	--

void	<a href="#"><u>publish()</u></a>
------	----------------------------------

void	<a href="#"><u>trim()</u></a>
------	-------------------------------

Methods inherited from class com.wowza.wms.stream.MediaStream
---

```

addAudioCodecConfigPacket, addAudioData, addAudioExtraData, addCallbackListener,
addClientListener, addClientListener, addClientListener, addDataData,
addDataExtraData, addLivePacketListener, addVideoCodecConfigPacket, addVideoData,
addVideoExtraData, addVideoH264SEIListener, clear, clearFastPlaySettings,
clearLoggingValues, close, flush, getAccess, getAudioCodecConfigPacket,
getAudioMissing, getAudioSize, getAudioTC, getBufferTime, getBurstStartStop,
getCacheName, getClient, getClientId, getContextStr, getDataMissing, getDataSize,
getDataTC, getDataType, getDvrRecorder, getDvrRecorder, getDvrRecorderList,
getDvrRepeater, getElapsedTime, getExt, getFastPlaySettings, getHeaderSize,
getHTTPStreamerSession, getLastKeyFrame, getLastPacket, getLiveStreamDvr,
getLiveStreamDvrs, getLiveStreamPacketizer, getLiveStreamPacketizer,
getLiveStreamPacketizerList, getLiveStreamRepeater, getLiveStreamTranscoder,
getLiveStreamTranscoderList, getLiveStreamTranscoders, getMaxTimecode,
getMediaIOPerformance, getMetaDataProvider, getName, getNetConnection, getPlayer,
getPlayPackets, getProperties, getPublishAudioCodecId, getPublishVideoCodecId,
getQueryStr, getReceiveVideoFPS, getRespAMFAudioObj, getRespAMFDataObj,
getRespAMFVideoObj, getRTPStream, getSrc, getStreamFileForRead, getStreamFileForRead,
getStreamFileForWrite, getStreamFileForWrite, getStreams, getStreamType, getTss,
getUniqueStreamIdStr, getVideoCodecConfigPacket, getVideoMissing, getVideoSize,
getVideoTC, handleCallback, idle, incrementMediaInBytes, incrementMediaLossBytes,
incrementMediaOutBytes, init, initDvrRepeating, initLiveStreamRepeating, isAppend,
isClustered, isMediaCasterPlay, isMergeOnMetadata, isOpen, isPlay, isPlaying,
isPublisherStream, isPublishStreamReady, isReceiveAudio, isReceiveVideo, isRecord,
isSendDirectMessages, isSendPlayStopLogEvent, isSendPublishStopLogEvent,
isSendRecordStopLogEvent, isTranscodeResult, isVideoH264SEIListenerEmpty, length,
notifyActionOnCodecInfoAudio, notifyActionOnCodecInfoVideo, notifyActionOnMetaData,
notifyActionPause, notifyActionPauseRaw, notifyActionPlay, notifyActionPublish,
notifyActionSeek, notifyActionStop, notifyActionUnPublish, notifyLivePacket,
notifyVideoH264Packet, packetComplete, processSendDirectMessages, publish,
putDvrRecorder, putLiveStreamTranscoder, registerCallback, registerOnPlayStatus,
registerOnStatus, removeCallbackListener, removeClientListener, removeClientListener,
removeClientListener, removeDvrRecorder, removeLivePacketListener,
removeLiveStreamTranscoder, removeVideoH264SEIListener, reset, send, send, sendAMF3,
sendAMF3, sendControlBytes, sendDataToCallback, sendDirect, sendDirect,
sendDirectAMF3, sendDirectAMF3, sendDirectInternal, sendDirectInternal, sendInternal,
sendInternal, sendLivePlaySeek, sendLivePlayStart, sendLivePlaySwitch,
sendPauseNotify, sendPauseNotify, sendPlayReset, sendPlayReset, sendPlaySeek,
sendPlaySeek, sendPlaySeek, sendPlayStart, sendPlayStart, sendPlayStart,
sendPlayStart, sendPlayStatus, sendPlayStatus, sendPlayStop, sendPlayStop,
sendPlaySwitch, sendPlaySwitch, sendStreamNotFound, sendStreamNotFound,
sendUnpauseNotify, sendUnpauseNotify, sendUnpauseNotify, sendVODPlaySwitch,
setAppend, setAudioSize, setAudioTC, setAudioTC, setBufferTime, setClient,
setClustered, setDataSize, setDataTC, setDataTC, 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,
setPublisherStream, setPublishVideoCodecId, setQueryStr, setReceiveAudio,
setReceiveVideo, setReceiveVideoFPS, setRecord, setRTPStream,
setSendPlayStopLogEvent, setSendPublishStopLogEvent, setSendRecordStopLogEvent,
setSrc, setStreamType, setTranscodeResult, setTss, setVideoSize, setVideoTC,
setVideoTC, shutdown, size, startAudioPacket, startDataPacket, startPublishing,
startVideoPacket, stopName, stopPublishing, switchName, trim, unregisterCallback,
unregisterOnPlayStatus, unregisterOnStatus, updateLoggingDuration,
updateLoggingValues

```

**Methods inherited from class** `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`

**Methods inherited from interface** `com.wowza.wms.stream.IMediaStream`

```

addAudioCodecConfigPacket, addAudioData, addAudioExtraData, addCallbackListener,
addClientListener, addClientListener, addClientListener, addDataData,
addDataExtraData, addLivePacketListener, addVideoCodecConfigPacket, addVideoData,
addVideoExtraData, addVideoH264SEIListener, clear, clearFastPlaySettings,
clearLoggingValues, close, flush, getAccess, getAudioCodecConfigPacket,
getAudioMissing, getAudioSize, getAudioTC, getBufferTime, getBurstStartStop,
getCacheName, getClient, getClientId, getContextStr, getDataMissing, getDataSize,
getDataTC, getDataType, getDvrRecorder, getDvrRecorder, getDvrRecorderList,
getDvrRepeater, getElapsedTime, getExt, getFastPlaySettings, getHeaderSize,
getHTTPStreamerSession, getLastKeyFrame, getLastPacket, getLiveStreamPacketizer,
getLiveStreamPacketizer, getLiveStreamPacketizerList, getLiveStreamRepeater,
getLiveStreamTranscoder, getLiveStreamTranscoderList, getLiveStreamTranscoders,
getMaxTimecode, getMediaIOPerformance, getMetaDataProvider, getName,
getNetConnection, getPlayer, getPlayPackets, getProperties, getPublishAudioCodecId,
getPublishVideoCodecId, getQueryStr, getReceiveVideoFPS, getRespAMFAudioObj,
getRespAMFDataObj, getRespAMFVideoObj, getRTPStream, getSrc, getStreamFileForRead,
getStreamFileForRead, getStreamFileForWrite, getStreamFileForWrite, getStreams,
getStreamType, getUniqueStreamIdStr, getVideoCodecConfigPacket, getVideoMissing,
getVideoSize, getVideoTC, handleCallback, idle, incrementMediaInBytes,
incrementMediaLossBytes, incrementMediaOutBytes, init, initLiveStreamRepeating,
isAppend, isClustered, isMediaCasterPlay, isMergeOnMetadata, isOpen, isPlay,
isPlaying, isPublisherStream, isPublishStreamReady, isReceiveAudio, isReceiveVideo,
isRecord, isSendPlayStopLogEvent, isSendPublishStopLogEvent,
isSendRecordStopLogEvent, isTranscodeResult, isVideoH264SEIListenerEmpty, length,
notifyActionOnCodecInfoAudio, notifyActionOnCodecInfoVideo, notifyActionOnMetaDataAdapter,
notifyActionPause, notifyActionPauseRaw, notifyActionPlay, notifyActionPublish,
notifyActionSeek, notifyActionStop, notifyActionUnPublish, notifyLivePacket,
notifyVideoH264Packet, packetComplete, publish, putDvrRecorder,
putLiveStreamTranscoder, registerCallback, registerOnPlayStatus, registerOnStatus,
removeCallbackListener, removeClientListener, removeClientListener,
removeClientListener, removeDvrRecorder, removeLivePacketListener,
removeLiveStreamTranscoder, removeVideoH264SEIListener, send, send, sendAMF3,
sendAMF3, sendControlBytes, sendDirect, sendDirect, sendDirectAMF3, sendDirectAMF3,
sendLivePlaySeek, sendLivePlayStart, sendLivePlaySwitch, sendPauseNotify,
sendPauseNotify, sendPlayReset, sendPlayReset, sendPlaySeek, sendPlaySeek,
sendPlaySeek, sendPlayStart, sendPlayStart, sendPlayStart, sendPlayStart,
sendPlayStatus, sendPlayStatus, sendPlayStop, sendPlayStop, sendPlaySwitch,
sendPlaySwitch, sendStreamNotFound, sendStreamNotFound, sendUnpauseNotify,
sendUnpauseNotify, sendUnpauseNotify, sendVODPlaySwitch, setAppend, setAudioSize,
setAudioTC, setAudioTC, setBufferTime, setClient, setClustered, setDataSize,
setDataTC, setDataTC, setDataType, setDvrRecorder, setDvrRecorderList, setExt,
setFastPlaySettings, setHeaderSize, setHTTPStreamerSession, setIsPlaying,
setLiveStreamPacketizer, setLiveStreamPacketizerList, setLiveStreamRepeater,
setLiveStreamTranscoderList, setMediaCasterPlay, setMergeOnMetadata,
setMetaDataProvider, setName, setName, setName, setNetConnection, setOpen, setPlay,
setPlayer, setPublishAudioCodecId, setPublisherStream, setPublishVideoCodecId,
setQueryStr, setReceiveAudio, setReceiveVideo, setReceiveVideoFPS, setRecord,
setRTPStream, setSendPlayStopLogEvent, setSendPublishStopLogEvent,
setSendRecordStopLogEvent, setSrc, setStreamType, setTranscodeResult, setVideoSize,
setVideoTC, setVideoTC, shutdown, size, startAudioPacket, startDataPacket,
startPublishing, startVideoPacket, stopName, stopPublishing, switchName, trim,
unregisterCallback, unregisterOnPlayStatus, unregisterOnStatus,
updateLoggingDuration, updateLoggingValues

```

## Fields

---

(continued from last page)

## sinfo

```
public static java.util.Map sinfo
```

Deprecated.

---

## p

```
public static java.lang.String p
```

Deprecated.

## Constructors

### MediaStreamBase

```
public MediaStreamBase()
```

## Methods

### init

```
public void init(MediaStreamMap parent,  
                 int src,  
                 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="#">appInstance</a>
protected	<a href="#">dvrRecorders</a>
protected	<a href="#">liveStreamPacketizers</a>
public static final	<a href="#">MAXSTREAMINDEX</a> Value: <b>65536</b>
protected	<a href="#">mediaStreamListeners</a>
protected	<a href="#">nameGroupDelayRemoveMap</a>
protected	<a href="#">nameGroupId</a>
protected	<a href="#">nameGroupRemoveDelayTime</a>
protected	<a href="#">nameGroups</a>
protected	<a href="#">nextStreamId</a>
protected	<a href="#">packetizerLicenses</a>
protected	<a href="#">streamLicenses</a>
protected	<a href="#">streamLock</a>
protected	<a href="#">streamNames</a>
protected	<a href="#">streamNamesLock</a>
protected	<a href="#">streamNameToGroup</a>

protected	<a href="#">streams</a>
-----------	-------------------------

## Constructor Summary

public	<a href="#">MediaStreamMap(IApplicationInstance appInstance)</a> Create empty MediaStreamMap collection
--------	--

## Method Summary

LicenseHolder	<a href="#">addLicense(ILiveStreamPacketizer liveStreamPacketizer, int licenseType)</a>
LicenseHolder	<a href="#">addLicense(IMediaStream stream, int licenseType)</a>
void	<a href="#">addMediaStreamListener(IMediaStreamNotify mediaStreamListener)</a> Add a media stream listener.
MediaStreamMapGroup	<a href="#">addNameGroup(MediaStreamMapGroup newGroup)</a>
void	<a href="#">broadcastPlayMessage(IMediaStream stream, long timecode, java.nio.ByteBuffer msg, int objectEncoding)</a> Send a broadcast message to all play stream that are listening to this live published stream.
int	<a href="#">broadcastGetObjectEncoding(IMediaStream stream)</a> Get the minimum object encoding level for the clients playing this stream.
void	<a href="#">clearStreamName(String name)</a> Unregister a published live media stream name.
void	<a href="#">clearStreamName(String name, IMediaStream stream)</a> Unregister a published live media stream name.
void	<a href="#">doWatchdog()</a>
<a href="#">IApplicationInstance</a>	<a href="#">getAppInstance()</a> Get the parent applicationInstance.
String	<a href="#">getAppName()</a> Get the name of the parent application.
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(String streamName, String recorderName, boolean doCreate)</a> Get a DVR recorder by name and recorder name
java.util.List	<a href="#">getDvrRecorders()</a> Returns a list of ILiveStreamDvrRecorder objects
<a href="#">ILiveStreamPacketizer</a>	<a href="#">getLiveStreamPacketizer(String streamName, String packetizerName, boolean doCreate)</a> Get a live stream packetizer by name and packetizer id

Object	<a href="#"><u>getLiveStreamPacketizerLock()</u></a> Get the lock to the live stream packetizer system
MediaStreamMapGroup	<a href="#"><u>getNameGroupByGroupName</u>(String groupName)</a>
int	<a href="#"><u>getNameGroupRemoveDelayTime()</u></a>
java.util.Set	<a href="#"><u>getNameGroups()</u></a>
java.util.Set	<a href="#"><u>getNameGroups</u>(String streamName)</a>
java.util.Set	<a href="#"><u>getNameGroupStreamNames</u>(String streamName)</a>
long	<a href="#"><u>getNextNameGroupId</u>(MediaStreamMapGroup newGroup)</a>
int	<a href="#"><u>getNextStreamIndex()</u></a> Reserve a clientless stream id for a new media stream.
int	<a href="#"><u>getNextStreamIndex</u>(IClient client)</a> Reserve a stream for a client connection.
int	<a href="#"><u>getNextStreamIndex</u>(com.wowza.wms.netconnection.INetConnection netConnection)</a> 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>(IClient client, int index)</a> Get a media stream reference by stream id.
<a href="#"><u>IMediaStream</u></a>	<a href="#"><u>getStream</u>(IClient client, int index, boolean doCreate)</a> Get a media stream reference by stream id.
<a href="#"><u>IMediaStream</u></a>	<a href="#"><u>getStream</u>(com.wowza.wms.netconnection.INetConnection netConnection, int index)</a> 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>(com.wowza.wms.netconnection.INetConnection netConnection, int index, boolean doCreate)</a> Get a media stream reference by stream id.
<a href="#"><u>IMediaStream</u></a>	<a href="#"><u>getStream</u>(String name)</a> Get a media stream by stream name.
<a href="#"><u>IMediaStream</u></a>	<a href="#"><u>getStreamClientless</u>(int index, String streamTypeStr)</a> 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

<code>java.util.List</code>	<code><a href="#">getStreams()</a></code> Returns a list of IMediaStream objects
<code><a href="#">IVHost</a></code>	<code><a href="#">getVHost()</a></code> Get the parent vHost.
<code>void</code>	<code><a href="#">notifyMediaStreamCreate(IMediaStream mediaStream)</a></code> Notify all media stream listeners that a new media stream object has been created.
<code>void</code>	<code><a href="#">notifyMediaStreamDestroy(IMediaStream mediaStream)</a></code> Notify all media stream listeners that a media stream object is being destroyed.
<code>void</code>	<code><a href="#">notifyPlayPublish(IMediaStream stream)</a></code> Notify all play streams that are listening to this stream that the stream is going into a state of publish (NetStream.Play.PublishNotify).
<code>void</code>	<code><a href="#">notifyPlayUnpublish(IMediaStream stream)</a></code> Notify all play streams that are listening to this stream that the stream is going into a state of unpublished (NetStream.Play.UnpublishNotify).
<code>void</code>	<code><a href="#">notifyPlayUnpublish(String streamName)</a></code> Notify all play streams that are listening to this stream name that the stream is going into a state of unpublished (NetStream.Play.UnpublishNotify).
<code>void</code>	<code><a href="#">removeDvrRecorder(String streamName)</a></code>
<code><a href="#">ILiveStreamDvrRecorder</a></code>	<code><a href="#">removeDvrRecorder(String streamName, String recorderName)</a></code> Remove DVR Recorder
<code>void</code>	<code><a href="#">removeLiveStreamPacketizer(String streamName)</a></code> Remove all live stream packetizers for this stream name
<code><a href="#">ILiveStreamPacketizer</a></code>	<code><a href="#">removeLiveStreamPacketizer(String streamName, String packetizerName)</a></code> Remove live stream packetizer
<code>void</code>	<code><a href="#">removeMediaStreamListener(IMediaStreamNotify mediaStreamListener)</a></code> Remove a media stream listener.
<code>MediaStreamMapGroup</code>	<code><a href="#">removeNameGroup(int groupId)</a></code>
<code>MediaStreamMapGroup</code>	<code><a href="#">removeNameGroup(MediaStreamMapGroup nameGroup)</a></code>
<code>void</code>	<code><a href="#">removeStream(IClient client, int index)</a></code> Remove a stream associated with a client connection
<code>void</code>	<code><a href="#">removeStream(com.wowza.wms.netconnection.INetConnection netConnection, int index)</a></code> Remove a stream associated with a netConnection object
<code>void</code>	<code><a href="#">removeStream(int index)</a></code> Remove a clientless media stream
<code>void</code>	<code><a href="#">setNameGroupRemoveDelayTime(int nameGroupRemoveDelayTime)</a></code>
<code>void</code>	<code><a href="#">setStreamName(IMediaStream stream, String name)</a></code> Insert live media stream into the mediaStreamMap by name.

long	<a href="#">streamToIndex(<i>IMediaStream</i> stream)</a>
	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`

## Fields

### **MAXSTREAMINDEX**

`public static final int MAXSTREAMINDEX`

Constant value: **65536**

### **streams**

`protected java.util.Map streams`

### **streamNames**

`protected java.util.Map streamNames`

### **streamLicenses**

`protected java.util.Map streamLicenses`

### **packetizerLicenses**

`protected java.util.Map packetizerLicenses`

### **streamNameToGroup**

`protected java.util.Map streamNameToGroup`

### **nameGroups**

`protected java.util.List nameGroups`

(continued from last page)

**nameGroupId**

```
protected long nameGroupId
```

---

**nameGroupRemoveDelayTime**

```
protected int nameGroupRemoveDelayTime
```

---

**nameGroupDelayRemoveMap**

```
protected java.util.Map nameGroupDelayRemoveMap
```

---

**streamLock**

```
protected edu.emory.mathcs.backport.java.util.concurrent.locks.WMSReadWriteLock  
streamLock
```

---

**streamNamesLock**

```
protected edu.emory.mathcs.backport.java.util.concurrent.locks.WMSReadWriteLock  
streamNamesLock
```

---

**appInstance**

```
protected com.wowza.wms.application.IApplicationInstance appInstance
```

---

**mediaStreamListeners**

```
protected java.util.List mediaStreamListeners
```

---

**nextStreamId**

```
protected java.util.concurrent.atomic.AtomicLong nextStreamId
```

---

**liveStreamPacketizers**

```
protected java.util.Map liveStreamPacketizers
```

(continued from last page)

## dvrRecorders

```
protected java.util.Map dvrRecorders
```

## Constructors

### MediaStreamMap

```
public MediaStreamMap(IAplicationInstance appInstance)
```

Create empty MediaStreamMap collection

**Parameters:**

appInstance - parent applicationInstance

## Methods

### getStreamListLock

```
public edu.emory.mathcs.backport.java.util.concurrent.locks.WMSReadWriteLock  
getStreamListLock()
```

Get the underlying read/write lock associated with the list of streams

**Returns:**

underlying read/write lock associated with the list of streams

---

### getStreamNameLock

```
public edu.emory.mathcs.backport.java.util.concurrent.locks.WMSReadWriteLock  
getStreamNameLock()
```

Get the underlying read/write lock associated with the stream names list

**Returns:**

underlying read/write lock associated with the stream names list

---

### getPublishStreamNames

```
public java.util.List getPublishStreamNames()
```

Returns a List of published stream names

**Returns:**

List of published stream names

---

### getStreams

```
public java.util.List getStreams()
```

Returns a list of IMediaStream objects

**Returns:**

list of IMediaStream objects

(continued from last page)

## 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:**

(continued from last page)

media stream object or null if does not exist

---

## getStream

```
public IMediaStream getStream(com.wowza.wms.netconnection.INetConnection  
netConnection,  
    int index,  
    boolean doCreate)
```

Get a media stream reference by stream id. If it does not exist, create a new one with the given id. If creating a new media stream object, its id must first be obtained by calling [getNextStreamIndex\(INetConnection\)](#).

**Parameters:**

netConnection - netConnection to remote server  
index - stream id  
doCreate - true to create if it does not exist

**Returns:**

media stream object

---

## getStream

```
public IMediaStream getStream(String name)
```

Get a media stream by stream name. Only published live streams are stored in the mediaStreamMap by name. This method is used to lookup a published live stream by name.

**Parameters:**

name - stream name

**Returns:**

media stream object or null if does not exist

---

## setStreamName

```
public void setStreamName(IMediaStream stream,  
                           String name)
```

Insert live media stream into the mediaStreamMap by name.

**Parameters:**

stream - media stream object  
name - media stream name

---

## clearStreamName

```
public void clearStreamName(String name)
```

Unregister a published live media stream name.

**Parameters:**

name - stream name

---

## streamToIndex

```
public long streamToIndex(IMediaStream stream)
```

Get the unique stream identifier for a given stream

(continued from last page)

**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:**

(continued from last page)

total number of streams stored in the mediaStreamMap

---

## removeStream

```
public void removeStream(com.wowza.wms.netconnection.INetConnection netConnection,  
    int index)
```

Remove a stream associated with a netConnection object

**Parameters:**

netConnection - net connection  
index - stream index

---

## removeStream

```
public void removeStream(IClient client,  
    int index)
```

Remove a stream associated with a client connection

**Parameters:**

client - client  
index - stream index

---

## removeStream

```
public void removeStream(int index)
```

Remove a clientless media stream

**Parameters:**

index - stream index

---

## getAppName

```
public String getAppName()
```

Get the name of the parent application.

**Returns:**

parent application name.

---

## getInstanceName

```
public String getInstanceName()
```

Get the name of the parent application instance.

**Returns:**

parent application instance 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)

---

(continued from last page)

## notifyPlayPublish

```
public void notifyPlayPublish(IMediaStream stream)
```

Notify all play streams that are listening to this stream that the stream is going into a state of publish (NetStream.Play.PublishNotify).

**Parameters:**

stream - live published stream that is being published

---

## addMediaStreamListener

```
public void addMediaStreamListener(IMediaStreamNotify mediaStreamListener)
```

Add a media stream listener. A media stream listener will receive the following events: onMediaStreamCreate, onMediaStreamDestroy.

**Parameters:**

mediaStreamListener - media stream listener

---

## removeMediaStreamListener

```
public void removeMediaStreamListener(IMediaStreamNotify mediaStreamListener)
```

Remove a media stream listener.

**Parameters:**

mediaStreamListener - media stream listener

---

## notifyMediaStreamCreate

```
public void notifyMediaStreamCreate(IMediaStream mediaStream)
```

Notify all media stream listeners that a new media stream object has been created.

**Parameters:**

mediaStream - new media stream object

---

## notifyMediaStreamDestroy

```
public void notifyMediaStreamDestroy(IMediaStream mediaStream)
```

Notify all media stream listeners that a media stream object is being destroyed.

**Parameters:**

mediaStream - media stream object being destroyed

---

## removeLiveStreamPacketizer

```
public void removeLiveStreamPacketizer(String streamName)
```

Remove all live stream packetizers for this stream name

**Parameters:**

streamName - streamName

(continued from last page)

## removeLiveStreamPacketizer

```
public ILiveStreamPacketizer removeLiveStreamPacketizer(String streamName,  
String packetizerName)
```

Remove live stream packetizer

**Parameters:**

streamName - stream name  
packetizerName - packetizer id

**Returns:**

live stream packetizer

---

## getLiveStreamPacketizerLock

```
public Object getLiveStreamPacketizerLock()
```

Get the lock to the live stream packetizer system

**Returns:**

lock to the live stream packetizer system

---

## getLiveStreamPacketizer

```
public ILiveStreamPacketizer getLiveStreamPacketizer(String streamName,  
String packetizerName,  
boolean doCreate)
```

Get a live stream packetizer by name and packetizer id

**Parameters:**

streamName - stream name  
packetizerName - packetizer id  
doCreate - create if does not exist

**Returns:**

live stream packetizer

---

## getNextNameGroupId

```
public long getNextNameGroupId(MediaStreamMapGroup newGroup)
```

---

## addNameGroup

```
public MediaStreamMapGroup addNameGroup(MediaStreamMapGroup newGroup)
```

---

## removeNameGroup

```
public MediaStreamMapGroup removeNameGroup(int groupId)
```

(continued from last page)

**removeNameGroup**

```
public MediaStreamMapGroup removeNameGroup(MediaStreamMapGroup nameGroup)
```

**getNameGroups**

```
public java.util.Set getNameGroups()
```

**getNameGroupByGroupName**

```
public MediaStreamMapGroup getNameGroupByGroupName(String groupName)
```

**getNameGroups**

```
public java.util.Set getNameGroups(String streamName)
```

**getNameGroupStreamNames**

```
public java.util.Set getNameGroupStreamNames(String streamName)
```

**getDvrRecorder**

```
public ILiveStreamDvrRecorder getDvrRecorder(String streamName,  
                                         String recorderName,  
                                         boolean doCreate)
```

Get a DVR recorder by name and recorder name

**Parameters:**

streamName - stream name  
recorderName - packetizer id  
doCreate - create if does not exist

**Returns:**

dvr recorder

**removeDvrRecorder**

```
public void removeDvrRecorder(String streamName)
```

**removeDvrRecorder**

```
public ILiveStreamDvrRecorder removeDvrRecorder(String streamName,  
                                              String recorderName)
```

Remove DVR Recorder

(continued from last page)

**Parameters:**

streamName - stream name  
recorderName - recorder name

**Returns:**

DVR Recorder

---

**getDvrRecorders**

```
public java.util.List getDvrRecorders()
```

Returns a list of ILiveStreamDvrRecorder objects

**Returns:**

list of ILiveStreamDvrRecorder objects

---

**addLicense**

```
public LicenseHolder addLicense(IMediaStream stream,  
int licenseType)
```

---

**addLicense**

```
public LicenseHolder addLicense(ILiveStreamPacketizer liveStreamPacketizer,  
int licenseType)
```

---

**doWatchdog**

```
public void doWatchdog()
```

---

**getNameGroupRemoveDelayTime**

```
public int getNameGroupRemoveDelayTime()
```

---

**setNameGroupRemoveDelayTime**

```
public void setNameGroupRemoveDelayTime(int nameGroupRemoveDelayTime)
```

---

Package

**com.wowza.wms.stream.livedvr**

## com.wowza.wms.stream.livedvr Interface IDvrStreamManagerActionNotify

All Subinterfaces:

[IDvrStreamManagerActionNotify2](#)

---

public interface **IDvrStreamManagerActionNotify**  
extends

IDvrStreamManagerActionNotify notify interface for dvr stores. See  
IAplicationInstance.addDvrStreamManagerListener(IDvrStreamManagerActionNotify listener)

### 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 IDvrStreamManagerActionNotify2

All Superinterfaces:

[IDvrStreamManagerActionNotify](#)

---

public interface **IDvrStreamManagerActionNotify2**

extends [IDvrStreamManagerActionNotify](#)

IDvrStreamManagerActionNotify notify interface for dvr stores. See  
IApplicationInstance.addDvrStreamManagerListener(IDvrStreamManagerActionNotify listener)

### 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.
void	<a href="#">onDvrStreamManagerInitStreamName</a> ( <a href="#">IDvrStreamManager</a> dvrManager, String streamName)

Methods inherited from interface [com.wowza.wms.stream.livedvr.IDvrStreamManagerActionNotify](#)

[onDvrStreamManagerCreate](#), [onDvrStreamManagerDestroy](#), [onDvrStreamManagerInit](#)

### Methods

#### onDvrStreamManagerCreate

public void **onDvrStreamManagerCreate**([IDvrStreamManager](#) dvrMgr)

Called when DVR stream manager created but before it is initialized.

**Parameters:**

dvrMgr - DVR stream manager

---

#### onDvrStreamManagerInit

public void **onDvrStreamManagerInit**([IDvrStreamManager](#) dvrMgr)

Called when DVR stream manager initialized.

**Parameters:**

dvrMgr - DVR stream manager

(continued from last page)

## onDvrStreamManagerDestroy

```
public void onDvrStreamManagerDestroy(IDvrStreamManager dvrMgr)
```

Called when DVR stream manager destroyed.

**Parameters:**

dvrMgr - DVR stream manager

---

## onDvrStreamManagerInitStreamName

```
public void onDvrStreamManagerInitStreamName(IDvrStreamManager dvrManager,  
String streamName)
```

## com.wowza.wms.stream.livedvr Interface ILiveStreamDvrRecorder

public interface **ILiveStreamDvrRecorder**  
extends

ILiveStreamDvrRecorder: DVR Recorder interface.

### Method Summary

boolean	<a href="#">canRecordAudio()</a> True if audio is being recorded
boolean	<a href="#">canRecordData()</a> True if data is being recorded
boolean	<a href="#">canRecordVideo()</a> True if video is being recorded
<a href="#">IApplicationInstance</a>	<a href="#">getAppInstance()</a> Get associated applicationInstance.
<a href="#">IDvrStreamManager</a>	<a href="#">getDvrManager()</a> Get DVR recorder's associated DVR stream manager
int	<a href="#">getDvrRecorderId()</a> Get the DVR recorder id
LicenseHolder	<a href="#">getLicenseHolder()</a>
<a href="#">WMSProperties</a>	<a href="#">getProperties()</a> Get properties
String	<a href="#">getRecordingName()</a>
<a href="#">IMediaStream</a>	<a href="#">getStream()</a> Get the current stream that is being recorded
void	<a href="#">handlePacket(<a href="#">IMediaStream</a> stream, <a href="#">AMFPacket</a> packet)</a> Called to handle an incoming packet
void	<a href="#">init(String streamName, String recorderName, <a href="#">IApplicationInstance</a> appInstance, DvrRecorderItem dvrRecorderItem)</a> Initialize DVR recorder.
boolean	<a href="#">isActive()</a> Is the DVR recorder active
boolean	<a href="#">isRecording()</a> Is this stream currently recording.
boolean	<a href="#">isRecordingPaused()</a> Is this stream currently paused from recording.

boolean	<a href="#">pauseRecording()</a> Request that stream recording pause.
void	<a href="#">resetStream(IMediaStream stream)</a> Called when something happens that forces the stream to reset
boolean	<a href="#">resumeRecording()</a> Request that stream recording resume.
void	<a href="#">setDvrRecorderId(int liveStreamId)</a> Set the DVR recorder id
void	<a href="#">setRecordAudio(boolean recordAudio)</a> Set to true to record audio
void	<a href="#">setRecordData(boolean recordVideo)</a> Set to true to record data
void	<a href="#">setRecordingName(String name)</a>
void	<a href="#">setRecordVideo(boolean recordVideo)</a> Set to true to record video
void	<a href="#">setStartRecordingOnStartup(boolean shouldStartRecordingOnStartup)</a> Set recording behavior of DVR Manager on startup.
boolean	<a href="#">shouldStartRecordingOnStartup()</a> Should DVR start recording when packets start flowing.
void	<a href="#">shutdown()</a> Called to shutdown the DVR recorder
<a href="#">IDvrStreamManager</a>	<a href="#">startRecording()</a> Request that stream recording start.
void	<a href="#">startStream(IMediaStream stream)</a> Called when the stream starts
boolean	<a href="#">stopRecording()</a> Request that stream recording stop.
void	<a href="#">touch(long timecode)</a> 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(ILiveStreamDvrRecorder recorder, String streamName)</a> Called when recorder created
void	<a href="#">onLiveStreamDvrRecorderDestroy(ILiveStreamDvrRecorder recorder)</a> Called when recorder destroyed
void	<a href="#">onLiveStreamDvrRecorderInit(ILiveStreamDvrRecorder recorder, String streamName)</a> 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(String recorderName, IMediaStream stream)</a>
	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="#">IMediaStream</a>	<a href="#"><code>getAndSetStartStream(IMediaStream stream)</code></a> Get the current stream that is being packetized
<a href="#">IApplicationInstance</a>	<a href="#"><code>getApplicationInstance()</code></a> Get the application instance associated with this live stream packetizer.
int	<a href="#"><code>getLiveStreamPacketizerId()</code></a> Get the live stream packetizer id
<a href="#">WMSProperties</a>	<a href="#"><code>getProperties()</code></a> Get properties
long	<a href="#"><code>getRepeaterLastSeqence()</code></a> Get the sequence number of the last added repeater item
void	<a href="#"><code>handlePacket(IMediaStream stream, AMFPacket packet)</code></a> Called to handle an incoming packet
void	<a href="#"><code>init(String streamName, String packetizerName, IApplicationInstance appInstance, LiveStreamPacketizerItem liveStreamPacketizerItem)</code></a> Initialize live stream packetizer
boolean	<a href="#"><code>isActive()</code></a> Is the live stream packetizer active
boolean	<a href="#"><code>isPacketizeAudio()</code></a> True if audio is being packetized
boolean	<a href="#"><code>isPacketizeData()</code></a> True if data is being packetized
boolean	<a href="#"><code>isPacketizeVideo()</code></a> True if video is being packetized
boolean	<a href="#"><code>isRepeaterEdge()</code></a> Is this packetizer a live repeater edge
void	<a href="#"><code>resetStream(IMediaStream stream)</code></a> Called when something happens that forces the stream to reset
void	<a href="#"><code>setLiveStreamPacketizerId(int id)</code></a> 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

---

**getRepeaterLastSeqence**

```
public long getRepeaterLastSeqence( )
```

Get the sequence number of the last added repeater item

**Returns:**

sequence number

---

**getLiveStreamPacketizerId**

```
public int getLiveStreamPacketizerId( )
```

Get the live stream packetizer id

**Returns:**

live stream packetizer id

---

**setLiveStreamPacketizerId**

```
public void setLiveStreamPacketizerId(int id)
```

Set the live stream packetizer id

**Parameters:**

id

---

**handlePacket**

```
public void handlePacket(IMediaStream stream,  
                        AMFPacket packet)
```

Called to handle an incoming packet

**Parameters:**

stream - stream

packet - packet

---

**startStream**

```
public void startStream(IMediaStream stream)
```

Called when the stream starts

**Parameters:**

stream - stream

---

**resetStream**

```
public void resetStream(IMediaStream stream)
```

Called when something happens that forces the stream to reset

**Parameters:**

(continued from last page)

**stream** - stream

---

**touch****public void touch(long timecode)**

Touch the stream to keep it active

**Parameters:**

timecode - timecode of touch in milliseconds

---

**isRepeaterEdge****public boolean isRepeaterEdge()**

Is this packetizer a live repeater edge

**Returns:**

true if live repeater edge

---

**setRepeaterEdge****public void setRepeaterEdge(boolean isRepeaterEdge)**

Set is live repeater edge

**Parameters:**

isRepeaterEdge - is live repeater edge

---

**isPacketizeAudio****public boolean isPacketizeAudio()**

True if audio is being packetized

**Returns:**

True if audio is being packetized

---

**setPacketizeAudio****public void setPacketizeAudio(boolean packetizeAudio)**

Set to true to packetize audio

**Parameters:**

packetizeAudio - true to packetize audio

---

**isPacketizeVideo****public boolean isPacketizeVideo()**

True if video is being packetized

**Returns:**

True if video is being packetized

---

(continued from last page)

## setPacketizeVideo

```
public void setPacketizeVideo(boolean packetizeVideo)
```

Set to true to packetize video

**Parameters:**

packetizeVideo - true to packetize video

---

## isPacketizeData

```
public boolean isPacketizeData()
```

True if data is being packetized

**Returns:**

True if data is being packetized

---

## setPacketizeData

```
public void setPacketizeData(boolean packetizeVideo)
```

Set to true to packetize data

**Parameters:**

packetizeVideo - true to packetize data

---

## getAndSetStartStream

```
public IMediaStream getAndSetStartStream(IMediaStream stream)
```

Get the current stream that is being packetized

**Returns:**

stream

---

## getApplicationInstance

```
public IApplicationInstance getApplicationInstance()
```

Get the application instance associated with this live stream packetizer.

**Returns:**

application instance

## com.wowza.wms.stream.livepacketizer Interface ILiveStreamPacketizerActionNotify

public interface ILiveStreamPacketizerActionNotify  
extends

ILiveStreamPacketizerActionNotify: notify interface for live stream packetizers. See  
IApplicationInstance.addLiveStreamPacketizerListener(ILiveStreamPacketizerActionNotify liveStreamPacketizerListener)

### Method Summary

void	<a href="#"><u>onLiveStreamPacketizerCreate(ILiveStreamPacketizer liveStreamPacketizer, String streamName)</u></a> Called when packetizer created
void	<a href="#"><u>onLiveStreamPacketizerDestroy(ILiveStreamPacketizer liveStreamPacketizer)</u></a> Called when packetizer destroyed
void	<a href="#"><u>onLiveStreamPacketizerInit(ILiveStreamPacketizer liveStreamPacketizer, String streamName)</u></a> 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	<b><a href="#">isLiveStreamPacketize</a></b> (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.

### Field Summary

public static final	<a href="#">TRANSCODERTYPE_LIVE</a>
	Value: 1
public static final	<a href="#">TRANSCODERTYPE_VOD</a>
	Value: 2

### Method Summary

void	<a href="#">close(<u>IMediaStream</u> stream)</a> Called when live stream transcoder is stream is closed
<a href="#">IApplicationInstance</a>	<a href="#">getAppInstance()</a> Get the application instance associated with this live stream transcoder.
String	<a href="#">getContextStr()</a> Get the streaming context for this live stream transcoder.
LicenseHolder	<a href="#">getLicenseHolder()</a> Get license holder.
LiveStreamTranscoderItem	<a href="#">getLiveStreamTranscoderItem()</a> Get the definition for live stream transcoder.
<a href="#">WMSProperties</a>	<a href="#">getProperties()</a> Get the user properties
String	<a href="#">getStreamName()</a> Get the stream name of the source stream.
String	<a href="#">getTranscoderName()</a> Get the live stream transcoder name
int	<a href="#">getTranscoderType()</a> Get the live stream transcoder type.
void	<a href="#">handleOnMetadata(<u>IMediaStream</u> stream, <u>AMFPacket</u> packet, long timecode, boolean isSetDataFrame)</a> Called for each new onMetaData packet
void	<a href="#">handlePacket(<u>IMediaStream</u> stream, <u>AMFPacket</u> packet)</a> Called for each new source packet

void	<code>init(String streamName, IMediaStream stream, String transcoderName, IApplicationInstance appInstance, LiveStreamTranscoderItem liveStreamTranscoderItem)</code> Called when live stream transcoder interface is initialized.
boolean	<code>isTemplateLoaded()</code> Is the transcoder template loaded.
boolean	<code>isTranscoderActive(long currTime)</code> Returns true if the transcoder is actively receiving packets
void	<code>resetStream(IMediaStream stream)</code> Called when source stream changes.
void	<code>setAppInstance(IApplicationInstance appInstance)</code> Set the application instance associated with this live stream transcoder.
void	<code>setLiveStreamTranscoderItem(LiveStreamTranscoderItem liveStreamTranscoderItem)</code> Set the definition for live stream transcoder.
void	<code>setStreamName(String streamName)</code> Set source stream name.
void	<code>setTranscoderName(String transcoderName)</code> Get the live stream transcoder name
void	<code>setTranscoderType(int transcoderType)</code> Set the live stream transcoder type.
void	<code>shutdown(IMediaStream stream)</code> Called when live stream transcoder is shutdown

## Fields

### TRANSCODERTYPE\_LIVE

```
public static final int TRANSCODERTYPE_LIVE
```

Constant value: 1

### TRANSCODERTYPE\_VOD

```
public static final int TRANSCODERTYPE_VOD
```

Constant value: 2

## Methods

### init

```
public void init(String streamName,
IMediaStream stream,
String transcoderName,
IApplicationInstance appInstance,
LiveStreamTranscoderItem liveStreamTranscoderItem)
```

---

(continued from last page)

Called when live stream transcoder interface is initialized.

**Parameters:**

streamName - stream name  
transcoderName - transcoder name  
appInstance - application instance  
liveStreamTranscoderItem - live stream transcoder config item

---

## handlePacket

```
public void handlePacket(IMediaStream stream,  
                        AMFPacket packet)
```

Called for each new source packet

**Parameters:**

stream - stream  
packet - packet

---

## handleOnMetadata

```
public void handleOnMetadata(IMediaStream stream,  
                            AMFPacket packet,  
                            long timecode,  
                            boolean isSetDataFrame)
```

Called for each new onMetaData packet

**Parameters:**

stream - stream  
packet - packet  
timecode - timecode (milliseconds)  
isSetDataFrame - is SetDataFrame call

---

## resetStream

```
public void resetStream(IMediaStream stream)
```

Called when source stream changes.

**Parameters:**

stream - stream interface

---

## close

```
public void close(IMediaStream stream)
```

Called when live stream transcoder is stream is closed

**Parameters:**

stream - stream interface

---

## shutdown

```
public void shutdown(IMediaStream stream)
```

Called when live stream transcoder is shutdown

**Parameters:**

(continued from last page)

**stream** - stream interface

---

## getStreamName

```
public String getStreamName( )
```

Get the stream name of the source stream.

**Returns:**

stream name of the source stream

---

## setStreamName

```
public void setStreamName(String streamName)
```

Set source stream name.

**Parameters:**

streamName - source stream name

---

## getTranscoderName

```
public String getTranscoderName( )
```

Get the live stream transcoder name

**Returns:**

live stream transcoder name

---

## setTranscoderName

```
public void setTranscoderName(String transcoderName)
```

Get the live stream transcoder name

**Parameters:**

transcoderName - live stream transcoder name

---

## getAppInstance

```
public IApplicationInstance getAppInstance( )
```

Get the application instance associated with this live stream transcoder.

**Returns:**

application instance associated with this live stream transcoder

---

## setAppInstance

```
public void setAppInstance(IApplicationInstance appInstance)
```

Set the application instance associated with this live stream transcoder.

**Parameters:**

appInstance - application instance associated with this live stream transcoder

(continued from last page)

## getLiveStreamTranscoderItem

```
public LiveStreamTranscoderItem getLiveStreamTranscoderItem()
```

Get the definition for live stream transcoder.

**Returns:**

definition for live stream transcoder

---

## setLiveStreamTranscoderItem

```
public void setLiveStreamTranscoderItem(LiveStreamTranscoderItem  
liveStreamTranscoderItem)
```

Set the definition for live stream transcoder.

**Parameters:**

liveStreamTranscoderItem - definition for live stream transcoder

---

## getProperties

```
public WMSProperties getProperties()
```

Get the user properties

**Returns:**

user properties

---

## getLicenseHolder

```
public LicenseHolder getLicenseHolder()
```

Get license holder.

**Returns:**

license holder

---

## isTranscoderActive

```
public boolean isTranscoderActive(long currTime)
```

Returns true if the transcoder is actively receiving packets

**Parameters:**

currTime - current timecode in milliseconds

**Returns:**

true if the transcoder is actively receiving packets

---

## isTemplateLoaded

```
public boolean isTemplateLoaded()
```

Is the transcoder template loaded.

**Returns:**

true if transcoder template is loaded.

## getContextStr

```
public String getContextStr()
```

Get the streaming context for this live stream transcoder.

**Returns:**

streaming context for this live stream transcoder

---

## getTranscoderType

```
public int getTranscoderType()
```

Get the live stream transcoder type. See ILiveStreamTranscoder.TRANSCODERTYPE\_\*

**Returns:**

live stream transcoder type

---

## setTranscoderType

```
public void setTranscoderType(int transcoderType)
```

Set the live stream transcoder type. See ILiveStreamTranscoder.TRANSCODERTYPE\_\*

**Parameters:**

transcoderType - live stream transcoder type

## com.wowza.wms.stream.livetranscoder Interface ILiveStreamTranscoderControl

public interface ILiveStreamTranscoderControl  
extends

Interface used to control if stream is transcoded. See  
IAplicationInstance.setLiveStreamTranscoderControl(ILiveStreamTranscoderControl)

### Method Summary

boolean	<a href="#">isLiveStreamTranscode</a> ( String transcoder, <a href="#">IMediaStream</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  
IAplicationInstance.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  
IAplicationInstance.addLiveStreamTranscoderListener(ILiveStreamTranscoderNotify)

### Method Summary

void	<a href="#">onLiveStreamTranscoderCreate(ILiveStreamTranscoder liveStreamTranscoder, IMediaStream stream)</a> Triggered when live stream transcoder is created.
void	<a href="#">onLiveStreamTranscoderDestroy(ILiveStreamTranscoder liveStreamTranscoder, IMediaStream stream)</a> Triggered when live stream transcoder is destroyed.
void	<a href="#">onLiveStreamTranscoderInit(ILiveStreamTranscoder liveStreamTranscoder, IMediaStream stream)</a> 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="#"><b>close()</b></a> Invoked on stream close
boolean	<a href="#"><b>isSendOnMetadata()</b></a> Get to send onMetadata event when stream starts
boolean	<a href="#"><b>play(Publisher publisher)</b></a> Invoked on play
boolean	<a href="#"><b>seek(long timecode)</b></a> Invoked on seek
boolean	<a href="#"><b>seek(long timecode, int seekType)</b></a> Invoked on seek
void	<a href="#"><b>setDuration(long duration)</b></a> Set target duration for playback (milliseconds)
void	<a href="#"><b>setRealTimeStartTime(long realTimeStartTime)</b></a> Set real start time (milliseconds)
void	<a href="#"><b>setSendOnMetadata(boolean sendOnMetadata)</b></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="#"><u>onPlaylistItemStart</u></a> ( <a href="#"><u>Stream</u></a> stream, <a href="#"><u>PlaylistItem</u></a> playlistItem) Invoked when playlist item playback is started
void	<a href="#"><u>onPlaylistItemStop</u></a> ( <a href="#"><u>Stream</u></a> stream, <a href="#"><u>PlaylistItem</u></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(String sName)</a>
Class constructor - A simple structure that maintains a list of playlist items.	

### Method Summary

void	<a href="#">addItem(String sName, int start, int length)</a> 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(Stream s)</a> Opens this playlist on the given stream...
void	<a href="#">removeItem(int index)</a> Remove an item from the list
void	<a href="#">setRepeat(boolean repeat)</a>

#### Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, 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(String name, int start, int length, int index)</a>
	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(int index)</a>
String	<a href="#">toString()</a>

#### Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, 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 ASA0 16KHz Mono
- 5 Nelly Moser ASA0 8KHz Mono
- 6 Nelly Moser ASA0
- 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="#"><u>addAudioData</u></a> (byte[] data, int offset, int len, long timecode)
------	---

void	<a href="#"><u>addAudioData</u></a> (byte[] data, int offset, int len, long timecode, java.util.Map extraData) Add audio data
void	<a href="#"><u>addAudioData</u></a> (byte[] data, int len, long timecode) Add audio data
void	<a href="#"><u>addAudioData</u></a> (byte[] data, long timecode) Add audio data
void	<a href="#"><u>addAudioDataInc</u></a> (byte[] data, int offset, int len)
void	<a href="#"><u>addDataData</u></a> (byte[] data, int offset, int len, long timecode)
void	<a href="#"><u>addDataData</u></a> (byte[] data, int offset, int len, long timecode, java.util.Map extraData) Add metadata
void	<a href="#"><u>addDataData</u></a> (byte[] data, int len, long timecode) Add metadata
void	<a href="#"><u>addDataData</u></a> (byte[] data, long timecode) Add metadata
void	<a href="#"><u>addDataDataInc</u></a> (byte[] data, int offset, int len)
void	<a href="#"><u>addVideoData</u></a> (byte[] data, int offset, int len, long timecode) Add video data
void	<a href="#"><u>addVideoData</u></a> (byte[] data, int offset, int len, long timecode, java.util.Map extraData) Add video data
void	<a href="#"><u>addVideoData</u></a> (byte[] data, int len, long timecode) Add video data
void	<a href="#"><u>addVideoData</u></a> (byte[] data, long timecode) Add video data
void	<a href="#"><u>addVideoDataInc</u></a> (byte[] data, int offset, int len)
void	<a href="#"><u>close</u></a> () Close the publisher
static Publisher	<a href="#"><u>createInstance</u></a> (IAplicationInstance appInstance)
static Publisher	<a href="#"><u>createInstance</u></a> (IVHost vhost, String applicationName)
static Publisher	<a href="#"><u>createInstance</u></a> (IVHost vhost, String applicationName, String appInstanceName)
void	<a href="#"><u>createStream</u></a> () Create underlying IMediaStream object if not already created
void	<a href="#"><u>flush</u></a> () Flush the packets from the input buffer to the output buffer

<a href="#"><u>IApplicationInstance</u></a>	<a href="#"><u>getAppInstance()</u></a>
String	<a href="#"><u>getFileExtension()</u></a> Get the file extension (default flv)
long	<a href="#"><u>getLastAudioTimecode()</u></a> Get last audio timecode written through this publisher (milliseconds).
long	<a href="#"><u>getLastDataTimecode()</u></a> Get last data timecode written through this publisher (milliseconds).
long	<a href="#"><u>getLastVideoTimecode()</u></a> Get last video timecode written through this publisher (milliseconds).
long	<a href="#"><u>getMaxTimecode()</u></a> Highest timecode written through this publisher (milliseconds).
<a href="#"><u>IMediaStream</u></a>	<a href="#"><u>getStream()</u></a> Get the media stream object
String	<a href="#"><u>getStreamType()</u></a>
boolean	<a href="#"><u>isPublishDataEvents()</u></a>
void	<a href="#"><u>publish(String streamName)</u></a> Publish a stream (null to stop publishing)
void	<a href="#"><u>publish(String streamName, String howToPublish)</u></a> Start publishing a stream (streamName = null to stop).
void	<a href="#"><u>setFileExtension(String fileExtension)</u></a> Set the file extension
void	<a href="#"><u>setPublishDataEvents(boolean publishDataEvents)</u></a>
void	<a href="#"><u>setStreamType(String streamType)</u></a> Set the stream type (default live)
void	<a href="#"><u>startAudioData(int len, long timecode)</u></a>
void	<a href="#"><u>startAudioData(int len, long timecode, java.util.Map extraData)</u></a>
void	<a href="#"><u>startDataData(int len, long timecode)</u></a>
void	<a href="#"><u>startDataData(int len, long timecode, java.util.Map extraData)</u></a>
void	<a href="#"><u>startVideoData(int len, long timecode)</u></a>
void	<a href="#"><u>startVideoData(int len, long timecode, java.util.Map extraData)</u></a>
void	<a href="#"><u>unpublish()</u></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)

(continued from last page)

**Parameters:**

streamName - stream name

**getFileExtension****public String getFileExtension( )**

Get the file extension (default flv)

**Returns:**

file extension

**setFileExtension****public void setFileExtension(String fileExtension)**

Set the file extension

**Parameters:**

fileExtension - file extension

**getAppInstance****public IApplicationInstance getAppInstance( )****unpublish****public void unpublish( )****createStream****public void createStream( )**

Create underlying IMediaStream object if not already created

**publish****public void publish(String streamName,  
String howToPublish)**

Start publishing a stream (streamName = null to stop). Valid howToPublish values are (live, record, append)

**Parameters:**

streamName - stream name

howToPublish - publish method (live, record, append)

**addVideoData****public void addVideoData(byte[ ] data,  
long timecode)**

Add video data

**Parameters:**

data - data

(continued from last page)

`timecode - absolute timecode (milliseconds)`

---

## addVideoData

```
public void addVideoData(byte[] data,  
                         int len,  
                         long timecode)
```

Add video data

**Parameters:**

`data - data`  
`len - data length`  
`timecode - absolute timecode (milliseconds)`

---

## addVideoData

```
public void addVideoData(byte[] data,  
                         int offset,  
                         int len,  
                         long timecode)
```

Add video data

**Parameters:**

`data - data`  
`offset - offset`  
`len - data length`  
`timecode - absolute timecode (milliseconds)`

---

## addVideoData

```
public void addVideoData(byte[] data,  
                         int offset,  
                         int len,  
                         long timecode,  
                         java.util.Map extraData)
```

Add video data

**Parameters:**

`data - data`  
`offset - offset`  
`len - data length`  
`timecode - absolute timecode (milliseconds)`  
`extraData - extra data`

---

## startVideoData

```
public void startVideoData(int len,  
                           long timecode)
```

---

## startVideoData

```
public void startVideoData(int len,  
                           long timecode,  
                           java.util.Map extraData)
```

(continued from last page)

---

**addVideoDataInc**

```
public void addVideoDataInc(byte[] data,
    int offset,
    int len)
```

---

**startAudioData**

```
public void startAudioData(int len,
    long timecode)
```

---

**startAudioData**

```
public void startAudioData(int len,
    long timecode,
    java.util.Map extraData)
```

---

**addAudioDataInc**

```
public void addAudioDataInc(byte[] data,
    int offset,
    int len)
```

---

**addAudioData**

```
public void addAudioData(byte[] data,
    long timecode)
```

Add audio data

**Parameters:**

data - data  
timecode - absolute timecode (milliseconds)

---

**addAudioData**

```
public void addAudioData(byte[] data,
    int len,
    long timecode)
```

Add audio data

**Parameters:**

data - data  
len - data length  
timecode - absolute timecode (milliseconds)

---

(continued from last page)

## addAudioData

```
public void addAudioData(byte[] data,
    int offset,
    int len,
    long timecode)
```

---

## addAudioData

```
public void addAudioData(byte[] data,
    int offset,
    int len,
    long timecode,
    java.util.Map extraData)
```

Add audio data

**Parameters:**

data - data  
len - data length  
offset - offset  
timecode - absolute timecode (milliseconds)

---

## addDataData

```
public void addDataData(byte[] data,
    long timecode)
```

Add metadata

**Parameters:**

data - data  
timecode - absolute timecode (milliseconds)

---

## addDataData

```
public void addDataData(byte[] data,
    int len,
    long timecode)
```

Add metadata

**Parameters:**

data - data  
len - data length  
timecode - absolute timecode (milliseconds)

---

## addDataData

```
public void addDataData(byte[] data,
    int offset,
    int len,
    long timecode)
```

---

(continued from last page)

## addDataData

```
public void addDataData(byte[] data,  
                      int offset,  
                      int len,  
                      long timecode,  
                      java.util.Map extraData)
```

Add metadata

### Parameters:

data - data  
offset - offset  
len - data length  
timecode - absolute timecode (milliseconds)

---

## startDataData

```
public void startDataData(int len,  
                        long timecode)
```

---

## startDataData

```
public void startDataData(int len,  
                        long timecode,  
                        java.util.Map extraData)
```

---

## addDataDataInc

```
public void addDataDataInc(byte[] data,  
                          int offset,  
                          int len)
```

---

## flush

```
public void flush()
```

Flush the packets from the input buffer to the output buffer

---

## close

```
public void close()
```

Close the publisher

---

## getMaxTimecode

```
public long getMaxTimecode()
```

Highest timecode written through this publisher (milliseconds).

### Returns:

highest timecode written through this publisher (milliseconds)

## getLastAudioTimecode

```
public long getLastAudioTimecode()
```

Get last audio timecode written through this publisher (milliseconds).

**Returns:**

last audio timecode written through this publisher (milliseconds).

---

## getLastVideoTimecode

```
public long getLastVideoTimecode()
```

Get last video timecode written through this publisher (milliseconds).

**Returns:**

last video timecode written through this publisher (milliseconds).

---

## getLastDataTimecode

```
public long getLastDataTimecode()
```

Get last data timecode written through this publisher (milliseconds).

**Returns:**

last data timecode written through this publisher (milliseconds).

---

## isPublishDataEvents

```
public boolean isPublishDataEvents()
```

---

## setPublishDataEvents

```
public void setPublishDataEvents(boolean publishDataEvents)
```

## com.wowza.wms.stream.publish Class PublishingProviderBase

```
java.lang.Object
+-com.wowza.wms.stream.publish.PublishingProviderBase
```

### Direct Known Subclasses:

[PublishingProviderMediaReader](#), [PublishingProviderLive](#)

```
public class PublishingProviderBase
extends Object
```

PublishingProviderBase: Base class for publishing providers.

### Field Summary

protected	<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(boolean sendOnMetadata)</a> 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
--

### 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
+- com.wowza.wms.stream.publish.PublishingProviderBase
  +- com.wowza.wms.stream.publish.PublishingProviderLive
```

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="#">isStopIfStreamMissing()</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="#"><u>setStartOnPreviousKeyFrame</u></a> (boolean startOnPreviousKeyFrame)
void	<a href="#"><u>setStopIfStreamMissing</u></a> (boolean stopIfStreamMissing)

**Methods inherited from class** [com.wowza.wms.stream.publish.PublishingProviderBase](#)[isSendOnMetadata](#), [setSendOnMetadata](#)**Methods inherited from class** [java.lang.Object](#)

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

**Methods inherited from interface** [com.wowza.wms.stream.publish.IPublishingProvider](#)[close](#), [isSendOnMetadata](#), [play](#), [seek](#), [seek](#), [setDuration](#), [setRealTimeStartTime](#), [setSendOnMetadata](#)

## Constructors

### PublishingProviderLive

```
public PublishingProviderLive(Publisher publisher,
                           long timeoffset,
                           String streamName)
```

## Methods

### close

```
public void close()
```

---

### play

```
public boolean play(Publisher publisher)
```

---

### seek

```
public boolean seek(long timecode)
```

---

### seek

```
public boolean seek(long timecode,
                     int seekType)
```

---

### **getDuration**

```
public long getDuration()
```

---

### **setDuration**

```
public void setDuration(long duration)
```

---

### **setRealTimeStartTime**

```
public void setRealTimeStartTime(long realTimeStartTime)
```

---

### **isStartOnPreviousKeyFrame**

```
public boolean isStartOnPreviousKeyFrame()
```

---

### **setStartOnPreviousKeyFrame**

```
public void setStartOnPreviousKeyFrame(boolean startOnPreviousKeyFrame)
```

---

### **getStartOnPreviousBufferTime**

```
public long getStartOnPreviousBufferTime()
```

---

### **setStartOnPreviousBufferTime**

```
public void setStartOnPreviousBufferTime(long startOnPreviousBufferTime)
```

---

### **isStopIfStreamMissing**

```
public boolean isStopIfStreamMissing()
```

---

### **setStopIfStreamMissing**

```
public void setStopIfStreamMissing(boolean stopIfStreamMissing)
```

---

## com.wowza.wms.stream.publish Class PublishingProviderMediaReader

```
java.lang.Object
+- com.wowza.wms.stream.publish.PublishingProviderBase
  +- com.wowza.wms.stream.publish.PublishingProviderMediaReader
```

All Implemented Interfaces:  
[IPublishingProvider](#)

---

```
public class PublishingProviderMediaReader
extends PublishingProviderBase
implements IPublishingProvider
```

PublishingProviderMediaReader: Publishing provider for media reader based stream (vod).

### Field Summary

<code>public static final</code>	<a href="#">ONMETADATA_FILTER_DURATION</a>
	Value: 1
<code>public static final</code>	<a href="#">ONMETADATA_FILTER_NONE</a>
	Value: 0

### Fields inherited from class [com.wowza.wms.stream.publish.PublishingProviderBase](#)

[sendOnMetadata](#)

### Constructor Summary

<code>public</code>	<code>PublishingProviderMediaReader(Publisher publisher, long timeoffset, String streamName)</code>
	Constructor

### Method Summary

<code>void</code>	<a href="#">close()</a>
<code>byte[]</code>	<a href="#">filterMetadata(byte[] dataBytes)</a>
<code>long</code>	<a href="#">getDuration()</a> Get the target playback duration (milliseconds)
<code>long</code>	<a href="#">getOnMetadataFilter()</a> Filter that control what metadata items are removed from onMetaData event.
<code>double</code>	<a href="#">getSpeed()</a>
<code>boolean</code>	<a href="#">play(Publisher publisher)</a>

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="#">setOnMetadataFilter</a> (long onMetadataFilter) Filter that control what metadata items are removed from onMetaData event.
void	<a href="#">setRealTimeStartTime</a> (long realTimeStartTime)
void	<a href="#">setSpeed</a> (double speed)

Methods inherited from class [com.wowza.wms.stream.publish.PublishingProviderBase](#)

[isSendOnMetadata](#), [setSendOnMetadata](#)

Methods inherited from class [java.lang.Object](#)

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#)

Methods inherited from interface [com.wowza.wms.stream.publish.IPublishingProvider](#)

[close](#), [isSendOnMetadata](#), [play](#), [seek](#), [seek](#), [setDuration](#), [setRealTimeStartTime](#), [setSendOnMetadata](#)

## Fields

### ONMETADATA\_FILTER\_NONE

public static final long **ONMETADATA\_FILTER\_NONE**

Constant value: 0

### ONMETADATA\_FILTER\_DURATION

public static final long **ONMETADATA\_FILTER\_DURATION**

Constant value: 1

## Constructors

### PublishingProviderMediaReader

```
public PublishingProviderMediaReader(Publisher publisher,
                                     long timeoffset,
                                     String streamName)
```

Constructor

**Parameters:**

(continued from last page)

publisher - publisher  
timeoffset - timeoffset (milliseconds)  
streamName - stream name

## Methods

### seek

```
public boolean seek(long timecode)
```

---

### seek

```
public boolean seek(long timecode,  
                    int seekType)
```

---

### filterMetadata

```
public byte[] filterMetadata(byte[] dataBytes)
```

---

### play

```
public boolean play(Publisher publisher)
```

---

### close

```
public void close()
```

---

### getDuration

```
public long getDuration()
```

Get the target playback duration (milliseconds)

**Returns:**

playback duration (milliseconds)

---

### setDuration

```
public void setDuration(long duration)
```

---

### setRealTimeStartTime

```
public void setRealTimeStartTime(long realTimeStartTime)
```

---

**getSpeed**

```
public double getSpeed()
```

---

**setSpeed**

```
public void setSpeed(double speed)
```

---

**getOnMetadataFilter**

```
public long getOnMetadataFilter()
```

Filter that control what metadata items are removed from onMetaData event. Default is ONMETADATA\_FILTER\_DURATION. Value is a bit mask of ONMETADATA\_FILTER\_\* value.

**Returns:**

control what metadata items are removed from onMetaData event

---

**setOnMetadataFilter**

```
public void setOnMetadataFilter(long onMetadataFilter)
```

Filter that control what metadata items are removed from onMetaData event. Default is ONMETADATA\_FILTER\_DURATION. Value is a bit mask of ONMETADATA\_FILTER\_\* value.

**Parameters:**

onMetadataFilter - control what metadata items are removed from onMetaData event

## com.wowza.wms.stream.publish Class Stream

```
java.lang.Object
+-com.wowza.wms.stream.publish.Stream
```

All Implemented Interfaces:  
Runnable

```
public class Stream
extends Object
implements Runnable
```

### Constructor Summary

public	<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 Stream	<a href="#">createInstance(IApplicationInstance appInstance, String sName)</a> Use this to create a named Stream on an application instance.
static Stream	<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 Stream	<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.
PlaylistItem	<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="#"><u>getName()</u></a> Returns the name of the playlist stream - the client would play this stream by this name.
java.util.List	<a href="#"><u>getPlaylist()</u></a> Get the current playlist
int	<a href="#"><u>getPollingInterval()</u></a> Get the polling interval (milliseconds)
<a href="#"><u>Publisher</u></a>	<a href="#"><u>getPublisher()</u></a>
boolean	<a href="#"><u>getRepeat()</u></a> Use this to determine if the playlist is auto-repeating
long	<a href="#"><u>getStartLiveOnPreviousBufferTime()</u></a> Get time in milliseconds to go back in live stream buffer to get previous key frame
int	<a href="#"><u>getTimeOffsetBetweenItems()</u></a> Get time in milliseconds to add to stream time between playlist items (default is zero)
double	<a href="#"><u>getVODSpeed()</u></a>
boolean	<a href="#"><u>isMoveToNextIfLiveStreamMissing()</u></a> If true, will move to next playlist item if live stream is missing or is unpublished.
boolean	<a href="#"><u>isSendOnMetadata()</u></a> True if sending onMetadata events
boolean	<a href="#"><u>isStartLiveOnPreviousKeyFrame()</u></a> Set to true to start live streams on most recent key frame (smoother switching)
boolean	<a href="#"><u>isSwitchLog()</u></a> Log when a playlist switch occurs
boolean	<a href="#"><u>isTimesInMilliseconds()</u></a> If true start time and duration are milliseconds.
boolean	<a href="#"><u>isUnpublishOnEnd()</u></a> If true, stop publishing if end of playlist and repeat is false
void	<a href="#"><u>next()</u></a>
void	<a href="#"><u>next(int n)</u></a>
void	<a href="#"><u>play(int n)</u></a>
boolean	<a href="#"><u>play(String sPlaylist)</u></a> Add a media item to the playlist as defined by an XML file ..
boolean	<a href="#"><u>play(String name, int start, int length, boolean reset)</u></a> Adds a media source item to this playlist -
void	<a href="#"><u>previous()</u></a>
void	<a href="#"><u>previous(int n)</u></a>

boolean	<a href="#"><u>removeFromPlaylist(int index)</u></a> Remove item from playlist based on index.
boolean	<a href="#"><u>removeFromPlaylist(String name)</u></a> Remove all items matching the given stream name from the playlist.
void	<a href="#"><u>removeListener(IStreamActionNotify listener)</u></a> Remove a listener
void	<a href="#"><u>run()</u></a> Overridden from class Runnable ....
void	<a href="#"><u>setMoveToNextIfLiveStreamMissing(boolean moveToNextIfLiveStreamMissing)</u></a> If true, will move to next playlist item if live stream is missing or is unpublished.
void	<a href="#"><u>setPollingInterval(int pollingInterval)</u></a> Set the polling interval (milliseconds)
void	<a href="#"><u>setRepeat(boolean repeat)</u></a> Use this to make the playlist repeat or not...
void	<a href="#"><u>setSendOnMetadata(boolean sendOnMetadata)</u></a> True if sending onMetadata events
void	<a href="#"><u>setStartLiveOnPreviousBufferTime(long startLiveOnPreviousBufferTime)</u></a> Set time in milliseconds to go back in live stream buffer to get previous key frame
void	<a href="#"><u>setStartLiveOnPreviousKeyFrame(boolean startLiveOnPreviousKeyFrame)</u></a> Set to true to start live streams on most recent key frame (smoother switching)
void	<a href="#"><u>setSwitchLog(boolean switchLog)</u></a> Log when a playlist switch occurs
void	<a href="#"><u>setTimeOffsetBetweenItems(int timeOffsetBetweenItems)</u></a> Set time in milliseconds to add to stream time between playlist items (default is zero)
void	<a href="#"><u>setTimesInMilliseconds(boolean timesInMilliseconds)</u></a> If true start time and duration and are milliseconds.
void	<a href="#"><u>setUnpublishOnEnd(boolean stopOnEnd)</u></a> If true, stop publishing if end of playlist and repeat is false
void	<a href="#"><u>setVODSpeed(double vodSpeed)</u></a>

**Methods inherited from class java.lang.Object**

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait`

**Methods inherited from interface java.lang.Runnable**

`run`

**Constructors**

(continued from last page)

## Stream

```
public Stream()
```

## Methods

### createInstance

```
public static Stream createInstance(IVHost vhost,  
String applicationName,  
String sName)
```

Use this to create a named Stream on the default instance (`_definst_`) of an application on a particular VHost.

**Parameters:**

vhost -- Virtual Host  
applicationName -- Application name  
sName -- Name of Stream

**Returns:**

stream interface

### createInstance

```
public static Stream createInstance(IVHost vhost,  
String applicationName,  
String appInstanceName,  
String sName)
```

Use this to create a named Stream on the non-default instance of an application on a particular VHost.

**Parameters:**

vhost -- Virtual Host  
applicationName -- Application name  
appInstanceName -- Instance name  
sName -- Name of Stream

**Returns:**

stream interface

### getLock

```
public Object getLock()
```

Get the synchronization lock for this interface.

**Returns:**

synchronization lock for this interface

### createInstance

```
public static Stream createInstance(IApplicationInstance appInstance,  
String sName)
```

Use this to create a named Stream on an application instance.

**Parameters:**

(continued from last page)

appInstance -- Application instance  
 sName -- Name of Stream

**Returns:**

stream interface

**run****public void run()**

Overridden from class Runnable .... do NOT call this directly. This method handles switching between media sources in the playlist.

**getPlaylist****public java.util.List getPlaylist()**

Get the current playlist

**Returns:**

current playlist

**getCurrentItem****public PlaylistItem getCurrentItem()**

Get the currently playing playlist item

**Returns:**

currently playing playlist item

**play**

```
public boolean play(String name,
                   int start,
                   int length,
                   boolean reset)
```

Adds a media source item to this playlist -

**Parameters:**

- name -- name of media item
- start -- where to start playing the item. (-2 implies play a live stream)
- length -- how much of the item to play (-1 implies play the entire file or live stream)
- reset -- if true, will begin a new playlist, otherwise items are appended

**Returns:**

- returns true if item was added successfully ...otherwise false.

**addToPlaylist**

```
public boolean addToPlaylist(int index,
                            String name,
                            int start,
                            int length)
```

Inserts a media source item to this playlist at a particular index, without interruption.

**Parameters:**

- index -- insertion index

(continued from last page)

`name` -- name of media item being inserted  
`start` -- where to start playing the item. (-2 implies play a live stream)  
`length` -- how much of the item to play (-1 implies play the entire file or live stream)

**Returns:**

- returns true if item was inserted successfully ...otherwise false. this should not interrupt anything that might be currently playing.... will do nothing if existing item does not exist.. if existing == "" insert at head of list

---

**addToPlaylist**

```
public boolean addToPlaylist(String existing,  
                           String name,  
                           int start,  
                           int length)
```

Insert item into playlist just after the first item in the playlist with a given name.

**Parameters:**

`existing` - name of playlist item in which to insert the item after  
`name` - name of new item  
`start` - where to start playing the item. (-2 implies play a live stream)  
`length` - how much of the item to play (-1 implies play the entire file or live stream)

**Returns:**

true if item inserted

---

**removeFromPlaylist**

```
public boolean removeFromPlaylist(int index)
```

Remove item from playlist based on index.

**Parameters:**

`index` - item index

**Returns:**

true if at least one item from removed

---

**removeFromPlaylist**

```
public boolean removeFromPlaylist(String name)
```

Remove all items matching the given stream name from the playlist. If the item is currently being played, it is not removed.

**Parameters:**

`name` - stream name

**Returns:**

true if at least one item from removed

---

**play**

```
public boolean play(String sPlaylist)
```

Add a media item to the playlist as defined by an XML file ..

**Parameters:**

`sPlaylist` -- the playlist XML definition file

(continued from last page)

**Returns:**

- returns true if item was added successfully ...otherwise false.

---

**close**

```
public void close()
```

Call this method when you have finished with the playlist object. It will terminate the playlist thread. The playlist object cannot be used after this call.

---

**closeAndWait**

```
public void closeAndWait()
```

Call this method when you have finished with the playlist object. It will terminate the playlist thread. The playlist object cannot be used after this call. This method will wait for the thread to exit.

---

**getName**

```
public String getName()
```

Returns the name of the playlist stream - the client would play this stream by this name.

**Returns:**

- the name of the playlist stream

---

**getRepeat**

```
public boolean getRepeat()
```

Use this to determine if the playlist is auto-repeating

**Returns:**

- true if repeating otherwise false.

---

**setRepeat**

```
public void setRepeat(boolean repeat)
```

Use this to make the playlist repeat or not...

**Parameters:**

- repeat -- true to repeat otherwise false

---

**play**

```
public void play(int n)
```

---

**next**

```
public void next(int n)
```

(continued from last page)

**next**

```
public void next()
```

**previous**

```
public void previous()
```

**previous**

```
public void previous(int n)
```

**getPollingInterval**

```
public int getPollingInterval()
```

Get the polling interval (milliseconds)

**Returns:**

polling interval (milliseconds)

**setPollingInterval**

```
public void setPollingInterval(int pollingInterval)
```

Set the polling interval (milliseconds)

**Parameters:**

pollingInterval - polling interval (milliseconds)

**getPublisher**

```
public Publisher getPublisher()
```

**addListener**

```
public void addListener(IStreamActionNotify listener)
```

Add a listener

**Parameters:**

listener - listener

**removeListener**

```
public void removeListener(IStreamActionNotify listener)
```

Remove a listener

**Parameters:**

(continued from last page)

---

listener - listener

---

**isSwitchLog**

```
public boolean isSwitchLog()
```

Log when a playlist switch occurs

**Returns:**

log when a playlist switch occurs

---

**setSwitchLog**

```
public void setSwitchLog(boolean switchLog)
```

Log when a playlist switch occurs

**Parameters:**

switchLog - log when a playlist switch occurs

---

**isSendOnMetadata**

```
public boolean isSendOnMetadata()
```

True if sending onMetadata events

**Returns:**

True if sending onMetadata events

---

**setSendOnMetadata**

```
public void setSendOnMetadata(boolean sendOnMetadata)
```

True if sending onMetadata events

**Parameters:**

sendOnMetadata - True if sending onMetadata events

---

**isTimesInMilliseconds**

```
public boolean isTimesInMilliseconds()
```

If true start time and duration are milliseconds. If false startTime and duration are in seconds.

**Returns:**

true start time and duration and are milliseconds

---

**setTimesInMilliseconds**

```
public void setTimesInMilliseconds(boolean timesInMilliseconds)
```

If true start time and duration and are milliseconds. If false startTime and duration are in seconds.

**Parameters:**

timesInMilliseconds - true start time and duration and are milliseconds

---

(continued from last page)

## isStartLiveOnPreviousKeyFrame

```
public boolean isStartLiveOnPreviousKeyFrame()
```

Set to true to start live streams on most recent key frame (smoother switching)

**Returns:**

true to start live streams on most recent key frame

---

## setStartLiveOnPreviousKeyFrame

```
public void setStartLiveOnPreviousKeyFrame(boolean startLiveOnPreviousKeyFrame)
```

Set to true to start live streams on most recent key frame (smoother switching)

**Parameters:**

startLiveOnPreviousKeyFrame - true to start live streams on most recent key frame

---

## getStartLiveOnPreviousBufferTime

```
public long getStartLiveOnPreviousBufferTime()
```

Get time in milliseconds to go back in live stream buffer to get previous key frame

**Returns:**

time in milliseconds to go back in live stream buffer to get previous key frame

---

## setStartLiveOnPreviousBufferTime

```
public void setStartLiveOnPreviousBufferTime(long startLiveOnPreviousBufferTime)
```

Set time in milliseconds to go back in live stream buffer to get previous key frame

**Parameters:**

startLiveOnPreviousBufferTime - time in milliseconds to go back in live stream buffer to get previous key frame

---

## getTimeOffsetBetweenItems

```
public int getTimeOffsetBetweenItems()
```

Get time in milliseconds to add to stream time between playlist items (default is zero)

**Returns:**

time in milliseconds to add to stream time between playlist items

---

## setTimeOffsetBetweenItems

```
public void setTimeOffsetBetweenItems(int timeOffsetBetweenItems)
```

Set time in milliseconds to add to stream time between playlist items (default is zero)

**Parameters:**

timeOffsetBetweenItems - time in milliseconds to add to stream time between playlist items

---

## isMoveToNextIfLiveStreamMissing

```
public boolean isMoveToNextIfLiveStreamMissing()
```

---

(continued from last page)

If true, will move to next playlist item if live stream is missing or is unpublished.

**Returns:**

move to next playlist item if live stream is missing or is unpublished

---

## **setMoveToNextIfLiveStreamMissing**

```
public void setMoveToNextIfLiveStreamMissing(boolean moveToNextIfLiveStreamMissing)
```

If true, will move to next playlist item if live stream is missing or is unpublished.

**Parameters:**

moveToNextIfLiveStreamMissing - move to next playlist item if live stream is missing or is unpublished

---

## **isUnpublishOnEnd**

```
public boolean isUnpublishOnEnd()
```

If true, stop publishing if end of playlist and repeat is false

**Returns:**

stop publishing if end of playlist and repeat is false

---

## **setUnpublishOnEnd**

```
public void setUnpublishOnEnd(boolean stopOnEnd)
```

If true, stop publishing if end of playlist and repeat is false

**Parameters:**

stopOnEnd - stop publishing if end of playlist and repeat is false

---

## **getVODSpeed**

```
public double getVODSpeed()
```

---

## **setVODSpeed**

```
public void setVODSpeed(double vodSpeed)
```

---

Package

**com.wowza.wms.timedtext-cea608**

## com.wowza.wms.timedtext.cea608 Class ClosedCaptionCEA608Utils

```
java.lang.Object
+-com.wowza.wms.timedtext.cea608.CEA608Constants
  +-com.wowza.wms.timedtext.cea608.ClosedCaptionCEA608Utils
```

### Direct Known Subclasses:

[ClosedCaptionCEA608Utils](#)

```
public class ClosedCaptionCEA608Utils
extends CEA608Constants
```

## Field Summary

public static final	<a href="#">CEA608_EXTENDED_FR_SP_MAP</a>
public static final	<a href="#">CEA608_EXTENDED_NA_MAP</a>
public static final	<a href="#">CEA608_EXTENDED_PORT_GERMAN_DANISH_MAP</a>
public static	<a href="#">CEA608_MUSICAL_NOTE</a>
public static	<a href="#">CEA608_SOLID_BLOCK</a>
public static final	<a href="#">CHANNEL1</a> Value: <b>0</b>
public static final	<a href="#">CHANNEL2</a> Value: <b>1</b>
public static final	<a href="#">COL_0</a> Value: <b>0</b>
public static final	<a href="#">COL_12</a> Value: <b>6</b>
public static final	<a href="#">COL_16</a> Value: <b>8</b>
public static final	<a href="#">COL_20</a> Value: <b>10</b>
public static final	<a href="#">COL_24</a> Value: <b>12</b>

public static final	<a href="#">COL_28</a>
	Value: <b>14</b>
public static final	<a href="#">COL_4</a>
	Value: <b>2</b>
public static final	<a href="#">COL_5</a>
	Value: <b>4</b>
public static final	<a href="#">COL0_BLUE</a>
	Value: <b>4</b>
public static final	<a href="#">COL0_CYAN</a>
	Value: <b>6</b>
public static final	<a href="#">COL0_GREEN</a>
	Value: <b>2</b>
public static final	<a href="#">COL0_HIBYTE</a>
	Move to set color (high-byte) [channel][row]
public static final	<a href="#">COL0_LOWBYTE</a>
	Move to set color (low-byte) [color][row]
public static final	<a href="#">COL0_MAGENTA</a>
	Value: <b>12</b>
public static final	<a href="#">COL0_RED</a>
	Value: <b>8</b>
public static final	<a href="#">COL0_WHITE</a>
	Value: <b>0</b>
public static final	<a href="#">COL0_YELLOW</a>
	Value: <b>10</b>
public static final	<a href="#">COLN_LOWBYTE</a>
public static final	<a href="#">CONTROLCODES_AOF</a>
	AOF Reserved (formerly Alarm Off) [channel]
public static final	<a href="#">CONTROLCODES_AON</a>
	AON Reserved (formerly Alarm On) [channel]
public static final	<a href="#">CONTROLCODES_BAO</a>
	BAO Background Black, Opaque [channel]
public static final	<a href="#">CONTROLCODES_BAS</a>
	BAS Background Black, Semi-transparent [channel]
public static final	<a href="#">CONTROLCODES_BBO</a>
	BBO Background Blue, Opaque [channel]

public static final	<a href="#"><u>CONTROLCODES_BBS</u></a> BBS Background Blue, Semi-transparent [channel]
public static final	<a href="#"><u>CONTROLCODES_BCO</u></a> BCO Background Cyan, Opaque [channel]
public static final	<a href="#"><u>CONTROLCODES_BCS</u></a> BCS Background Cyan, Semi-transparent [channel]
public static final	<a href="#"><u>CONTROLCODES_BGO</u></a> BGO Background Green, Opaque [channel]
public static final	<a href="#"><u>CONTROLCODES_BGS</u></a> BGS Background Green, Semi-transparent [channel]
public static final	<a href="#"><u>CONTROLCODES_BMO</u></a> BMO Background Magenta, Opaque [channel]
public static final	<a href="#"><u>CONTROLCODES_BMS</u></a> BMS Background Magenta, Semi-transparent [channel]
public static final	<a href="#"><u>CONTROLCODES_BRO</u></a> BRO Background Red, Opaque [channel]
public static final	<a href="#"><u>CONTROLCODES_BRS</u></a> BRS Background Red, Semi-transparent [channel]
public static final	<a href="#"><u>CONTROLCODES_BS</u></a> BS Backspace [channel]
public static final	<a href="#"><u>CONTROLCODES_BT</u></a> BT Background Transparent [channel]
public static final	<a href="#"><u>CONTROLCODES_BWO</u></a> BWO Background White, Opaque [channel]
public static final	<a href="#"><u>CONTROLCODES_BWS</u></a> BWS Background White, Semi-transparent [channel]
public static final	<a href="#"><u>CONTROLCODES_BYO</u></a> BYO Background Yellow, Opaque [channel]
public static final	<a href="#"><u>CONTROLCODES_BYS</u></a> BYS Background Yellow, Semi-transparent [channel]
public static final	<a href="#"><u>CONTROLCODES_CR</u></a> CR Carriage Return [channel]
public static final	<a href="#"><u>CONTROLCODES_DER</u></a> DER Delete to End of Row [channel]
public static final	<a href="#"><u>CONTROLCODES_EDM</u></a> EDM Erase Displayed Memory [channel]
public static final	<a href="#"><u>CONTROLCODES_ENM</u></a> ENM Erase Non-Displayed Memory [channel]
public static final	<a href="#"><u>CONTROLCODES_EOC</u></a> EOC End of Caption (Flip Memories) [channel]

public static final	<a href="#"><u>CONTROLCODES_FA</u></a> FA Foreground Black [channel]
public static final	<a href="#"><u>CONTROLCODES_FAU</u></a> FAU Foreground Black Underline [channel]
public static final	<a href="#"><u>CONTROLCODES_FON</u></a> FON Flash On [channel]
public static final	<a href="#"><u>CONTROLCODES_RCL</u></a> RCL Resume caption loading [channel]
public static final	<a href="#"><u>CONTROLCODES_RDC</u></a> RDC Resume Direct Captioning [channel]
public static final	<a href="#"><u>CONTROLCODES_RTD</u></a> RTD Resume Text Display [channel]
public static final	<a href="#"><u>CONTROLCODES_RU2</u></a> RU2 Roll-Up Captions-2 Rows [channel]
public static final	<a href="#"><u>CONTROLCODES_RU3</u></a> RU3 Roll-Up Captions-3 Rows [channel]
public static final	<a href="#"><u>CONTROLCODES_RU4</u></a> RU4 Roll-Up Captions-4 Rows [channel]
public static final	<a href="#"><u>CONTROLCODES_TO1</u></a> TO1 Tab Offset 1 Column [channel]
public static final	<a href="#"><u>CONTROLCODES_TO2</u></a> TO2 Tab Offset 2 Columns [channel]
public static final	<a href="#"><u>CONTROLCODES_TO3</u></a> TO3 Tab Offset 3 Columns [channel]
public static final	<a href="#"><u>CONTROLCODES_TR</u></a> TR Text Restart [channel]
public static final	<a href="#"><u>COUNT_COLS</u></a> Total number of columns per frame Value: <b>31</b>
public static final	<a href="#"><u>COUNT_ROWS</u></a> Total number of rows of characters per frame Value: <b>15</b>
public static final	<a href="#"><u>DONOTHING_BYTE</u></a> Value: <b>-128</b>
public static final	<a href="#"><u>DONOTHING_SHORT</u></a> Value: <b>-32640</b>
public static final	<a href="#"><u>FORMAT_COLOR</u></a>
public static final	<a href="#"><u>FORMAT_FLASH</u></a> Value: <b>-27480</b>

public static final	<a href="#"><u>FORMAT_ITALICS</u></a>
	Value: <b>-28242</b>
public static final	<a href="#"><u>FORMAT_ITALICS_UNDERLINE</u></a>
	Value: <b>-28369</b>
public static final	<a href="#"><u>LANG_ASCII</u></a>
	Select the standard line 21 character set in normal size.
public static final	<a href="#"><u>LANG_ASCII_DBL</u></a>
	Select the standard line 21 character set in double size.
public static final	<a href="#"><u>LANG_GB231280</u></a>
	Select the People's Republic of China character set: GB 2312-80.
public static final	<a href="#"><u>LANG_KSC56011987</u></a>
	Select the Korean Standard character set: KSC 5601-1987.
public static final	<a href="#"><u>LANG_PRIV1</u></a>
	Select the first private character set.
public static final	<a href="#"><u>LANG_PRIV2</u></a>
	Select the second private character set.
public static final	<a href="#"><u>LANG_REG1</u></a>
	Select the first registered character set.
public static final	<a href="#"><u>MAX_CCCOUNT</u></a>
	Maximum number of closed caption codes per SEI packet Value: <b>31</b>
public static final	<a href="#"><u>ODDPARITY</u></a>
	Odd parity calculator
public static final	<a href="#"><u>SEI_PAYLOADTYPE</u></a>
	Value: <b>4</b>
public static final	<a href="#"><u>SEI_STARTCODE</u></a>
	SEI start code
public static final	<a href="#"><u>SEI_USERDATATYPECODE</u></a>
	Value: <b>3</b>
public static final	<a href="#"><u>UTF8_BRITISH_POUND</u></a>
	Value: <b>163</b>
public static final	<a href="#"><u>UTF8_BULLET</u></a>
	Value: <b>8226</b>
public static final	<a href="#"><u>UTF8_CENTS</u></a>
	Value: <b>162</b>
public static final	<a href="#"><u>UTF8_CLOSING_DOUBLE_QUOTE</u></a>
	Value: <b>8221</b>

public static final	<a href="#"><u>UTF8_CLOSING_GUILLEMETS</u></a>
	Value: <b>187</b>
public static final	<a href="#"><u>UTF8_CONNECTING_VERTICAL_BAR</u></a>
	Value: <b>9474</b>
public static final	<a href="#"><u>UTF8_COPYRIGHT</u></a>
	Value: <b>169</b>
public static final	<a href="#"><u>UTF8_CURRENCY</u></a>
	Value: <b>164</b>
public static final	<a href="#"><u>UTF8_DASH</u></a>
	Value: <b>45</b>
public static final	<a href="#"><u>UTF8_DEGREE</u></a>
	Value: <b>176</b>
public static final	<a href="#"><u>UTF8_DOUBLEQUOTE</u></a>
	Value: <b>34</b>
public static final	<a href="#"><u>UTF8_EM_DASH</u></a>
	Value: <b>8212</b>
public static final	<a href="#"><u>UTF8_EXCLAMATION</u></a>
	Value: <b>33</b>
public static final	<a href="#"><u>UTF8_GREATERTHAN</u></a>
	Value: <b>62</b>
public static final	<a href="#"><u>UTF8_INVERTED_EXCLAMATION</u></a>
	Value: <b>161</b>
public static final	<a href="#"><u>UTF8_LEFT_SINGLE_QUOTATION</u></a>
	Value: <b>8216</b>
public static final	<a href="#"><u>UTF8_LEFTPAREN</u></a>
	Value: <b>40</b>
public static final	<a href="#"><u>UTF8_LESSTHAN</u></a>
	Value: <b>60</b>
public static final	<a href="#"><u>UTF8_LOWER_A</u></a>
	Value: <b>97</b>
public static final	<a href="#"><u>UTF8_LOWER_A_WITH_CIRCUMFLEX</u></a>
	Value: <b>226</b>

public static final	<a href="#"><u>UTF8_LOWER_A_WITH_DIAERESIS</u></a>
	Value: <b>228</b>
public static final	<a href="#"><u>UTF8_LOWER_A_WITH_GRAVE</u></a>
	Value: <b>224</b>
public static final	<a href="#"><u>UTF8_LOWER_A_WITH_RING</u></a>
	Value: <b>229</b>
public static final	<a href="#"><u>UTF8_LOWER_A_WITH_TILDE</u></a>
	Value: <b>227</b>
public static final	<a href="#"><u>UTF8_LOWER_B</u></a>
	Value: <b>98</b>
public static final	<a href="#"><u>UTF8_LOWER_C</u></a>
	Value: <b>99</b>
public static final	<a href="#"><u>UTF8_LOWER_D</u></a>
	Value: <b>100</b>
public static final	<a href="#"><u>UTF8_LOWER_E</u></a>
	Value: <b>101</b>
public static final	<a href="#"><u>UTF8_LOWER_E_WITH_CIRCUMFLEX</u></a>
	Value: <b>234</b>
public static final	<a href="#"><u>UTF8_LOWER_E_WITH_DIAERESIS</u></a>
	Value: <b>235</b>
public static final	<a href="#"><u>UTF8_LOWER_E_WITH_GRAVE</u></a>
	Value: <b>232</b>
public static final	<a href="#"><u>UTF8_LOWER_F</u></a>
	Value: <b>102</b>
public static final	<a href="#"><u>UTF8_LOWER_G</u></a>
	Value: <b>103</b>
public static final	<a href="#"><u>UTF8_LOWER_H</u></a>
	Value: <b>104</b>
public static final	<a href="#"><u>UTF8_LOWER_I</u></a>
	Value: <b>105</b>
public static final	<a href="#"><u>UTF8_LOWER_I_WITH_CIRCUMFLEX</u></a>
	Value: <b>238</b>

public static final	<a href="#"><u>UTF8_LOWER_I_WITH_DIAERESIS</u></a>
	Value: <b>239</b>
public static final	<a href="#"><u>UTF8_LOWER_I_WITH_GRAVE</u></a>
	Value: <b>236</b>
public static final	<a href="#"><u>UTF8_LOWER_J</u></a>
	Value: <b>106</b>
public static final	<a href="#"><u>UTF8_LOWER_K</u></a>
	Value: <b>107</b>
public static final	<a href="#"><u>UTF8_LOWER_L</u></a>
	Value: <b>108</b>
public static final	<a href="#"><u>UTF8_LOWER_M</u></a>
	Value: <b>109</b>
public static final	<a href="#"><u>UTF8_LOWER_N</u></a>
	Value: <b>110</b>
public static final	<a href="#"><u>UTF8_LOWER_O</u></a>
	Value: <b>111</b>
public static final	<a href="#"><u>UTF8_LOWER_O_WITH_CIRCUMFLEX</u></a>
	Value: <b>244</b>
public static final	<a href="#"><u>UTF8_LOWER_O_WITH_DIAERESIS</u></a>
	Value: <b>246</b>
public static final	<a href="#"><u>UTF8_LOWER_O_WITH_GRAVE</u></a>
	Value: <b>242</b>
public static final	<a href="#"><u>UTF8_LOWER_O_WITH_STROKE</u></a>
	Value: <b>248</b>
public static final	<a href="#"><u>UTF8_LOWER_O_WITH_TILDE</u></a>
	Value: <b>245</b>
public static final	<a href="#"><u>UTF8_LOWER_P</u></a>
	Value: <b>112</b>
public static final	<a href="#"><u>UTF8_LOWER_Q</u></a>
	Value: <b>113</b>
public static final	<a href="#"><u>UTF8_LOWER_R</u></a>
	Value: <b>114</b>

public static final	<a href="#">UTF8_LOWER_S</a>
	Value: <b>115</b>
public static final	<a href="#">UTF8_LOWER_T</a>
	Value: <b>116</b>
public static final	<a href="#">UTF8_LOWER_U</a>
	Value: <b>117</b>
public static final	<a href="#">UTF8_LOWER_U_WITH_CIRCUMFLEX</a>
	Value: <b>251</b>
public static final	<a href="#">UTF8_LOWER_U_WITH_DIAERESIS</a>
	Value: <b>252</b>
public static final	<a href="#">UTF8_LOWER_U_WITH_GRAVE</a>
	Value: <b>249</b>
public static final	<a href="#">UTF8_LOWER_V</a>
	Value: <b>118</b>
public static final	<a href="#">UTF8_LOWER_W</a>
	Value: <b>119</b>
public static final	<a href="#">UTF8_LOWER_X</a>
	Value: <b>120</b>
public static final	<a href="#">UTF8_LOWER_Y</a>
	Value: <b>121</b>
public static final	<a href="#">UTF8_LOWER_Z</a>
	Value: <b>122</b>
public static final	<a href="#">UTF8_MINUSHYPHEN</a>
	Value: <b>45</b>
public static final	<a href="#">UTF8_MUSICAL_NOTE</a>
	Value: <b>9834</b>
public static final	<a href="#">UTF8_NBSP</a>
	Value: <b>160</b>
public static final	<a href="#">UTF8_NOTMATCH</a>
public static final	<a href="#">UTF8_ONE_HALF</a>
	Value: <b>189</b>

public static final	<a href="#"><u>UTF8_OPENING_DOUBLE_QUOTE</u></a>
	Value: <b>8220</b>
public static final	<a href="#"><u>UTF8_OPENING_GUILLEMETS</u></a>
	Value: <b>171</b>
public static final	<a href="#"><u>UTF8_PERIOD</u></a>
	Value: <b>46</b>
public static final	<a href="#"><u>UTF8_PIPE</u></a>
	Value: <b>124</b>
public static final	<a href="#"><u>UTF8_QUESTION</u></a>
	Value: <b>63</b>
public static final	<a href="#"><u>UTF8_REGISTERED_MARK</u></a>
	Value: <b>174</b>
public static final	<a href="#"><u>UTF8_REVERSE_QUERY</u></a>
	Value: <b>191</b>
public static final	<a href="#"><u>UTF8_RIGHTPAREN</u></a>
	Value: <b>41</b>
public static final	<a href="#"><u>UTF8_SERVICE_MARK</u></a>
	Value: <b>8480</b>
public static final	<a href="#"><u>UTF8_SINGLEQUOTE</u></a>
	Value: <b>39</b>
public static final	<a href="#"><u>UTF8_SPACE</u></a>
	Value: <b>32</b>
public static final	<a href="#"><u>UTF8_TRADEMARK</u></a>
	Value: <b>8482</b>
public static final	<a href="#"><u>UTF8_UNDERSCORE</u></a>
	Value: <b>84</b>
public static final	<a href="#"><u>UTF8_UPPER_A</u></a>
	Value: <b>65</b>
public static final	<a href="#"><u>UTF8_UPPER_A_WITH_ACUTE</u></a>
	Value: <b>193</b>
public static final	<a href="#"><u>UTF8_UPPER_A_WITH_CURCUMFLEX</u></a>
	Value: <b>194</b>

public static final	<a href="#"><u>UTF8_UPPER_A_WITH_DIAERESIS</u></a>
	Value: <b>196</b>
public static final	<a href="#"><u>UTF8_UPPER_A_WITH_GRAVE</u></a>
	Value: <b>192</b>
public static final	<a href="#"><u>UTF8_UPPER_A_WITH_RING</u></a>
	Value: <b>197</b>
public static final	<a href="#"><u>UTF8_UPPER_A_WITH_TILDE</u></a>
	Value: <b>195</b>
public static final	<a href="#"><u>UTF8_UPPER_B</u></a>
	Value: <b>66</b>
public static final	<a href="#"><u>UTF8_UPPER_C</u></a>
	Value: <b>67</b>
public static final	<a href="#"><u>UTF8_UPPER_C_WITH_CEDILLA</u></a>
	Value: <b>199</b>
public static final	<a href="#"><u>UTF8_UPPER_D</u></a>
	Value: <b>68</b>
public static final	<a href="#"><u>UTF8_UPPER_E</u></a>
	Value: <b>69</b>
public static final	<a href="#"><u>UTF8_UPPER_E_WITH_ACUTE</u></a>
	Value: <b>201</b>
public static final	<a href="#"><u>UTF8_UPPER_E_WITH_CIRCUMFLEX</u></a>
	Value: <b>202</b>
public static final	<a href="#"><u>UTF8_UPPER_E_WITH_DIAERESIS</u></a>
	Value: <b>203</b>
public static final	<a href="#"><u>UTF8_UPPER_E_WITH_GRAVE</u></a>
	Value: <b>200</b>
public static final	<a href="#"><u>UTF8_UPPER_F</u></a>
	Value: <b>70</b>
public static final	<a href="#"><u>UTF8_UPPER_G</u></a>
	Value: <b>71</b>
public static final	<a href="#"><u>UTF8_UPPER_H</u></a>
	Value: <b>72</b>

public static final	<a href="#"><u>UTF8_UPPER_I</u></a>
	Value: <b>73</b>
public static final	<a href="#"><u>UTF8_UPPER_I_WITH_CIRCUMFLEX</u></a>
	Value: <b>206</b>
public static final	<a href="#"><u>UTF8_UPPER_I_WITH_DIAERESIS</u></a>
	Value: <b>207</b>
public static final	<a href="#"><u>UTF8_UPPER_I_WITH_GRAVE</u></a>
	Value: <b>204</b>
public static final	<a href="#"><u>UTF8_UPPER_I_WITH_TILDE</u></a>
	Value: <b>205</b>
public static final	<a href="#"><u>UTF8_UPPER_J</u></a>
	Value: <b>74</b>
public static final	<a href="#"><u>UTF8_UPPER_K</u></a>
	Value: <b>75</b>
public static final	<a href="#"><u>UTF8_UPPER_L</u></a>
	Value: <b>76</b>
public static final	<a href="#"><u>UTF8_UPPER_M</u></a>
	Value: <b>77</b>
public static final	<a href="#"><u>UTF8_UPPER_N</u></a>
	Value: <b>78</b>
public static final	<a href="#"><u>UTF8_UPPER_O</u></a>
	Value: <b>79</b>
public static final	<a href="#"><u>UTF8_UPPER_O_WITH_ACUTE</u></a>
	Value: <b>211</b>
public static final	<a href="#"><u>UTF8_UPPER_O_WITH_CIRCUMFLEX</u></a>
	Value: <b>212</b>
public static final	<a href="#"><u>UTF8_UPPER_O_WITH_DIAERESIS</u></a>
	Value: <b>214</b>
public static final	<a href="#"><u>UTF8_UPPER_O_WITH_GRAVE</u></a>
	Value: <b>210</b>
public static final	<a href="#"><u>UTF8_UPPER_O_WITH_STROKE</u></a>
	Value: <b>216</b>

public static final	<u>UTF8_UPPER_O_WITH_TILDE</u> Value: <b>213</b>
public static final	<u>UTF8_UPPER_P</u> Value: <b>80</b>
public static final	<u>UTF8_UPPER_Q</u> Value: <b>81</b>
public static final	<u>UTF8_UPPER_R</u> Value: <b>82</b>
public static final	<u>UTF8_UPPER_S</u> Value: <b>83</b>
public static final	<u>UTF8_UPPER_T</u> Value: <b>84</b>
public static final	<u>UTF8_UPPER_U</u> Value: <b>85</b>
public static final	<u>UTF8_UPPER_U_WITH_ACUTE</u> Value: <b>218</b>
public static final	<u>UTF8_UPPER_U_WITH_CIRCUMFLEX</u> Value: <b>219</b>
public static final	<u>UTF8_UPPER_U_WITH_DIAERESIS</u> Value: <b>220</b>
public static final	<u>UTF8_UPPER_U_WITH_GRAVE</u> Value: <b>217</b>
public static final	<u>UTF8_UPPER_V</u> Value: <b>86</b>
public static final	<u>UTF8_UPPER_W</u> Value: <b>87</b>
public static final	<u>UTF8_UPPER_X</u> Value: <b>88</b>
public static final	<u>UTF8_UPPER_Y</u> Value: <b>89</b>
public static final	<u>UTF8_UPPER_Z</u> Value: <b>90</b>

public static final	<a href="#">UTF8_YEN</a>
	Value: <b>165</b>
public static final	<a href="#">UTF8MAP</a>
public static final	<a href="#">UTF8MAP_INDEX_OFFSET</a>
	Value: <b>32</b>

**Fields inherited from class com.wowza.wms.timedtext.cea608.CEA608Constants**

CC\_CHANNEL\_CC1, CC\_CHANNEL\_CC2, CC\_CHANNEL\_CC3, CC\_CHANNEL\_CC4, CC\_TYPE\_DTVCC\_PACKET\_DATA, CC\_TYPE\_DTVCC\_PACKET\_START, CC\_TYPE\_NTSC\_FIELD\_1, CC\_TYPE\_NTSC\_FIELD\_2

**Constructor Summary**

public	<a href="#">ClosedCaptionCEA608Utils()</a>
--------	--

**Method Summary**

static byte	<a href="#">calcParity(byte inByte)</a>
static short	<a href="#">calcParity(short inShort)</a>
static char	<a href="#">cea608CharToCharUTF8(short b)</a>
static Character	<a href="#">cea608ExtendedCharacterToUTF8(short command)</a>
static byte[]	<a href="#">charUTF8ToCAE608(CEACaptionConversionContext ceaContext, String text, int pos, String characterSet, int channel)</a>
static byte[]	<a href="#">charUTF8ToCAE608(String text, int pos, String characterSet, int channel)</a>
static boolean	<a href="#">checkParity(byte inByte)</a>
static boolean	<a href="#">checkParity(short inShort)</a>

**Methods inherited from class java.lang.Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

**Fields****SEI\_PAYLOADTYPE**

public static final byte **SEI\_PAYLOADTYPE**

Constant value: **4**

---

## SEI\_USERDATATYPECODE

```
public static final byte SEI_USERDATATYPECODE
```

Constant value: 3

---

## UTF8\_BULLET

```
public static final char UTF8_BULLET
```

Constant value: 8226

---

## UTF8\_CLOSING\_DOUBLE\_QUOTE

```
public static final char UTF8_CLOSING_DOUBLE_QUOTE
```

Constant value: 8221

---

## UTF8\_CLOSING\_GUILLEMETS

```
public static final char UTF8_CLOSING_GUILLEMETS
```

Constant value: 187

---

## UTF8\_CONNECTING\_VERTICAL\_BAR

```
public static final char UTF8_CONNECTING_VERTICAL_BAR
```

Constant value: 9474

---

## UTF8\_COPYRIGHT

```
public static final char UTF8_COPYRIGHT
```

Constant value: 169

---

## UTF8\_CURRENCY

```
public static final char UTF8_CURRENCY
```

Constant value: 164

---

## UTF8\_DEGREE

```
public static final char UTF8_DEGREE
```

Constant value: 176

---

(continued from last page)

## UTF8\_EM\_DASH

```
public static final char UTF8_EM_DASH
```

Constant value: **8212**

---

## UTF8\_INVERTED\_EXCLAMATION

```
public static final char UTF8_INVERTED_EXCLAMATION
```

Constant value: **161**

---

## UTF8\_LEFT\_SINGLE\_QUOTATION

```
public static final char UTF8_LEFT_SINGLE_QUOTATION
```

Constant value: **8216**

---

## UTF8\_OPENING\_DOUBLE\_QUOTE

```
public static final char UTF8_OPENING_DOUBLE_QUOTE
```

Constant value: **8220**

---

## UTF8\_OPENING\_GUILLEMETS

```
public static final char UTF8_OPENING_GUILLEMETS
```

Constant value: **171**

---

## UTF8\_REGISTERED\_MARK

```
public static final char UTF8_REGISTERED_MARK
```

Constant value: **174**

---

## UTF8\_SERVICE\_MARK

```
public static final char UTF8_SERVICE_MARK
```

Constant value: **8480**

---

## UTF8\_YEN

```
public static final char UTF8_YEN
```

Constant value: **165**

---

## UTF8\_LOWER\_A\_WITH\_DIAERESIS

```
public static final char UTF8_LOWER_A_WITH_DIAERESIS
```

(continued from last page)

Constant value: **228**

---

## UTF8\_LOWER\_A\_WITH\_RING

```
public static final char UTF8_LOWER_A_WITH_RING
```

Constant value: **229**

---

## UTF8\_LOWER\_A\_WITH\_TILDE

```
public static final char UTF8_LOWER_A_WITH_TILDE
```

Constant value: **227**

---

## UTF8\_LOWER\_E\_WITH\_DIAERESIS

```
public static final char UTF8_LOWER_E_WITH_DIAERESIS
```

Constant value: **235**

---

## UTF8\_LOWER\_I\_WITH\_DIAERESIS

```
public static final char UTF8_LOWER_I_WITH_DIAERESIS
```

Constant value: **239**

---

## UTF8\_LOWER\_I\_WITH\_GRAVE

```
public static final char UTF8_LOWER_I_WITH_GRAVE
```

Constant value: **236**

---

## UTF8\_LOWER\_O\_WITH\_DIAERESIS

```
public static final char UTF8_LOWER_O_WITH_DIAERESIS
```

Constant value: **246**

---

## UTF8\_LOWER\_O\_WITH\_GRAVE

```
public static final char UTF8_LOWER_O_WITH_GRAVE
```

Constant value: **242**

---

## UTF8\_LOWER\_O\_WITH\_STROKE

```
public static final char UTF8_LOWER_O_WITH_STROKE
```

Constant value: **248**

## UTF8\_LOWER\_O\_WITH\_TILDE

```
public static final char UTF8_LOWER_O_WITH_TILDE
```

Constant value: **245**

---

## UTF8\_LOWER\_U\_WITH\_DIAERESIS

```
public static final char UTF8_LOWER_U_WITH_DIAERESIS
```

Constant value: **252**

---

## UTF8\_LOWER\_U\_WITH\_GRAVE

```
public static final char UTF8_LOWER_U_WITH_GRAVE
```

Constant value: **249**

---

## UTF8\_UPPER\_A\_WITH\_ACUTE

```
public static final char UTF8_UPPER_A_WITH_ACUTE
```

Constant value: **193**

---

## UTF8\_UPPER\_A\_WITH\_CURCUMFLEX

```
public static final char UTF8_UPPER_A_WITH_CURCUMFLEX
```

Constant value: **194**

---

## UTF8\_UPPER\_A\_WITH\_DIAERESIS

```
public static final char UTF8_UPPER_A_WITH_DIAERESIS
```

Constant value: **196**

---

## UTF8\_UPPER\_A\_WITH\_GRAVE

```
public static final char UTF8_UPPER_A_WITH_GRAVE
```

Constant value: **192**

---

## UTF8\_UPPER\_A\_WITH\_RING

```
public static final char UTF8_UPPER_A_WITH_RING
```

Constant value: **197**

---

(continued from last page)

## **UTF8\_UPPER\_A\_WITH\_TILDE**

```
public static final char UTF8_UPPER_A_WITH_TILDE
```

Constant value: **195**

---

## **UTF8\_UPPER\_C\_WITH\_CEDILLA**

```
public static final char UTF8_UPPER_C_WITH_CEDILLA
```

Constant value: **199**

---

## **UTF8\_UPPER\_E\_WITH\_ACUTE**

```
public static final char UTF8_UPPER_E_WITH_ACUTE
```

Constant value: **201**

---

## **UTF8\_UPPER\_E\_WITH\_CIRCUMFLEX**

```
public static final char UTF8_UPPER_E_WITH_CIRCUMFLEX
```

Constant value: **202**

---

## **UTF8\_UPPER\_E\_WITH\_DIAERESIS**

```
public static final char UTF8_UPPER_E_WITH_DIAERESIS
```

Constant value: **203**

---

## **UTF8\_UPPER\_E\_WITH\_GRAVE**

```
public static final char UTF8_UPPER_E_WITH_GRAVE
```

Constant value: **200**

---

## **UTF8\_UPPER\_I\_WITH\_CIRCUMFLEX**

```
public static final char UTF8_UPPER_I_WITH_CIRCUMFLEX
```

Constant value: **206**

---

## **UTF8\_UPPER\_I\_WITH\_DIAERESIS**

```
public static final char UTF8_UPPER_I_WITH_DIAERESIS
```

Constant value: **207**

---

## **UTF8\_UPPER\_I\_WITH\_GRAVE**

```
public static final char UTF8_UPPER_I_WITH_GRAVE
```

(continued from last page)

Constant value: **204**

---

### **UTF8\_UPPER\_I\_WITH\_TILDE**

```
public static final char UTF8_UPPER_I_WITH_TILDE
```

Constant value: **205**

---

### **UTF8\_UPPER\_O\_WITH\_ACUTE**

```
public static final char UTF8_UPPER_O_WITH_ACUTE
```

Constant value: **211**

---

### **UTF8\_UPPER\_O\_WITH\_CIRCUMFLEX**

```
public static final char UTF8_UPPER_O_WITH_CIRCUMFLEX
```

Constant value: **212**

---

### **UTF8\_UPPER\_O\_WITH\_DIAERESIS**

```
public static final char UTF8_UPPER_O_WITH_DIAERESIS
```

Constant value: **214**

---

### **UTF8\_UPPER\_O\_WITH\_GRAVE**

```
public static final char UTF8_UPPER_O_WITH_GRAVE
```

Constant value: **210**

---

### **UTF8\_UPPER\_O\_WITH\_STROKE**

```
public static final char UTF8_UPPER_O_WITH_STROKE
```

Constant value: **216**

---

### **UTF8\_UPPER\_O\_WITH\_TILDE**

```
public static final char UTF8_UPPER_O_WITH_TILDE
```

Constant value: **213**

---

### **UTF8\_UPPER\_U\_WITH\_ACUTE**

```
public static final char UTF8_UPPER_U_WITH_ACUTE
```

Constant value: **218**

---

---

## UTF8\_UPPER\_U\_WITH\_CIRCUMFLEX

```
public static final char UTF8_UPPER_U_WITH_CIRCUMFLEX
```

Constant value: **219**

---

## UTF8\_UPPER\_U\_WITH\_DIAERESIS

```
public static final char UTF8_UPPER_U_WITH_DIAERESIS
```

Constant value: **220**

---

## UTF8\_UPPER\_U\_WITH\_GRAVE

```
public static final char UTF8_UPPER_U_WITH_GRAVE
```

Constant value: **217**

---

## MAX\_CCCOUNT

```
public static final int MAX_CCCOUNT
```

Maximum number of closed caption codes per SEI packet  
Constant value: **31**

---

## SEI\_STARTCODE

```
public static final byte SEI_STARTCODE
```

SEI start code

---

## ODDPARITY

```
public static final byte ODDPARITY
```

Odd parity calculator

---

## COL0\_WHITE

```
public static final int COL0_WHITE
```

Constant value: **0**

---

## COL0\_GREEN

```
public static final int COL0_GREEN
```

Constant value: **2**

---

## COL0\_BLUE

```
public static final int COL0_BLUE
```

(continued from last page)

Constant value: **4**

---

## **COL0\_CYAN**

```
public static final int COL0_CYAN
```

Constant value: **6**

---

## **COL0\_RED**

```
public static final int COL0_RED
```

Constant value: **8**

---

## **COL0\_YELLOW**

```
public static final int COL0_YELLOW
```

Constant value: **10**

---

## **COL0\_MAGENTA**

```
public static final int COL0_MAGENTA
```

Constant value: **12**

---

## **CHANNEL1**

```
public static final int CHANNEL1
```

Constant value: **0**

---

## **CHANNEL2**

```
public static final int CHANNEL2
```

Constant value: **1**

---

## **DONOTHING\_SHORT**

```
public static final short DONOTHING_SHORT
```

Constant value: **-32640**

---

## **DONOTHING\_BYTE**

```
public static final byte DONOTHING_BYTE
```

Constant value: **-128**

---

(continued from last page)

## **CONTROLCODES\_RCL**

public static final short **CONTROLCODES\_RCL**

RCL Resume caption loading [channel]

---

## **CONTROLCODES\_BS**

public static final short **CONTROLCODES\_BS**

BS Backspace [channel]

---

## **CONTROLCODES\_AOF**

public static final short **CONTROLCODES\_AOF**

AOF Reserved (formerly Alarm Off) [channel]

---

## **CONTROLCODES\_AON**

public static final short **CONTROLCODES\_AON**

AON Reserved (formerly Alarm On) [channel]

---

## **CONTROLCODES\_DER**

public static final short **CONTROLCODES\_DER**

DER Delete to End of Row [channel]

---

## **CONTROLCODES\_RU2**

public static final short **CONTROLCODES\_RU2**

RU2 Roll-Up Captions-2 Rows [channel]

---

## **CONTROLCODES\_RU3**

public static final short **CONTROLCODES\_RU3**

RU3 Roll-Up Captions-3 Rows [channel]

---

## **CONTROLCODES\_RU4**

public static final short **CONTROLCODES\_RU4**

RU4 Roll-Up Captions-4 Rows [channel]

---

## **CONTROLCODES\_FON**

public static final short **CONTROLCODES\_FON**

FON Flash On [channel]

---

## **CONTROLCODES\_RDC**

public static final short **CONTROLCODES\_RDC**

(continued from last page)

RDC Resume Direct Captioning [channel]

---

## **CONTROLCODES\_TR**

public static final short **CONTROLCODES\_TR**

TR Text Restart [channel]

---

## **CONTROLCODES\_RTD**

public static final short **CONTROLCODES\_RTD**

RTD Resume Text Display [channel]

---

## **CONTROLCODES\_EDM**

public static final short **CONTROLCODES\_EDM**

EDM Erase Displayed Memory [channel]

---

## **CONTROLCODES\_CR**

public static final short **CONTROLCODES\_CR**

CR Carriage Return [channel]

---

## **CONTROLCODES\_ENM**

public static final short **CONTROLCODES\_ENM**

ENM Erase Non-Displayed Memory [channel]

---

## **CONTROLCODES\_EOC**

public static final short **CONTROLCODES\_EOC**

EOC End of Caption (Flip Memories) [channel]

---

## **CONTROLCODES\_TO1**

public static final short **CONTROLCODES\_TO1**

TO1 Tab Offset 1 Column [channel]

---

## **CONTROLCODES\_TO2**

public static final short **CONTROLCODES\_TO2**

TO2 Tab Offset 2 Columns [channel]

---

## **CONTROLCODES\_TO3**

public static final short **CONTROLCODES\_TO3**

TO3 Tab Offset 3 Columns [channel]

---

(continued from last page)

## **CONTROLCODES\_BWO**

public static final short **CONTROLCODES\_BWO**

BWO Background White, Opaque [channel]

---

## **CONTROLCODES\_BWS**

public static final short **CONTROLCODES\_BWS**

BWS Background White, Semi-transparent [channel]

---

## **CONTROLCODES\_BGO**

public static final short **CONTROLCODES\_BGO**

BGO Background Green, Opaque [channel]

---

## **CONTROLCODES\_BGS**

public static final short **CONTROLCODES\_BGS**

BGS Background Green, Semi-transparent [channel]

---

## **CONTROLCODES\_BBO**

public static final short **CONTROLCODES\_BBO**

BBO Background Blue, Opaque [channel]

---

## **CONTROLCODES\_BBS**

public static final short **CONTROLCODES\_BBS**

BBS Background Blue, Semi-transparent [channel]

---

## **CONTROLCODES\_BCO**

public static final short **CONTROLCODES\_BCO**

BCO Background Cyan, Opaque [channel]

---

## **CONTROLCODES\_BCS**

public static final short **CONTROLCODES\_BCS**

BCS Background Cyan, Semi-transparent [channel]

---

## **CONTROLCODES\_BRO**

public static final short **CONTROLCODES\_BRO**

BRO Background Red, Opaque [channel]

---

## **CONTROLCODES\_BRS**

public static final short **CONTROLCODES\_BRS**

(continued from last page)

BRS Background Red, Semi-transparent [channel]

---

## **CONTROLCODES\_BYO**

public static final short **CONTROLCODES\_BYO**

BYO Background Yellow, Opaque [channel]

---

## **CONTROLCODES\_BYS**

public static final short **CONTROLCODES\_BYS**

BYS Background Yellow, Semi-transparent [channel]

---

## **CONTROLCODES\_BMO**

public static final short **CONTROLCODES\_BMO**

BMO Background Magenta, Opaque [channel]

---

## **CONTROLCODES\_BMS**

public static final short **CONTROLCODES\_BMS**

BMS Background Magenta, Semi-transparent [channel]

---

## **CONTROLCODES\_BAO**

public static final short **CONTROLCODES\_BAO**

BAO Background Black, Opaque [channel]

---

## **CONTROLCODES\_BAS**

public static final short **CONTROLCODES\_BAS**

BAS Background Black, Semi-transparent [channel]

---

## **CONTROLCODES\_BT**

public static final short **CONTROLCODES\_BT**

BT Background Transparent [channel]

---

## **CONTROLCODES\_FA**

public static final short **CONTROLCODES\_FA**

FA Foreground Black [channel]

---

## **CONTROLCODES\_FAU**

public static final short **CONTROLCODES\_FAU**

FAU Foreground Black Underline [channel]

---

---

(continued from last page)

## **LANG\_ASCII**

```
public static final short LANG_ASCII
```

Select the standard line 21 character set in normal size. [channel]

---

## **LANG\_ASCII\_DBL**

```
public static final short LANG_ASCII_DBL
```

Select the standard line 21 character set in double size. [channel]

---

## **LANG\_PRIV1**

```
public static final short LANG_PRIV1
```

Select the first private character set. [channel]

---

## **LANG\_PRIV2**

```
public static final short LANG_PRIV2
```

Select the second private character set. [channel]

---

## **LANG\_GB231280**

```
public static final short LANG_GB231280
```

Select the People's Republic of China character set: GB 2312-80. [channel]

---

## **LANG\_KSC56011987**

```
public static final short LANG_KSC56011987
```

Select the Korean Standard character set: KSC 5601-1987. [channel]

---

## **LANG\_REG1**

```
public static final short LANG_REG1
```

Select the first registered character set. [channel]

---

## **COUNT\_ROWS**

```
public static final int COUNT_ROWS
```

Total number of rows of characters per frame  
Constant value: **15**

---

## **COUNT\_COLS**

```
public static final int COUNT_COLS
```

Total number of columns per frame  
Constant value: **31**

---

(continued from last page)

## **COL\_HIBYTE**

```
public static final short COL_HIBYTE
```

Move to set color (high-byte) [channel][row]

---

## **COL\_LOWBYTE**

```
public static final short COL_LOWBYTE
```

Move to set color (low-byte) [color][row]

---

## **COL\_0**

```
public static final int COL_0
```

Constant value: **0**

---

## **COL\_4**

```
public static final int COL_4
```

Constant value: **2**

---

## **COL\_5**

```
public static final int COL_5
```

Constant value: **4**

---

## **COL\_12**

```
public static final int COL_12
```

Constant value: **6**

---

## **COL\_16**

```
public static final int COL_16
```

Constant value: **8**

---

## **COL\_20**

```
public static final int COL_20
```

Constant value: **10**

---

## **COL\_24**

```
public static final int COL_24
```

(continued from last page)

Constant value: **12**

---

**COL\_28**

```
public static final int COL_28
```

Constant value: **14**

---

**COLN\_LOWBYTE**

```
public static final short COLN_LOWBYTE
```

---

**FORMAT\_COLOR**

```
public static final short FORMAT_COLOR
```

---

**FORMAT\_ITALICS**

```
public static final short FORMAT_ITALICS
```

Constant value: **-28242**

---

**FORMAT\_ITALICS\_UNDERLINE**

```
public static final short FORMAT_ITALICS_UNDERLINE
```

Constant value: **-28369**

---

**FORMAT\_FLASH**

```
public static final short FORMAT_FLASH
```

Constant value: **-27480**

---

**UTF8\_NOTMATCH**

```
public static final byte UTF8_NOTMATCH
```

---

**UTF8\_PERIOD**

```
public static final byte UTF8_PERIOD
```

Constant value: **46**

---

**UTF8\_SINGLEQUOTE**

```
public static final byte UTF8_SINGLEQUOTE
```

(continued from last page)

Constant value: **39**

---

## **UTF8\_DOUBLEQUOTE**

```
public static final byte UTF8_DOUBLEQUOTE
```

Constant value: **34**

---

## **UTF8\_DASH**

```
public static final byte UTF8_DASH
```

Constant value: **45**

---

## **UTF8\_UNDERSCORE**

```
public static final byte UTF8_UNDERSCORE
```

Constant value: **84**

---

## **UTF8\_LEFTPAREN**

```
public static final byte UTF8_LEFTPAREN
```

Constant value: **40**

---

## **UTF8\_RIGHTPAREN**

```
public static final byte UTF8_RIGHTPAREN
```

Constant value: **41**

---

## **UTF8\_SPACE**

```
public static final byte UTF8_SPACE
```

Constant value: **32**

---

## **UTF8\_EXCLAMATION**

```
public static final byte UTF8_EXCLAMATION
```

Constant value: **33**

---

## **UTF8\_QUESTION**

```
public static final byte UTF8_QUESTION
```

Constant value: **63**

---

---

(continued from last page)

## **UTF8\_LESS\_THAN**

```
public static final byte UTF8_LESS_THAN
```

Constant value: **60**

---

## **UTF8\_GREATER\_THAN**

```
public static final byte UTF8_GREATER_THAN
```

Constant value: **62**

---

## **UTF8\_MINUS\_HYPHEN**

```
public static final byte UTF8_MINUS_HYPHEN
```

Constant value: **45**

---

## **UTF8\_PIPE**

```
public static final byte UTF8_PIPE
```

Constant value: **124**

---

## **UTF8\_LOWER\_A**

```
public static final byte UTF8_LOWER_A
```

Constant value: **97**

---

## **UTF8\_LOWER\_B**

```
public static final byte UTF8_LOWER_B
```

Constant value: **98**

---

## **UTF8\_LOWER\_C**

```
public static final byte UTF8_LOWER_C
```

Constant value: **99**

---

## **UTF8\_LOWER\_D**

```
public static final byte UTF8_LOWER_D
```

Constant value: **100**

---

## **UTF8\_LOWER\_E**

```
public static final byte UTF8_LOWER_E
```

(continued from last page)

Constant value: **101**

---

## **UTF8\_LOWER\_F**

public static final byte **UTF8\_LOWER\_F**

Constant value: **102**

---

## **UTF8\_LOWER\_G**

public static final byte **UTF8\_LOWER\_G**

Constant value: **103**

---

## **UTF8\_LOWER\_H**

public static final byte **UTF8\_LOWER\_H**

Constant value: **104**

---

## **UTF8\_LOWER\_I**

public static final byte **UTF8\_LOWER\_I**

Constant value: **105**

---

## **UTF8\_LOWER\_J**

public static final byte **UTF8\_LOWER\_J**

Constant value: **106**

---

## **UTF8\_LOWER\_K**

public static final byte **UTF8\_LOWER\_K**

Constant value: **107**

---

## **UTF8\_LOWER\_L**

public static final byte **UTF8\_LOWER\_L**

Constant value: **108**

---

## **UTF8\_LOWER\_M**

public static final byte **UTF8\_LOWER\_M**

Constant value: **109**

---

## UTF8\_LOWER\_N

```
public static final byte UTF8_LOWER_N
```

Constant value: **110**

---

## UTF8\_LOWER\_O

```
public static final byte UTF8_LOWER_O
```

Constant value: **111**

---

## UTF8\_LOWER\_P

```
public static final byte UTF8_LOWER_P
```

Constant value: **112**

---

## UTF8\_LOWER\_Q

```
public static final byte UTF8_LOWER_Q
```

Constant value: **113**

---

## UTF8\_LOWER\_R

```
public static final byte UTF8_LOWER_R
```

Constant value: **114**

---

## UTF8\_LOWER\_S

```
public static final byte UTF8_LOWER_S
```

Constant value: **115**

---

## UTF8\_LOWER\_T

```
public static final byte UTF8_LOWER_T
```

Constant value: **116**

---

## UTF8\_LOWER\_U

```
public static final byte UTF8_LOWER_U
```

Constant value: **117**

---

(continued from last page)

## UTF8\_LOWER\_V

```
public static final byte UTF8_LOWER_V
```

Constant value: **118**

---

## UTF8\_LOWER\_W

```
public static final byte UTF8_LOWER_W
```

Constant value: **119**

---

## UTF8\_LOWER\_X

```
public static final byte UTF8_LOWER_X
```

Constant value: **120**

---

## UTF8\_LOWER\_Y

```
public static final byte UTF8_LOWER_Y
```

Constant value: **121**

---

## UTF8\_LOWER\_Z

```
public static final byte UTF8_LOWER_Z
```

Constant value: **122**

---

## UTF8\_UPPER\_A

```
public static final byte UTF8_UPPER_A
```

Constant value: **65**

---

## UTF8\_UPPER\_B

```
public static final byte UTF8_UPPER_B
```

Constant value: **66**

---

## UTF8\_UPPER\_C

```
public static final byte UTF8_UPPER_C
```

Constant value: **67**

---

## UTF8\_UPPER\_D

```
public static final byte UTF8_UPPER_D
```

(continued from last page)

Constant value: **68**

---

## **UTF8\_UPPER\_E**

public static final byte **UTF8\_UPPER\_E**

Constant value: **69**

---

## **UTF8\_UPPER\_F**

public static final byte **UTF8\_UPPER\_F**

Constant value: **70**

---

## **UTF8\_UPPER\_G**

public static final byte **UTF8\_UPPER\_G**

Constant value: **71**

---

## **UTF8\_UPPER\_H**

public static final byte **UTF8\_UPPER\_H**

Constant value: **72**

---

## **UTF8\_UPPER\_I**

public static final byte **UTF8\_UPPER\_I**

Constant value: **73**

---

## **UTF8\_UPPER\_J**

public static final byte **UTF8\_UPPER\_J**

Constant value: **74**

---

## **UTF8\_UPPER\_K**

public static final byte **UTF8\_UPPER\_K**

Constant value: **75**

---

## **UTF8\_UPPER\_L**

public static final byte **UTF8\_UPPER\_L**

Constant value: **76**

---

## **UTF8\_UPPER\_M**

```
public static final byte UTF8_UPPER_M
```

Constant value: **77**

---

## **UTF8\_UPPER\_N**

```
public static final byte UTF8_UPPER_N
```

Constant value: **78**

---

## **UTF8\_UPPER\_O**

```
public static final byte UTF8_UPPER_O
```

Constant value: **79**

---

## **UTF8\_UPPER\_P**

```
public static final byte UTF8_UPPER_P
```

Constant value: **80**

---

## **UTF8\_UPPER\_Q**

```
public static final byte UTF8_UPPER_Q
```

Constant value: **81**

---

## **UTF8\_UPPER\_R**

```
public static final byte UTF8_UPPER_R
```

Constant value: **82**

---

## **UTF8\_UPPER\_S**

```
public static final byte UTF8_UPPER_S
```

Constant value: **83**

---

## **UTF8\_UPPER\_T**

```
public static final byte UTF8_UPPER_T
```

Constant value: **84**

---

---

(continued from last page)

## **UTF8\_UPPER\_U**

```
public static final byte UTF8_UPPER_U
```

Constant value: **85**

---

## **UTF8\_UPPER\_V**

```
public static final byte UTF8_UPPER_V
```

Constant value: **86**

---

## **UTF8\_UPPER\_W**

```
public static final byte UTF8_UPPER_W
```

Constant value: **87**

---

## **UTF8\_UPPER\_X**

```
public static final byte UTF8_UPPER_X
```

Constant value: **88**

---

## **UTF8\_UPPER\_Y**

```
public static final byte UTF8_UPPER_Y
```

Constant value: **89**

---

## **UTF8\_UPPER\_Z**

```
public static final byte UTF8_UPPER_Z
```

Constant value: **90**

---

## **UTF8\_LOWER\_A\_WITH\_CIRCUMFLEX**

```
public static final char UTF8_LOWER_A_WITH_CIRCUMFLEX
```

Constant value: **226**

---

## **UTF8\_LOWER\_E\_WITH\_CIRCUMFLEX**

```
public static final char UTF8_LOWER_E_WITH_CIRCUMFLEX
```

Constant value: **234**

---

## **UTF8\_LOWER\_I\_WITH\_CIRCUMFLEX**

```
public static final char UTF8_LOWER_I_WITH_CIRCUMFLEX
```

(continued from last page)

Constant value: **238**

---

## **UTF8\_LOWER\_O\_WITH\_CIRCUMFLEX**

```
public static final char UTF8_LOWER_O_WITH_CIRCUMFLEX
```

Constant value: **244**

---

## **UTF8\_LOWER\_U\_WITH\_CIRCUMFLEX**

```
public static final char UTF8_LOWER_U_WITH_CIRCUMFLEX
```

Constant value: **251**

---

## **UTF8\_LOWER\_A\_WITH\_GRAVE**

```
public static final char UTF8_LOWER_A_WITH_GRAVE
```

Constant value: **224**

---

## **UTF8\_LOWER\_E\_WITH\_GRAVE**

```
public static final char UTF8_LOWER_E_WITH_GRAVE
```

Constant value: **232**

---

## **UTF8\_NBSP**

```
public static final char UTF8_NBSP
```

Constant value: **160**

---

## **UTF8\_MUSICAL\_NOTE**

```
public static final char UTF8_MUSICAL_NOTE
```

Constant value: **9834**

---

## **UTF8\_BRITISH\_POUND**

```
public static final char UTF8_BRITISH_POUND
```

Constant value: **163**

---

## **UTF8\_CENTS**

```
public static final char UTF8_CENTS
```

Constant value: **162**

---

## UTF8\_TRADEMARK

```
public static final char UTF8_TRADEMARK
```

Constant value: **8482**

---

## UTF8\_REVERSE\_QUERY

```
public static final char UTF8_REVERSE_QUERY
```

Constant value: **191**

---

## UTF8\_ONE\_HALF

```
public static final char UTF8_ONE_HALF
```

Constant value: **189**

---

## CEA608\_MUSICAL\_NOTE

```
public static byte CEA608_MUSICAL_NOTE
```

---

## CEA608\_SOLID\_BLOCK

```
public static byte CEA608_SOLID_BLOCK
```

---

## UTF8MAP\_INDEX\_OFFSET

```
public static final int UTF8MAP_INDEX_OFFSET
```

Constant value: **32**

---

## UTF8MAP

```
public static final byte UTF8MAP
```

---

## CEA608\_EXTENDED\_NA\_MAP

```
public static final char CEA608_EXTENDED_NA_MAP
```

---

## CEA608\_EXTENDED\_FR\_SP\_MAP

```
public static final char CEA608_EXTENDED_FR_SP_MAP
```

## CEA608\_EXTENDED\_PORT\_GERMAN\_DANISH\_MAP

```
public static final char CEA608_EXTENDED_PORT_GERMAN_DANISH_MAP
```

### Constructors

#### ClosedCaptionCEA608Utils

```
public ClosedCaptionCEA608Utils()
```

### Methods

#### charUTF8ToCAE608

```
public static byte[] charUTF8ToCAE608(CEACaptionConversionContext ceaContext,  
                                     String text,  
                                     int pos,  
                                     String characterSet,  
                                     int channel)
```

---

#### charUTF8ToCAE608

```
public static byte[] charUTF8ToCAE608(String text,  
                                      int pos,  
                                      String characterSet,  
                                      int channel)
```

---

#### cea608CharToCharUTF8

```
public static char cea608CharToCharUTF8(short b)
```

---

#### calcParity

```
public static byte calcParity(byte inByte)
```

---

#### calcParity

```
public static short calcParity(short inShort)
```

---

#### checkParity

```
public static boolean checkParity(byte inByte)
```

---

## checkParity

```
public static boolean checkParity(short inShort)
```

---

## cea608ExtendedCharacterToUTF8

```
public static Character cea608ExtendedCharacterToUTF8(short command)
```

---

Package

**com.wowza.wms.timedtext.model**

## com.wowza.wms.timedtext.model Interface ITimedTextConstants

public interface **ITimedTextConstants**  
extends ITimedTextPrivateConstants

### Field Summary

public static final	<a href="#">AMF_TEXTTYPE_ON_TEXT_DATA</a> Value: <b>onTextData</b>
public static final	<a href="#">DEFAULT_AMF_CONVERTER_TEXT_TYPE</a> Value: <b>onTextData</b>
public static final	<a href="#">DEFAULT_AMF_CONVERTER_TRACK_INDEX</a> Value: <b>99</b>
public static final	<a href="#">DEFAULT_CAPTION_CHARACTER_SET</a> Default value for Timed Text Property "captionCharacterSet". Value: <b>UTF-8</b>
public static final	<a href="#">DEFAULT_MAXIMUM_CAPTION_DURATION</a> Default value for Timed Text Property "maximumCaptionDuration". Value: <b>5000</b>
public static final	<a href="#">DEFAULT_PROPERTY_CAPION_LANGUAGES</a> Default value for Timed Text Property "captionLanguages". Value: <b>*</b>
public static final	<a href="#">DEFAULT_PROPERTY_CAPTION_FILE_NAMING_RULE</a> Default value for Timed Text Property "captionFileNamingRule". Value: <b> \${SourceStreamName}.\${FileExtension}</b>
public static final	<a href="#">DEFAULT_PROPERTY_CAPTION_PATH_NAMING_RULE</a> Default value for Timed Text Property "captionFileNamingRule". Value: <b> \${SourceBasePath}</b>
public static final	<a href="#">LANGUAGE_ID_ENGLISH</a> Value: <b>eng</b>
public static final	<a href="#">LANGUAGE_LOCALE</a> Value: <b>locale</b>
public static final	<a href="#">ON_CAPTION</a> Value: <b>onCaption</b>
public static final	<a href="#">ON_CAPTION_INFO</a> Value: <b>onCaptionInfo</b>

public static final	<a href="#"><u>ON_TEXT_DATA</u></a> Value: <b>onTextData</b>
public static final	<a href="#"><u>OUTPUT_TYPE_AMF</u></a> Value: <b>AMF</b>
public static final	<a href="#"><u>OUTPUT_TYPE_CEA608</u></a> Value: <b>CEA608</b>
public static final	<a href="#"><u>OUTPUT_TYPE_TTML</u></a> Value: <b>TTML</b>
public static final	<a href="#"><u>PROP_3GPP_CAPTION_CHARACTER_SET</u></a> Timed Text Property "reader3GPPCaptionCharacterSet": used to determine character set when ingesting TTML captions Overrides general property "captionCharacterSet" Value: <b>reader3GPPCaptionCharacterSet</b>
public static final	<a href="#"><u>PROP_AMF_CONVERTER_INSERT_ERASES_IN_GAPS</u></a> Timed Text Property "amfCaptionConverterInsertErases": insert erase captions if gaps between captions are greater than PROP_MAXIMUM_CAPTION_DURATION Value: <b>amfCaptionConverterInsertErases</b>
public static final	<a href="#"><u>PROP_AMF_CONVERTER_TEXT_TYPE</u></a> Timed Text Property "amfCaptionConverterTextType": AMF text type used in conversion Value: <b>amfCaptionConverterTextType</b>
public static final	<a href="#"><u>PROP_AMF_CONVERTER_TRACK_INDEX</u></a> Timed Text Property "amfCaptionConverterTrackIndex": track index used for captions Value: <b>amfCaptionConverterTrackIndex</b>
public static final	<a href="#"><u>PROP_CAPTION_CHARACTER_SET</u></a> Timed Text Property "captionCharacterSet": used to determine character set when ingesting captions Value: <b>captionCharacterSet</b>
public static final	<a href="#"><u>PROP_CAPTION_FILE_NAMING_RULE</u></a> Timed Text Property "captionFileNamingRule": to change logic determining the caption companion file naming scheme Value: <b>captionFileNamingRule</b>
public static final	<a href="#"><u>PROP_CAPTION_FILENAME_QUERY_PARAMETER</u></a> HTTP Streamer Property "captionFilenameQueryParameter": used for changing query string param used for caption file selection. Value: <b>captionFilenameQueryParameter</b>
public static final	<a href="#"><u>PROP_CAPTION_LANGUAGE_QUERY_PARAMETER</u></a> HTTP Streamer Property "captionLanguageQueryParameter": used for changing query string param used for caption language selection. Value: <b>captionLanguageQueryParameter</b>
public static final	<a href="#"><u>PROP_CAPTION_LANGUAGES</u></a> Timed Text Property "captionLanguages": used to select presented caption Language. Value: <b>captionLanguages</b>

public static final	<a href="#"><u>PROP_CAPTION_PATH_NAMING_RULE</u></a> Timed Text Property "captionPathNamingRule": to change logic determining the caption companion path Value: <b>captionPathNamingRule</b>
public static final	<a href="#"><u>PROP_CEA608_CONVERTER_CHARACTER_SET</u></a> Timed Text Property "cea608CaptionConverterCharacterSet": character set in default CEA caption converter. Value: <b>cea608CaptionConverterCharacterSet</b>
public static final	<a href="#"><u>PROP_CEA608_CONVERTER_CHARS_PER_LINE</u></a> Timed Text Property "cea608CaptionConverterCharsPerLine": max number characters per line in default CEA caption converter. Value: <b>cea608CaptionConverterCharsPerLine</b>
public static final	<a href="#"><u>PROP_CEA608_CONVERTER_COLOR</u></a> Timed Text Property "cea608CaptionConverterColor": color of text in default CEA caption converter. Value: <b>cea608CaptionConverterColor</b>
public static final	<a href="#"><u>PROP_CUPERTINO_CAPTION_LANGUAGES</u></a> Timed Text Property "cupertinoCaptionLanguages": to enabled captionLanguage selection for Cupertino streaming. Value: <b>cupertinoCaptionLanguages</b>
public static final	<a href="#"><u>PROP_CUPERTINO_LIVE_USE_WEBVTT</u></a> Timed Text Property "cupertinoLiveCaptionsUseWebVTT" Value: <b>cupertinoLiveCaptionsUseWebVTT</b>
public static final	<a href="#"><u>PROP_CUPERTINO_VOD_CAPTIONS_ENABLED</u></a> Timed Text Property "cupertinoVODCaptionsEnabled": to enabled VOD captions for Cupertino streaming. Value: <b>cupertinoVODCaptionsEnabled</b>
public static final	<a href="#"><u>PROP_CUPERTINO_VOD_REMOVE_EXISTING_CEA608</u></a> Timed Text Property "cupertinoVODCaptionsRemoveExistingCEA608": if existing VOD has 608, remove them Value: <b>cupertinoVODCaptionsRemoveExistingCEA608</b>
public static final	<a href="#"><u>PROP_CUPERTINO_VOD_USE_WEBVTT</u></a> Timed Text Property "cupertinoVODCaptionsUseWebVTT" Value: <b>cupertinoVODCaptionsUseWebVTT</b>
public static final	<a href="#"><u>PROP_DEBUG_3GPP_CAPTION_PARSER</u></a> Timed Text Property "debug3GPPCaptionParser": adds info level logging regarding the 3GPP parser Value: <b>debug3GPPCaptionParser</b>
public static final	<a href="#"><u>PROP_DEBUG_3GPP_CAPTION_PARSER_TIME</u></a> Timed Text Property "debug3GPPCaptionParserTime": adds info level logging regarding how long it took parser to index Value: <b>debug3GPPCaptionParserTime</b>
public static final	<a href="#"><u>PROP_DEBUG_CUPERTINO_VOD_CAPTION_LANGUAGE_SELECTION</u></a> Timed Text Property "cupertinoDebugVODCaptionLanguageSelection": adds info level logging regarding language selection for Cupertino Streaming Value: <b>cupertinoDebugVODCaptionLanguageSelection</b>

public static final	<u><a href="#">PROP_DEBUG_CUPERTINO_VOD_CAPTION_PROVIDER_DETERMINATION</a></u> Timed Text Property "cupertinoDebugVODCaptionProviderDetermination": adds info level logging regarding caption provider determination for a given VOD asset for Cupertino Streaming. Value: <b>cupertinoDebugVODCaptionProviderDetermination</b>
public static final	<u><a href="#">PROP_DEBUG_LIVE_CAPTION_INGEST</a></u> Timed Text Property "debugCaptionLiveIngest" Value: <b>debugCaptionLiveIngest</b>
public static final	<u><a href="#">PROP_DEBUG_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608_LOG_AS_RAW</a></u> Timed Text Property "debugCaptionLiveIngestOnCaptionCEA608LogAsRaw" : Adds additional logging of decoding AMF onCaption events of type "708" (CEA-608) such that they are logged in a raw binary state Value: <b>debugCaptionLiveIngestOnCaptionCEA608LogAsRaw</b>
public static final	<u><a href="#">PROP_DEBUG_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608_LOG_AS_SCC</a></u> Timed Text Property "debugCaptionLiveIngestOnCaptionCEA608LogAssCC" : Adds additional logging of decoding AMF onCaption events of type "708" (CEA-608) such that they are logged in SCC format Value: <b>debugCaptionLiveIngestOnCaptionCEA608LogAssCC</b>
public static final	<u><a href="#">PROP_DEBUG_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608_VERBOSE</a></u> Timed Text Property "debugCaptionLiveIngestOnCaptionCEA608Verbose" : Adds verbose logging to AMF onCaption events of type "708" (CEA-608) Value: <b>debugCaptionLiveIngestOnCaptionCEA608Verbose</b>
public static final	<u><a href="#">PROP_DEBUG_LIVE_CAPTION_ONCAPTION_INGEST</a></u> Timed Text Property "debugCaptionLiveOnCaptionIngest" Value: <b>debugCaptionLiveOnCaptionIngest</b>
public static final	<u><a href="#">PROP_DEBUG_LIVE_CAPTION_ONTEXTDATA_INGEST</a></u> Timed Text Property "debugCaptionLiveOnTextDataIngest" Value: <b>debugCaptionLiveOnTextDataIngest</b>
public static final	<u><a href="#">PROP_DEBUG_RTMP_VOD_CAPTION_LANGUAGE_SELECTION</a></u> Timed Text Property "rtmpDebugVODCaptionLanguageSelection": adds info level logging regarding language selection for RTMP Streaming Value: <b>rtmpDebugVODCaptionLanguageSelection</b>
public static final	<u><a href="#">PROP_DEBUG_RTMP_VOD_CAPTION_PROVIDER_DETERMINATION</a></u> Timed Text Property "rtmpDebugVODCaptionProviderDetermination": adds info level logging regarding caption provider determination for a given VOD asset for RTMP Streaming. Value: <b>rtmpDebugVODCaptionProviderDetermination</b>
public static final	<u><a href="#">PROP_DEBUG_SANJOSE_VOD_CAPTION_LANGUAGE_SELECTION</a></u> Timed Text Property "sanJoseDebugVODCaptionLanguageSelection": adds info level logging regarding language selection for San Jose Streaming Value: <b>sanJoseDebugVODCaptionLanguageSelection</b>
public static final	<u><a href="#">PROP_DEBUG_SANJOSE_VOD_CAPTION_PROVIDER_DETERMINATION</a></u> Timed Text Property "sanJoseVODDebugCaptionProviderDetermination": adds info level logging regarding caption provider determination for a given VOD asset for San Jose Streaming. Value: <b>sanJoseVODDebugCaptionProviderDetermination</b>

public static final	<a href="#"><u>PROP_DEBUG_SCC_CAPTION_PARSER</u></a> Timed Text Property "debugSCCCaptionParser": adds info level logging regarding the SCC caption parser Value: <b>debugSCCCaptionParser</b>
public static final	<a href="#"><u>PROP_DEBUG_SCC_CAPTION_PARSER_TIME</u></a> Timed Text Property "debugSCCCaptionParserTime": adds info level logging regarding how long it took parser to index Value: <b>debugSCCCaptionParserTime</b>
public static final	<a href="#"><u>PROP_DEBUG_SCC_VOD_CAPTION_COMPANION_FILES</u></a> Timed Text Property "debugSCCVODCaptionFileDetermination": adds info level logging regarding caption companion file determination for a given VOD asset for SCC provider Value: <b>debugSCCVODCaptionFileDetermination</b>
public static final	<a href="#"><u>PROP_DEBUG_SRT_CAPTION_PARSER</u></a> Timed Text Property "debugSRTCaptionParser": adds info level logging regarding the SRT caption parser Value: <b>debugSRTCaptionParser</b>
public static final	<a href="#"><u>PROP_DEBUG_SRT_CAPTION_PARSER_TIME</u></a> Timed Text Property "debugSRTCaptionParserTime": adds info level logging regarding how long it took parser to index Value: <b>debugSRTCaptionParserTime</b>
public static final	<a href="#"><u>PROP_DEBUG_SRT_VOD_CAPTION_COMPANION_FILES</u></a> Timed Text Property "debugSRTVODCaptionFileDetermination": adds info level logging regarding caption companion file determination for a given VOD asset for SRT provider Value: <b>debugSRTVODCaptionFileDetermination</b>
public static final	<a href="#"><u>PROP_DEBUG_TTML_CAPTION_PARSER</u></a> Timed Text Property "debugTTMLCaptionParser": adds info level logging regarding the TTML caption parser Value: <b>debugTTMLCaptionParser</b>
public static final	<a href="#"><u>PROP_DEBUG_TTML_CAPTION_PARSER_TIME</u></a> Timed Text Property "debugTTMLCaptionParserTime": adds info level logging regarding how long it took parser to index Value: <b>debugTTMLCaptionParserTime</b>
public static final	<a href="#"><u>PROP_DEBUG_TTML_VOD_CAPTION_COMPANION_FILES</u></a> Timed Text Property "debugTTMLVODCaptionFileDetermination": adds info level logging regarding caption companion file determination for a given VOD asset for TTML Streaming. Value: <b>debugTTMLVODCaptionFileDetermination</b>
public static final	<a href="#"><u>PROP_DEBUG_VOD_CAPTION_COMPANION_FILES</u></a> Timed Text Property "debugVODCaptionFileDetermination": adds info level logging regarding caption companion file determination for a given VOD asset Value: <b>debugVODCaptionFileDetermination</b>
public static final	<a href="#"><u>PROP_DEBUG_VOD_CAPTION_LANGUAGE_SELECTION</u></a> Timed Text Property "debugVODCaptionLanguageSelection": adds info level logging regarding language selection Value: <b>debugVODCaptionLanguageSelection</b>

public static final	<a href="#"><u>PROP_DEBUG_VOD_CAPTION_PROVIDER_DETERMINATION</u></a> Timed Text Property "debugVODCaptionProviderDetermination": adds info level logging regarding caption provider determination for a given VOD asset Value: <b>debugVODCaptionProviderDetermination</b>
public static final	<a href="#"><u>PROP_DEBUG_WEBVTT_CAPTION_PARSER</u></a> Timed Text Property "debugWebVTTCaptionParser": adds info level logging regarding the WebVTT caption parser Value: <b>debugWebVTTCaptionParser</b>
public static final	<a href="#"><u>PROP_DEBUG_WEBVTT_CAPTION_PARSER_TIME</u></a> Timed Text Property "debugWebVTTCaptionParserTime": adds info level logging regarding how long it took parser to index Value: <b>debugWebVTTCaptionParserTime</b>
public static final	<a href="#"><u>PROP_DEBUG_WEBVTT_VOD_CAPTION_COMPANION_FILES</u></a> Timed Text Property "debugWebVTTVODCaptionFileDetermination": adds info level logging regarding caption companion file determination for a given VOD asset for WebVTT provider Value: <b>debugWebVTTVODCaptionFileDetermination</b>
public static final	<a href="#"><u>PROP_LIVE_CAPTION_DEFAULT_LANGUAGES</u></a> Timed Text Property "captionLiveIngestLanguages": comma separated ISO-639 language codes determining what languages are expected in the live stream Value: <b>captionLiveIngestLanguages</b>
public static final	<a href="#"><u>PROP_LIVE_CAPTION_INGEST_AMF_ID_STRING</u></a> Timed Text Property "captionLiveIngestAMFIDString": to change AMF identifier from default "onTextData" to something else Value: <b>captionLiveIngestAMFIDString</b>
public static final	<a href="#"><u>PROP_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608_ENABLE_FIELD1</u></a> Timed Text Property "captionLiveIngestOnCaptionCEA608EnableField2": When ingesting AMF onCaption 708 data events (actually CEA-608), controls whether CEA-608 field 1 is enabled for decoding Value: <b>captionLiveIngestOnCaptionCEA608EnableField1</b>
public static final	<a href="#"><u>PROP_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608_ENABLE_FIELD2</u></a> Timed Text Property "captionLiveIngestOnCaptionCEA608EnableField2": When ingesting AMF onCaption 708 data events (actually CEA-608), controls whether CEA-608 field 2 is enabled for decoding Value: <b>captionLiveIngestOnCaptionCEA608EnableField2</b>
public static final	<a href="#"><u>PROP_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608_IDLE_TIME_TO_FLUSH_DECODER</u></a> Timed Text Property "captionLiveIngestOnCaptionCEA608IdleTimeToFlushDecoder" Controls allowed amount of idle time when parsing CEA-608 captions when ingested from AMF 708 data events Value: <b>captionLiveIngestOnCaptionCEA608IdleTimeToFlushDecoder</b>
public static final	<a href="#"><u>PROP_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608_LANGUAGE_ID</u></a> Timed Text Property "captionLiveIngestOnCaptionCEA608LanguageID": When ingesting AMF onCaption 708 data events (actually CEA-608), controls the language ID used to add captions as onTextData Value: <b>captionLiveIngestOnCaptionCEA608LanguageID</b>

public static final	<a href="#"><u>PROP_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608_MAX_WARN_MESSAGES</u></a> Timed Text Property "captionLiveIngestOnCaptionCEA608MaxWarnMessages": When ingesting AMF onCaption 708 data events (actually CEA-608), controls the maximum number of warning messages that are logged by the decoder Value: <b>captionLiveIngestOnCaptionCEA608MaxWarnMessages</b>
public static final	<a href="#"><u>PROP_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608_TRACK_NUMBER</u></a> Timed Text Property "captionLiveIngestOnCaptionCEA608TrackNumber": When ingesting AMF onCaption 708 data events (actually CEA-608), controls the track number used to add captions as onTextData Value: <b>captionLiveIngestOnCaptionCEA608TrackNumber</b>
public static final	<a href="#"><u>PROP_LIVE_CAPTION_INGEST_ON_CAPTION_INFO_STRING</u></a> Timed Text Property "captionLiveIngestOnCaptionInfoString": to change AMF identifier from default "onCaptionInfo" to something else Value: <b>captionLiveIngestOnCaptionInfoString</b>
public static final	<a href="#"><u>PROP_LIVE_CAPTION_INGEST_ON_CAPTION_STRING</u></a> Timed Text Property "captionLiveIngestOnCaptionString": to change AMF identifier from default "onCaption" to something else Value: <b>captionLiveIngestOnCaptionString</b>
public static final	<a href="#"><u>PROP_LIVE_CAPTION_INGEST_TYPE</u></a> Timed Text Property "captionLiveIngestType": type of live caption ingestion. Value: <b>captionLiveIngestType</b>
public static final	<a href="#"><u>PROP_LIVE_CAPTION_USE_STREAM_NAME_GROUPS</u></a> Timed Text Property "captionLiveIngestUseStreamNameGroups": to enable use of smil files to determine available languages and ingest type Value: <b>captionLiveIngestUseStreamNameGroups</b>
public static final	<a href="#"><u>PROP_MAXIMUM_CAPTION_DURATION</u></a> Timed Text Property "maximumCaptionDuration": used to determine maximum caption duration Value: <b>maximumCaptionDuration</b>
public static final	<a href="#"><u>PROP_RTMP_CAPTION_LANGUAGES</u></a> Timed Text Property "rtmpCaptionLanguages": to enabled captionLanguage selection for RTMP Type String. Value: <b>rtmpCaptionLanguages</b>
public static final	<a href="#"><u>PROP_RTMP_VOD_CAPTIONS_ENABLED</u></a> Timed Text Property "rtmpVODCaptionsEnabled": to enabled VOD captions for RTMP streaming. Value: <b>rtmpVODCaptionsEnabled</b>
public static final	<a href="#"><u>PROP_SANJOSE_CAPTION_LANGUAGES</u></a> Timed Text Property "sanJoseCaptionLanguages": to enabled captionLanguage selection for San Jose streaming. Value: <b>sanJoseCaptionLanguages</b>
public static final	<a href="#"><u>PROP_SANJOSE_VOD_CAPTIONS_ENABLED</u></a> Timed Text Property "sanJoseVODCaptionsEnabled": to enabled VOD captions for San Jose streaming. Value: <b>sanJoseVODCaptionsEnabled</b>

public static final	<a href="#"><u>PROP_SRT_CAPTION_CHARACTER_SET</u></a> Timed Text Property "srtReaderCaptionCharacterSet": used to determine character set when ingesting SRT captions Overrides general property "captionCharacterSet" Value: <b>srtReaderCaptionCharacterSet</b>
public static final	<a href="#"><u>PROP_TTML_CAPTION_CHARACTER_SET</u></a> Timed Text Property "ttmlReaderCaptionCharacterSet": used to determine character set when ingesting TTML captions Overrides general property "captionCharacterSet" Value: <b>ttmlReaderCaptionCharacterSet</b>
public static final	<a href="#"><u>PROP_TTML_INLINE_TAGS_IN_TEXT</u></a> Timed Text Property "ttmlInlineTagsInText": TTML parser includes in-line styling and line-break tags in the sample text Value: <b>ttmlInlineTagsInText</b>
public static final	<a href="#"><u>PROP_UNDEFINED_LANG_CODE</u></a> Timed Text Property "captionUndefinedLanguageId": if incoming caption language is undefined, use this language ID. Value: <b>captionUndefinedLanguageId</b>
public static final	<a href="#"><u>PROP_VOD_CAPTIONS_ENABLED</u></a> Timed Text Property "vodCaptionsEnabled": to enabled VOD captions for all supported protocols Value: <b>vodCaptionsEnabled</b>
public static final	<a href="#"><u>PROP_WEBVTT_CONVERTER_INSERT_ERASES_IN_GAPS</u></a> Timed Text Property "webVTTCaptionConverterInsertErases": insert erase captions if gaps between captions are greater than PROP_MAXIMUM_CAPTION_DURATION Value: <b>webVTTCaptionConverterInsertErases</b>
public static final	<a href="#"><u>QUERYSTR_CAPTION_FILENAME</u></a> Query string used for choosing caption languages. Value: <b>wowzacaptionfile</b>
public static final	<a href="#"><u>QUERYSTR_CAPTIONLANGUAGES</u></a> Query string used for choosing caption languages. Value: <b>wowzacaptionlanguages</b>
public static final	<a href="#"><u>SMIL_PARAM_IS_WOWZA_CAPTION_STREAM</u></a> Value: <b>iswowzacadaptionstream</b>
public static final	<a href="#"><u>SMIL_PARAM_WOWZA_CAPTION_INGEST_TYPE</u></a> Value: <b>wowzacadaptioningesttype</b>
public static final	<a href="#"><u>SOURCE_TYPE_3GPP</u></a> Value: <b>3GPP</b>
public static final	<a href="#"><u>SOURCE_TYPE_AMF</u></a> Value: <b>AMF</b>
public static final	<a href="#"><u>SOURCE_TYPE_SCC</u></a> Value: <b>SCC</b>
public static final	<a href="#"><u>SOURCE_TYPE_SRT</u></a> Value: <b>SRT</b>

public static final	<a href="#"><u>SOURCE_TYPE_TTML</u></a>
	Value: <b>TTML</b>
public static final	<a href="#"><u>SOURCE_TYPE_WEBVTT</u></a>
	Value: <b>WebVTT</b>
public static final	<a href="#"><u>TIMED_TEXT_TYPE_CAPTION</u></a>
	Value: <b>Caption</b>

#### Fields inherited from interface com.wowza.wms.timedtext.model.ITimedTextPrivateConstants

```
DEFAULT_AMF_CONVERTER_CLASS, DEFAULT_CEA608_CONVERTER_CLASS,
DEFAULT_CEA608_IDENTITY_CONVERTER_CLASS,
DEFAULT_PROPERTY_CUPERTINO_VOD_CAPTIONS_DELEGATE_CLASS,
DEFAULT_PROPERTY_RTMP_VOD_CAPTIONS_DELEGATE_CLASS,
DEFAULT_PROPERTY_SANJOSE_VOD_CAPTIONS_DELEGATE_CLASS, DEFAULT_TIMED_TEXT_READER_SCC,
DEFAULT_TIMED_TEXT_READER_SRT, DEFAULT_TIMED_TEXT_READER_TTML,
DEFAULT_TIMED_TEXT_READER_WEBVTT, DEFAULT_WEBVTT_CONVERTER_CLASS,
PROP_AMF_CONVERTER_CLASS, PROP_CEA608_CONVERTER_CLASS,
PROP_CEA608_MAX_COMMANDS_PER_FRAME, PROP_CUPERTINO_VOD_CAPTIONS_DELEGATE_CLASS,
PROP_LIVE_ONCAPTION_TO_AMF_CONVERTER_CLASS,
PROP_LIVE_ONCAPTION_TO_WEBVTT_CONVERTER_CLASS,
PROP_LIVE_ONTEXTDATA_TO_AMF_CONVERTER_CLASS,
PROP_LIVE_ONTEXTDATA_TO_WEBVTT_CONVERTER_CLASS, PROP_MP4_TO_AMF_CONVERTER_CLASS,
PROP_MP4_TO_CEA608_CONVERTER_CLASS, PROP_MP4_TO_WEBVTT_CONVERTER_CLASS,
PROP_RTMP_VOD_CAPTIONS_DELEGATE_CLASS, PROP_SANJOSE_VOD_CAPTIONS_DELEGATE_CLASS,
PROP_SCC_TO_AMF_CONVERTER_CLASS, PROP_SCC_TO_CEA608_CONVERTER_CLASS,
PROP_SCC_TO_WEBVTT_CONVERTER_CLASS, PROP_SRT_TO_AMF_CONVERTER_CLASS,
PROP_SRT_TO_CEA608_CONVERTER_CLASS, PROP_SRT_TO_WEBVTT_CONVERTER_CLASS,
PROP_TTML_TO_AMF_CONVERTER_CLASS, PROP_TTML_TO_CEA608_CONVERTER_CLASS,
PROP_TTML_TO_WEBVTT_CONVERTER_CLASS, PROP_WEBVTT_CONVERTER_CLASS,
PROP_WEBVTT_TO_AMF_CONVERTER_CLASS, PROP_WEBVTT_TO_CEA608_CONVERTER_CLASS,
PROP_WEBVTT_TO_WEBVTT_CONVERTER_CLASS, PROPERTY_SOURCE_DURATION,
PROPERTY_SOURCE_TIME_SCALE, PROPERTY_SOURCE_TRACK_INDEX, TIMED_TEXT_READER_EXTENSION_SCC,
TIMED_TEXT_READER_EXTENSION_SRT, TIMED_TEXT_READER_EXTENSION_TTML,
TIMED_TEXT_READER_EXTENSION_WEBVTT
```

## Fields

### TIMED\_TEXT\_TYPE\_CAPTION

```
public static final java.lang.String TIMED_TEXT_TYPE_CAPTION
```

Constant value: **Caption**

### SOURCE\_TYPE\_AMF

```
public static final java.lang.String SOURCE_TYPE_AMF
```

Constant value: **AMF**

(continued from last page)

## SOURCE\_TYPE\_TTML

```
public static final java.lang.String SOURCE_TYPE_TTML
```

Constant value: **TTML**

---

## SOURCE\_TYPE\_3GPP

```
public static final java.lang.String SOURCE_TYPE_3GPP
```

Constant value: **3GPP**

---

## SOURCE\_TYPE\_SRT

```
public static final java.lang.String SOURCE_TYPE_SRT
```

Constant value: **SRT**

---

## SOURCE\_TYPE\_SCC

```
public static final java.lang.String SOURCE_TYPE_SCC
```

Constant value: **SCC**

---

## SOURCE\_TYPE\_WEBVTT

```
public static final java.lang.String SOURCE_TYPE_WEBVTT
```

Constant value: **WebVTT**

---

## OUTPUT\_TYPE\_CEA608

```
public static final java.lang.String OUTPUT_TYPE_CEA608
```

Constant value: **CEA608**

---

## OUTPUT\_TYPE\_AMF

```
public static final java.lang.String OUTPUT_TYPE_AMF
```

Constant value: **AMF**

---

## OUTPUT\_TYPE\_TTML

```
public static final java.lang.String OUTPUT_TYPE_TTML
```

Constant value: **TTML**

---

## AMF\_TEXTTYPE\_ON\_TEXT\_DATA

```
public static final java.lang.String AMF_TEXTTYPE_ON_TEXT_DATA
```

---

(continued from last page)

Constant value: **onTextData**

---

## LANGUAGE\_LOCALE

```
public static final java.lang.String LANGUAGE_LOCALE
```

Constant value: **locale**

---

## LANGUAGE\_ID\_ENGLISH

```
public static final java.lang.String LANGUAGE_ID_ENGLISH
```

Constant value: **eng**

---

## PROP\_VOD\_CAPTIONS\_ENABLED

```
public static final java.lang.String PROP_VOD_CAPTIONS_ENABLED
```

Timed Text Property "vodCaptionsEnabled": to enabled VOD captions for all supported protocols

Type boolean. The default is true.

Add the property to Application/TimedText/Properties in Application.xml

VOD Captions can be enabled on a per protocol basis using [PROP\\_CUPERTINO\\_VOD\\_CAPTIONS\\_ENABLED](#), [PROP\\_RTMP\\_VOD\\_CAPTIONS\\_ENABLED](#), [PROP\\_SANJOSE\\_VOD\\_CAPTIONS\\_ENABLED](#).

Constant value: **vodCaptionsEnabled**

See Also:

[PROP\\_CUPERTINO\\_VOD\\_CAPTIONS\\_ENABLED](#)  
[PROP\\_RTMP\\_VOD\\_CAPTIONS\\_ENABLED](#)  
[PROP\\_SANJOSE\\_VOD\\_CAPTIONS\\_ENABLED](#)

---

## PROP\_CUPERTINO\_VOD\_CAPTIONS\_ENABLED

```
public static final java.lang.String PROP_CUPERTINO_VOD_CAPTIONS_ENABLED
```

Timed Text Property "cupertinoVODCaptionsEnabled": to enabled VOD captions for Cupertino streaming.

Type boolean. The default is true.

Add the property to Application/TimedText/Properties in Application.xml

VOD Captions can be enabled for all protocols using [PROP\\_VOD\\_CAPTIONS\\_ENABLED](#).  
Constant value: **cupertinoVODCaptionsEnabled**

See Also:

[PROP\\_VOD\\_CAPTIONS\\_ENABLED](#)

---

## PROP\_RTMP\_VOD\_CAPTIONS\_ENABLED

```
public static final java.lang.String PROP_RTMP_VOD_CAPTIONS_ENABLED
```

(continued from last page)

Timed Text Property "rtmpVODCaptionsEnabled": to enabled VOD captions for RTMP streaming.

Type boolean. The default is true.

Add the property to Application/TimedText/Properties in Application.xml

VOD Captions can be enabled for all protocols using [PROP\\_VOD\\_CAPTIONS\\_ENABLED](#).  
Constant value: **rtmpVODCaptionsEnabled**

**See Also:**

[PROP\\_VOD\\_CAPTIONS\\_ENABLED](#)

## PROP\_SANJOSE\_VOD\_CAPTIONS\_ENABLED

public static final java.lang.String **PROP\_SANJOSE\_VOD\_CAPTIONS\_ENABLED**

Timed Text Property "sanJoseVODCaptionsEnabled": to enabled VOD captions for San Jose streaming.

Type boolean. The default is true.

Add the property to Application/TimedText/Properties in Application.xml

VOD Captions can be enabled for all protocols using [PROP\\_VOD\\_CAPTIONS\\_ENABLED](#).  
Constant value: **sanJoseVODCaptionsEnabled**

**See Also:**

[PROP\\_VOD\\_CAPTIONS\\_ENABLED](#)

## PROP\_CAPTION\_LANGUAGES

public static final java.lang.String **PROP\_CAPTION\_LANGUAGES**

Timed Text Property "captionLanguages": used to select presented caption Language.

Type String. The default is [DEFAULT\\_PROPERTY\\_CAPION\\_LANGUAGES](#)

Add the property to Application/TimedText/Properties in Application.xml

VOD Caption languages can be specified on a per protocol basis using [PROP\\_CUPERTINO\\_CAPTION\\_LANGUAGES](#),  
[PROP\\_RTMP\\_CAPTION\\_LANGUAGES](#), [PROP\\_SANJOSE\\_CAPTION\\_LANGUAGES](#).  
Constant value: **captionLanguages**

**See Also:**

[PROP\\_CUPERTINO\\_CAPTION\\_LANGUAGES](#)  
[PROP\\_RTMP\\_CAPTION\\_LANGUAGES](#)  
[PROP\\_SANJOSE\\_CAPTION\\_LANGUAGES](#)

## DEFAULT\_PROPERTY\_CAPION\_LANGUAGES

public static final java.lang.String **DEFAULT\_PROPERTY\_CAPION\_LANGUAGES**

Default value for Timed Text Property "captionLanguages".

Default value is "\*", meaning present all found languages.  
Constant value: **\***

**See Also:**

[PROP\\_CAPTION\\_LANGUAGES](#)

## PROP\_CUPERTINO\_CAPTION\_LANGUAGES

public static final java.lang.String **PROP\_CUPERTINO\_CAPTION\_LANGUAGES**

---

(continued from last page)

Timed Text Property "cupertinoCaptionLanguages": to enabled captionLanguage selection for Cupertino streaming.

Type String. The default is [DEFAULT\\_PROPERTY\\_CAPION\\_LANGUAGES](#)

Add the property to Application/TimedText/Properties in Application.xml

VOD Captions can be enabled for all protocols using [PROP\\_CAPTION\\_LANGUAGES](#).

Constant value: **cupertinoCaptionLanguages**

**See Also:**

[PROP\\_CAPTION\\_LANGUAGES](#)

---

## PROP\_RTMP\_CAPTION\_LANGUAGES

```
public static final java.lang.String PROP_RTMP_CAPTION_LANGUAGES
```

Timed Text Property "rtmpCaptionLanguages": to enabled captionLanguage selection for RTMP Type String. The default is [DEFAULT\\_PROPERTY\\_CAPION\\_LANGUAGES](#)

Add the property to Application/TimedText/Properties in Application.xml

VOD Captions can be enabled for all protocols using [PROP\\_CAPTION\\_LANGUAGES](#).

Constant value: **rtmpCaptionLanguages**

**See Also:**

[PROP\\_CAPTION\\_LANGUAGES](#)

---

## PROP\_SANJOSE\_CAPTION\_LANGUAGES

```
public static final java.lang.String PROP_SANJOSE_CAPTION_LANGUAGES
```

Timed Text Property "sanJoseCaptionLanguages": to enabled captionLanguage selection for San Jose streaming.

Type String. The default is [DEFAULT\\_PROPERTY\\_CAPION\\_LANGUAGES](#)

Add the property to Application/TimedText/Properties in Application.xml

VOD Captions can be enabled for all protocols using [PROP\\_CAPTION\\_LANGUAGES](#).

Constant value: **sanJoseCaptionLanguages**

**See Also:**

[PROP\\_CAPTION\\_LANGUAGES](#)

---

## PROP\_DEBUG\_VOD\_CAPTION\_LANGUAGE\_SELECTION

```
public static final java.lang.String PROP_DEBUG_VOD_CAPTION_LANGUAGE_SELECTION
```

Timed Text Property "debugVODCaptionLanguageSelection": adds info level logging regarding language selection

Type boolean. The default is false.

Add the property to Application/TimedText/Properties in Application.xml

Can be enabled on a per protocol basis using [PROP\\_DEBUG\\_CUPERTINO\\_VOD\\_CAPTION\\_LANGUAGE\\_SELECTION](#),  
[PROP\\_DEBUG\\_RTMP\\_VOD\\_CAPTION\\_LANGUAGE\\_SELECTION](#),

[PROP\\_DEBUG\\_SANJOSE\\_VOD\\_CAPTION\\_LANGUAGE\\_SELECTION](#).

Constant value: **debugVODCaptionLanguageSelection**

---

## PROP\_DEBUG\_CUPERTINO\_VOD\_CAPTION\_LANGUAGE\_SELECTION

```
public static final java.lang.String PROP_DEBUG_CUPERTINO_VOD_CAPTION_LANGUAGE_SELECTION
```

(continued from last page)

Timed Text Property "cupertinoDebugVODCaptionLanguageSelection": adds info level logging regarding language selection for Cupertino Streaming

Type boolean. The default is false.

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **cupertinoDebugVODCaptionLanguageSelection**

See Also:

[PROP\\_DEBUG\\_VOD\\_CAPTION\\_LANGUAGE\\_SELECTION](#)

## PROP\_DEBUG\_RTMP\_VOD\_CAPTION\_LANGUAGE\_SELECTION

public static final java.lang.String **PROP\_DEBUG\_RTMP\_VOD\_CAPTION\_LANGUAGE\_SELECTION**

Timed Text Property "rtmpDebugVODCaptionLanguageSelection": adds info level logging regarding language selection for RTMP Streaming

Type boolean. The default is false.

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **rtmpDebugVODCaptionLanguageSelection**

See Also:

[PROP\\_DEBUG\\_VOD\\_CAPTION\\_LANGUAGE\\_SELECTION](#)

## PROP\_DEBUG\_SANJOSE\_VOD\_CAPTION\_LANGUAGE\_SELECTION

public static final java.lang.String **PROP\_DEBUG\_SANJOSE\_VOD\_CAPTION\_LANGUAGE\_SELECTION**

Timed Text Property "sanJoseDebugVODCaptionLanguageSelection": adds info level logging regarding language selection for San Jose Streaming

Type boolean. The default is false.

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **sanJoseDebugVODCaptionLanguageSelection**

See Also:

[PROP\\_DEBUG\\_VOD\\_CAPTION\\_LANGUAGE\\_SELECTION](#)

## PROP\_DEBUG\_VOD\_CAPTION\_PROVIDER\_DETERMINATION

public static final java.lang.String **PROP\_DEBUG\_VOD\_CAPTION\_PROVIDER\_DETERMINATION**

Timed Text Property "debugVODCaptionProviderDetermination": adds info level logging regarding caption provider determination for a given VOD asset

Type boolean. The default is false.

Add the property to Application/TimedText/Properties in Application.xml

Can be enabled on a per protocol basis using [PROP\\_DEBUG\\_CUPERTINO\\_VOD\\_CAPTION\\_PROVIDER\\_DETERMINATION](#), [PROP\\_DEBUG\\_RTMP\\_VOD\\_CAPTION\\_PROVIDER\\_DETERMINATION](#), [PROP\\_DEBUG\\_SANJOSE\\_VOD\\_CAPTION\\_PROVIDER\\_DETERMINATION](#).

Constant value: **debugVODCaptionProviderDetermination**

(continued from last page)

## PROP\_DEBUG\_CUPERTINO\_VOD\_CAPTION\_PROVIDER\_DETERMINATION

```
public static final java.lang.String  
PROP_DEBUG_CUPERTINO_VOD_CAPTION_PROVIDER_DETERMINATION
```

Timed Text Property "cupertinoDebugVODCaptionProviderDetermination": adds info level logging regarding caption provider determination for a given VOD asset for Cupertino Streaming.

Type boolean. The default is false.

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **cupertinoDebugVODCaptionProviderDetermination**

See Also:

[PROP\\_DEBUG\\_VOD\\_CAPTION\\_PROVIDER\\_DETERMINATION](#)

---

## PROP\_DEBUG\_RTMP\_VOD\_CAPTION\_PROVIDER\_DETERMINATION

```
public static final java.lang.String  
PROP_DEBUG_RTMP_VOD_CAPTION_PROVIDER_DETERMINATION
```

Timed Text Property "rtmpDebugVODCaptionProviderDetermination": adds info level logging regarding caption provider determination for a given VOD asset for RTMP Streaming.

Type boolean. The default is false.

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **rtmpDebugVODCaptionProviderDetermination**

See Also:

[PROP\\_DEBUG\\_VOD\\_CAPTION\\_PROVIDER\\_DETERMINATION](#)

---

## PROP\_DEBUG\_SANJOSE\_VOD\_CAPTION\_PROVIDER\_DETERMINATION

```
public static final java.lang.String  
PROP_DEBUG_SANJOSE_VOD_CAPTION_PROVIDER_DETERMINATION
```

Timed Text Property "sanJoseVODDebugCaptionProviderDetermination": adds info level logging regarding caption provider determination for a given VOD asset for San Jose Streaming.

Type boolean. The default is false.

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **sanJoseVODDebugCaptionProviderDetermination**

See Also:

[PROP\\_DEBUG\\_VOD\\_CAPTION\\_PROVIDER\\_DETERMINATION](#)

---

## PROP\_DEBUG\_VOD\_CAPTION\_COMPANION\_FILES

```
public static final java.lang.String PROP_DEBUG_VOD_CAPTION_COMPANION_FILES
```

(continued from last page)

Timed Text Property "debugVODCaptionFileDetermination": adds info level logging regarding caption companion file determination for a given VOD asset

Type boolean. The default is false.

Add the property to Application/TimedText/Properties in Application.xml

Can be enabled on a per provider basis as well using [PROP\\_DEBUG\\_TTML\\_VOD\\_CAPTION\\_COMPANION\\_FILES](#)  
Constant value: **debugVODCaptionFileDetermination**

## PROP\_UNDEFINED\_LANG\_CODE

```
public static final java.lang.String PROP_UNDEFINED_LANG_CODE
```

Timed Text Property "captionUndefinedLanguageId": if incoming caption language is undefined, use this language ID.

Type String. If not defined, the Locale's 3 character language code is used. See Locale.getDefault().getISO3Language()

Add the property to Application/TimedText/Properties in Application.xml  
Constant value: **captionUndefinedLanguageId**

## PROP\_CAPTION\_FILE\_NAMING\_RULE

```
public static final java.lang.String PROP_CAPTION_FILE_NAMING_RULE
```

Timed Text Property "captionFileNamingRule": to change logic determining the caption companion file naming scheme

Type String. The default value is [DEFAULT\\_PROPERTY\\_CAPTION\\_FILE\\_NAMING\\_RULE](#).

This string is used to find the caption file for a given VOD asset. It is used in conjunction with [PROP\\_CAPTION\\_PATH\\_NAMING\\_RULE](#), which determines the file location.

It is comprised of variables preceded by a dollar sign and enclosed in squiggly brackets. For example: "\${var1}.{var2}"

Valid variables include these:

- \${SourceStreamName} : The baseName of the VOD file
- \${SourcebasePath} : The base path of the VOD asset
- \${FileExtension} : Registered file extensions with the Timed Text Provider

and these standard ones:

- \${com.wowza.wms.context.VHost}
- \${com.wowza.wms.context.VHostConfigHome}, appInstance.getVHost().getHomePath();
- \${com.wowza.wms.context.Application}, appInstance.getApplication().getName();
- \${com.wowza.wms.context.ApplicationInstance}, appInstance.getName();

Add the property to Application/TimedText/Properties in Application.xml  
Constant value: **captionFileNamingRule**

See Also:

[DEFAULT\\_PROPERTY\\_CAPTION\\_FILE\\_NAMING\\_RULE](#)

## DEFAULT\_PROPERTY\_CAPTION\_FILE\_NAMING\_RULE

```
public static final java.lang.String DEFAULT_PROPERTY_CAPTION_FILE_NAMING_RULE
```

Default value for Timed Text Property "captionFileNamingRule".

Default value is "\${SourceStreamName}.\${FileExtension}"  
Constant value: **\${SourceStreamName}.\${FileExtension}**

(continued from last page)

**See Also:**[PROP\\_CAPTION\\_FILE\\_NAMING\\_RULE](#)**PROP\_CAPTION\_PATH\_NAMING\_RULE****public static final java.lang.String PROP\_CAPTION\_PATH\_NAMING\_RULE**

Timed Text Property "captionPathNamingRule": to change logic determining the caption companion path

Type String. The default value is [DEFAULT\\_PROPERTY\\_CAPTION\\_PATH\\_NAMING\\_RULE](#).This string is used to find the caption file location for a given VOD asset. It is used in conjunction with [PROP\\_CAPTION\\_FILE\\_NAMING\\_RULE](#), which determines the file name.

It is comprised of variables preceded by a dollar sign and enclosed in squiggly brackets. For example: "\${var1}.{var2}"

Valid variables include these:

- \${SourceStreamName} : The baseName of the VOD file
- \${SourcebasePath} : The base path of the VOD asset
- \${FileExtension} : Registered file extensions with the Timed Text Provider

and these standard ones:

- \${com.wowza.wms.context.VHost}
- \${com.wowza.wms.context.VHostConfigHome}, appInstance.getVHost().getHomePath();
- \${com.wowza.wms.context.Application}, appInstance.getApplication().getName();
- \${com.wowza.wms.context.ApplicationInstance}, appInstance.getName();

Add the property to Application/TimedText/Properties in Application.xml  
Constant value: **captionPathNamingRule****See Also:**[DEFAULT\\_PROPERTY\\_CAPTION\\_PATH\\_NAMING\\_RULE](#)**DEFAULT\_PROPERTY\_CAPTION\_PATH\_NAMING\_RULE****public static final java.lang.String DEFAULT\_PROPERTY\_CAPTION\_PATH\_NAMING\_RULE**Default value for Timed Text Property "captionFileNamingRule". Default value is "\${SourcebasePath}"  
Constant value:  **\${sourcebasePath}****See Also:**[PROP\\_CAPTION\\_PATH\\_NAMING\\_RULE](#)**PROP\_MAXIMUM\_CAPTION\_DURATION****public static final java.lang.String PROP\_MAXIMUM\_CAPTION\_DURATION**

Timed Text Property "maximumCaptionDuration": used to determine maximum caption duration

Type Integer. In ms. The default is [DEFAULT\\_MAXIMUM\\_CAPTION\\_DURATION](#)

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **maximumCaptionDuration**

---

(continued from last page)

## DEFAULT\_MAXIMUM\_CAPTION\_DURATION

```
public static final int DEFAULT_MAXIMUM_CAPTION_DURATION
```

Default value for Timed Text Property "maximumCaptionDuration".

Default value is 10000.  
Constant value: **5000**

See Also:

[PROP\\_MAXIMUM\\_CAPTION\\_DURATION](#)

---

## PROP\_CAPTION\_CHARACTER\_SET

```
public static final java.lang.String PROP_CAPTION_CHARACTER_SET
```

Timed Text Property "captionCharacterSet": used to determine character set when ingesting captions

Type String. The default is [DEFAULT\\_CAPTION\\_CHARACTER\\_SET](#)

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **captionCharacterSet**

---

## DEFAULT\_CAPTION\_CHARACTER\_SET

```
public static final java.lang.String DEFAULT_CAPTION_CHARACTER_SET
```

Default value for Timed Text Property "captionCharacterSet".

Default value is "UTF-8"  
Constant value: **UTF-8**

See Also:

[PROP\\_CAPTION\\_CHARACTER\\_SET](#)

---

## PROP\_SRT\_CAPTION\_CHARACTER\_SET

```
public static final java.lang.String PROP_SRT_CAPTION_CHARACTER_SET
```

Timed Text Property "srtReaderCaptionCharacterSet": used to determine character set when ingesting SRT captions  
Overrides general property "captionCharacterSet"

Type String. The default is [DEFAULT\\_CAPTION\\_CHARACTER\\_SET](#)  
Constant value: **srtReaderCaptionCharacterSet**

See Also:

Add the property to Application/TimedText/Properties in Application.xml

---

## PROP\_TTML\_CAPTION\_CHARACTER\_SET

```
public static final java.lang.String PROP_TTML_CAPTION_CHARACTER_SET
```

Timed Text Property "ttmlReaderCaptionCharacterSet": used to determine character set when ingesting TTML  
captions Overrides general property "captionCharacterSet"

Type String. The default is [DEFAULT\\_CAPTION\\_CHARACTER\\_SET](#)

---

(continued from last page)

Constant value: **ttmlReaderCaptionCharacterSet**

**See Also:**

Add the property to Application/TimedText/Properties in Application.xml

---

## PROP\_3GPP\_CAPTION\_CHARACTER\_SET

public static final java.lang.String **PROP\_3GPP\_CAPTION\_CHARACTER\_SET**

Timed Text Property "reader3GPPCaptionCharacterSet": used to determine character set when ingesting TTML captions  
Overrides general property "captionCharacterSet"

Type String. The default is [DEFAULT\\_CAPTION\\_CHARACTER\\_SET](#)  
Constant value: **reader3GPPCaptionCharacterSet**

**See Also:**

Add the property to Application/TimedText/Properties in Application.xml

---

## PROP\_CEA608\_CONVERTER\_COLOR

public static final java.lang.String **PROP\_CEA608\_CONVERTER\_COLOR**

Timed Text Property "cea608CaptionConverterColor": color of text in default CEA caption converter.

Type Integer. Default is [ClosedCaptionCEA608Utils.COL0\\_WHITE](#)

Add the property to Application/TimedText/Properties in Application.xml  
Constant value: **cea608CaptionConverterColor**

**See Also:**

[ClosedCaptionCEA608Utils.COL0\\_WHITE](#)  
[ClosedCaptionCEA608Utils.COL0\\_GREEN](#)  
[ClosedCaptionCEA608Utils.COL0\\_BLUE](#)  
[ClosedCaptionCEA608Utils.COL0\\_CYAN](#)  
[ClosedCaptionCEA608Utils.COL0\\_RED](#)  
[ClosedCaptionCEA608Utils.COL0\\_YELLOW](#)  
[ClosedCaptionCEA608Utils.COL0\\_MAGENTA](#)

---

## PROP\_CEA608\_CONVERTER\_CHARS\_PER\_LINE

public static final java.lang.String **PROP\_CEA608\_CONVERTER\_CHARS\_PER\_LINE**

Timed Text Property "cea608CaptionConverterCharsPerLine": max number characters per line in default CEA caption converter.

Type Integer. Default is [ClosedCaptionCEA608Utils.COUNT\\_COLS](#)

Add the property to Application/TimedText/Properties in Application.xml  
Constant value: **cea608CaptionConverterCharsPerLine**

---

## PROP\_CEA608\_CONVERTER\_CHARACTER\_SET

public static final java.lang.String **PROP\_CEA608\_CONVERTER\_CHARACTER\_SET**

(continued from last page)

Timed Text Property "cea608CaptionConverterCharacterSet":character set in default CEA caption converter.

Type String. Default is "UTF-8"

Add the property to Application/TimedText/Properties in Application.xml  
Constant value: **cea608CaptionConverterCharacterSet**

## **PROP\_CUPERTINO\_VOD\_REMOVE\_EXISTING\_CEA608**

```
public static final java.lang.String PROP_CUPERTINO_VOD_REMOVE_EXISTING_CEA608
```

Timed Text Property "cupertinoVODCaptionsRemoveExistingCEA608": if existing VOD has 608, remove them

Type Boolean. Default is false.

Add the property to Application/TimedText/Properties in Application.xml  
Constant value: **cupertinoVODCaptionsRemoveExistingCEA608**

## **PROP\_CUPERTINO\_VOD\_USE\_WEBVTT**

```
public static final java.lang.String PROP_CUPERTINO_VOD_USE_WEBVTT
```

Timed Text Property "cupertinoVODCaptionsUseWebVTT"

Type Boolean. Default is false.

Add the property to Application/TimedText/Properties in Application.xml  
Constant value: **cupertinoVODCaptionsUseWebVTT**

## **PROP\_CUPERTINO\_LIVE\_USE\_WEBVTT**

```
public static final java.lang.String PROP_CUPERTINO_LIVE_USE_WEBVTT
```

Timed Text Property "cupertinoLiveCaptionsUseWebVTT"

Type Boolean. Default is false.

Add the property to Application/TimedText/Properties in Application.xml  
Constant value: **cupertinoLiveCaptionsUseWebVTT**

## **PROP\_AMF\_CONVERTER\_TEXT\_TYPE**

```
public static final java.lang.String PROP_AMF_CONVERTER_TEXT_TYPE
```

Timed Text Property "amfCaptionConverterTextType":AMF text type used in conversion

Type String. Default is [AMF\\_TEXTTYPE\\_ON\\_TEXT\\_DATA](#)

Add the property to Application/TimedText/Properties in Application.xml  
Constant value: **amfCaptionConverterTextType**

## **DEFAULT\_AMF\_CONVERTER\_TEXT\_TYPE**

```
public static final java.lang.String DEFAULT_AMF_CONVERTER_TEXT_TYPE
```

Constant value: **onTextData**

## **PROP\_AMF\_CONVERTER\_TRACK\_INDEX**

```
public static final java.lang.String PROP_AMF_CONVERTER_TRACK_INDEX
```

(continued from last page)

Timed Text Property "amfCaptionConverterTrackIndex": track index used for captions

Type Integer.

Add the property to Application/TimedText/Properties in Application.xml  
 Constant value: **amfCaptionConverterTrackIndex**

## **DEFAULT\_AMF\_CONVERTER\_TRACK\_INDEX**

```
public static final int DEFAULT_AMF_CONVERTER_TRACK_INDEX
```

Constant value: **99**

## **PROP\_AMF\_CONVERTER\_INSERT\_ERASES\_IN\_GAPS**

```
public static final java.lang.String PROP_AMF_CONVERTER_INSERT_ERASES_IN_GAPS
```

Timed Text Property "amfCaptionConverterInsertErases": insert erase captions if gaps between captions are greater than PROP\_MAXIMUM\_CAPTION\_DURATION

Type boolean. The default is true.

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **amfCaptionConverterInsertErases**

## **PROP\_WEBVTT\_CONVERTER\_INSERT\_ERASES\_IN\_GAPS**

```
public static final java.lang.String PROP_WEBVTT_CONVERTER_INSERT_ERASES_IN_GAPS
```

Timed Text Property "webVTTCaptionConverterInsertErases": insert erase captions if gaps between captions are greater than PROP\_MAXIMUM\_CAPTION\_DURATION

Type boolean. The default is false.

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **webVTTCaptionConverterInsertErases**

## **PROP\_DEBUG\_TTML\_VOD\_CAPTION\_COMPANION\_FILES**

```
public static final java.lang.String PROP_DEBUG_TTML_VOD_CAPTION_COMPANION_FILES
```

Timed Text Property "debugTTMLVODCaptionFileDetermination": adds info level logging regarding caption companion file determination for a given VOD asset for TTML Streaming.

Type boolean. The default is false.

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **debugTTMLVODCaptionFileDetermination**

See Also:

[PROP\\_DEBUG\\_VOD\\_CAPTION\\_COMPANION\\_FILES](#)

## **PROP\_DEBUG\_TTML\_CAPTION\_PARSER**

```
public static final java.lang.String PROP_DEBUG_TTML_CAPTION_PARSER
```

(continued from last page)

Timed Text Property "debugTTMLCaptionParser": adds info level logging regarding the TTML caption parser

Type boolean. The default is false.

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **debugTTMLCaptionParser**

---

## PROP\_DEBUG\_TTML\_CAPTION\_PARSER\_TIME

public static final java.lang.String PROP\_DEBUG\_TTML\_CAPTION\_PARSER\_TIME

Timed Text Property "debugTTMLCaptionParserTime": adds info level logging regarding how long it took parser to index

Type boolean. The default is false.

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **debugTTMLCaptionParserTime**

---

## PROP\_TTML\_INLINE\_TAGS\_IN\_TEXT

public static final java.lang.String PROP\_TTML\_INLINE\_TAGS\_IN\_TEXT

Timed Text Property "ttmlInlineTagsInText": TTML parser includes in-line styling and line-break tags in the sample text

Type boolean. The default is false.

Add the property to Application/TimedText/Properties in Application.xml

Example sample: "

I can linebreak

or change fontWeight to bold in the middle of a sample." If ttmlInlineTagsInText=true then the parser provides the sample text as: "I can linebreak

or change fontWeight to bold in the middle of a sample." If ttmlInlineTagsInText=false then the parser provides the sample text as: "I can linebreak or change fontWeight to bold in the middle of a sample."

Constant value: **ttmlInlineTagsInText**

---

## PROP\_DEBUG\_SRT\_VOD\_CAPTION\_COMPANION\_FILES

public static final java.lang.String PROP\_DEBUG\_SRT\_VOD\_CAPTION\_COMPANION\_FILES

Timed Text Property "debugSRTVODCaptionFileDetermination": adds info level logging regarding caption companion file determination for a given VOD asset for SRT provider

Type boolean. The default is false.

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **debugSRTVODCaptionFileDetermination**

See Also:

[PROP\\_DEBUG\\_VOD\\_CAPTION\\_COMPANION\\_FILES](#)

---

## PROP\_DEBUG\_SRT\_CAPTION\_PARSER

public static final java.lang.String PROP\_DEBUG\_SRT\_CAPTION\_PARSER

---

(continued from last page)

Timed Text Property "debugSRTCaptionParser": adds info level logging regarding the SRT caption parser

Type boolean. The default is false.

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **debugSRTCaptionParser**

---

## PROP\_DEBUG\_SRT\_CAPTION\_PARSER\_TIME

```
public static final java.lang.String PROP_DEBUG_SRT_CAPTION_PARSER_TIME
```

Timed Text Property "debugSRTCaptionParserTime": adds info level logging regarding how long it took parser to index

Type boolean. The default is false.

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **debugSRTCaptionParserTime**

---

## PROP\_DEBUG\_WEBVTT\_VOD\_CAPTION\_COMPANION\_FILES

```
public static final java.lang.String PROP_DEBUG_WEBVTT_VOD_CAPTION_COMPANION_FILES
```

Timed Text Property "debugWebVTTvodCaptionFileDetermination": adds info level logging regarding caption companion file determination for a given VOD asset for WebVTT provider

Type boolean. The default is false.

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **debugWebVTTvodCaptionFileDetermination**

See Also:

[PROP\\_DEBUG\\_VOD\\_CAPTION\\_COMPANION\\_FILES](#)

---

## PROP\_DEBUG\_WEBVTT\_CAPTION\_PARSER

```
public static final java.lang.String PROP_DEBUG_WEBVTT_CAPTION_PARSER
```

Timed Text Property "debugWebVTTCaptionParser": adds info level logging regarding the WebVTT caption parser

Type boolean. The default is false.

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **debugWebVTTCaptionParser**

---

## PROP\_DEBUG\_WEBVTT\_CAPTION\_PARSER\_TIME

```
public static final java.lang.String PROP_DEBUG_WEBVTT_CAPTION_PARSER_TIME
```

Timed Text Property "debugWebVTTCaptionParserTime": adds info level logging regarding how long it took parser to index

Type boolean. The default is false.

Add the property to Application/TimedText/Properties in Application.xml

---

(continued from last page)

Constant value: **debugWebVTTCaptionParserTime**

---

## PROP\_DEBUG\_SCC\_VOD\_CAPTION\_COMPANION\_FILES

```
public static final java.lang.String PROP_DEBUG_SCC_VOD_CAPTION_COMPANION_FILES
```

Timed Text Property "debugSCCVODCaptionFileDetermination": adds info level logging regarding caption companion file determination for a given VOD asset for SCC provider

Type boolean. The default is false.

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **debugSCCVODCaptionFileDetermination**

See Also:

[PROP\\_DEBUG\\_VOD\\_CAPTION\\_COMPANION\\_FILES](#)

---

## PROP\_DEBUG\_SCC\_CAPTION\_PARSER

```
public static final java.lang.String PROP_DEBUG_SCC_CAPTION_PARSER
```

Timed Text Property "debugSCCCaptionParser": adds info level logging regarding the SCC caption parser

Type boolean. The default is false.

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **debugSCCCaptionParser**

---

## PROP\_DEBUG\_SCC\_CAPTION\_PARSER\_TIME

```
public static final java.lang.String PROP_DEBUG_SCC_CAPTION_PARSER_TIME
```

Timed Text Property "debugSCCCaptionParserTime": adds info level logging regarding how long it took parser to index

Type boolean. The default is false.

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **debugSCCCaptionParserTime**

---

## PROP\_DEBUG\_3GPP\_CAPTION\_PARSER

```
public static final java.lang.String PROP_DEBUG_3GPP_CAPTION_PARSER
```

Timed Text Property "debug3GPPCaptionParser": adds info level logging regarding the 3GPP parser

Type boolean. The default is false.

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **debug3GPPCaptionParser**

---

## PROP\_DEBUG\_3GPP\_CAPTION\_PARSER\_TIME

```
public static final java.lang.String PROP_DEBUG_3GPP_CAPTION_PARSER_TIME
```

(continued from last page)

Timed Text Property "debug3GPPCaptionParserTime": adds info level logging regarding how long it took parser to index

Type boolean. The default is false.

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **debug3GPPCaptionParserTime**

## **QUERYSTR\_CAPTIONLANGUAGES**

```
public static final java.lang.String QUERYSTR_CAPTIONLANGUAGES
```

Query string used for choosing caption languages. Language must be presented using ISO 3 letter language codes.  
Constant value: **wowzacaptionlanguages**

## **PROP\_CAPTION\_LANGUAGE\_QUERY\_PARAMETER**

```
public static final java.lang.String PROP_CAPTION_LANGUAGE_QUERY_PARAMETER
```

HTTP Streamer Property "captionLanguageQueryParameter": used for changing query string param used for caption language selection. Note: Must go into HTTPStreamers.xml under Properties for each streamer type

Type String. The default is "wowzacaptionlanguages"

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **captionLanguageQueryParameter**

## **QUERYSTR\_CAPTION\_FILENAME**

```
public static final java.lang.String QUERYSTR_CAPTION_FILENAME
```

Query string used for choosing caption languages. Language must be presented using ISO 3 letter language codes.  
Constant value: **wowzacaptionfile**

## **PROP\_CAPTION\_FILENAME\_QUERY\_PARAMETER**

```
public static final java.lang.String PROP_CAPTION_FILENAME_QUERY_PARAMETER
```

HTTP Streamer Property "captionFilenameQueryParameter": used for changing query string param used for caption file selection. Note: Must go into HTTPStreamers.xml under Properties for each streamer type

Type String. The default is "wowzacaptionfile"

Add the property to Application/TimedText/Properties in Application.xml

Constant value: **captionFilenameQueryParameter**

## **PROP\_LIVE\_CAPTION\_INGEST\_AMF\_ID\_STRING**

```
public static final java.lang.String PROP_LIVE_CAPTION_INGEST_AMF_ID_STRING
```

Timed Text Property "captionLiveIngestAMFIDString": to change AMF identifier from default "onTextData" to something else

Type String. The default value is "onTextData".

Add the property to Application/TimedText/Properties in Application.xml  
Constant value: **captionLiveIngestAMFIDString**

## PROP\_LIVE\_CAPTION\_USE\_STREAM\_NAME\_GROUPS

```
public static final java.lang.String PROP_LIVE_CAPTION_USE_STREAM_NAME_GROUPS
```

Timed Text Property "captionLiveIngestUseStreamNameGroups": to enable use of smil files to determine available languages and ingest type

Type Boolean. The default value is true. Only affects streams started with .smil files using StartupStreams or stream manager.

Add the property to Application/TimedText/Properties in Application.xml  
Constant value: **captionLiveIngestUseStreamNameGroups**

---

## PROP\_LIVE\_CAPTION\_DEFAULT\_LANGUAGES

```
public static final java.lang.String PROP_LIVE_CAPTION_DEFAULT_LANGUAGES
```

Timed Text Property "captionLiveIngestLanguages": comma separated ISO-639 language codes determining what languages are expected in the live stream

Type String. The default value is the language returned by Locale.getDefault().getISO3Language().

Add the property to Application/TimedText/Properties in Application.xml  
Constant value: **captionLiveIngestLanguages**

---

## PROP\_LIVE\_CAPTION\_INGEST\_TYPE

```
public static final java.lang.String PROP_LIVE_CAPTION_INGEST_TYPE
```

Timed Text Property "captionLiveIngestType": type of live caption ingestion.

This corresponds to LiveIngestTypes in TimedTextProviders.

Type String. The default value is "onTextData". "onCaption" is also a valid value.

Add the property to Application/TimedText/Properties in Application.xml  
Constant value: **captionLiveIngestType**

---

## PROP\_DEBUG\_LIVE\_CAPTION\_INGEST

```
public static final java.lang.String PROP_DEBUG_LIVE_CAPTION_INGEST
```

Timed Text Property "debugCaptionLiveIngest"

Type Boolean. The default value is false.

Add the property to Application/TimedText/Properties in Application.xml  
Constant value: **debugCaptionLiveIngest**

---

## PROP\_DEBUG\_LIVE\_CAPTION\_ONTEXTDATA\_INGEST

```
public static final java.lang.String PROP_DEBUG_LIVE_CAPTION_ONTEXTDATA_INGEST
```

Timed Text Property "debugCaptionLiveOnTextDataIngest"

Type Boolean. The default value is false. Overrides [PROP\\_LIVE\\_CAPTION\\_DEFAULT\\_LANGUAGES](#)

Add the property to Application/TimedText/Properties in Application.xml  
Constant value: **debugCaptionLiveOnTextDataIngest**

---

## PROP\_DEBUG\_LIVE\_CAPTION\_ONCAPTION\_INGEST

```
public static final java.lang.String PROP_DEBUG_LIVE_CAPTION_ONCAPTION_INGEST
```

(continued from last page)

Timed Text Property "debugCaptionLiveOnCaptionIngest"

Type Boolean. The default value is false. Overrides [PROP\\_LIVE\\_CAPTION\\_DEFAULT\\_LANGUAGES](#)

Add the property to Application/TimedText/Properties in Application.xml  
Constant value: **debugCaptionLiveOnCaptionIngest**

## **PROP\_LIVE\_CAPTION\_INGEST\_ON\_CAPTION\_STRING**

```
public static final java.lang.String PROP_LIVE_CAPTION_INGEST_ON_CAPTION_STRING
```

Timed Text Property "captionLiveIngestOnCaptionString": to change AMF identifier from default "onCaption" to something else

Type String. The default value is "onCaption".

Add the property to Application/TimedText/Properties in Application.xml  
Constant value: **captionLiveIngestOnCaptionString**

## **PROP\_LIVE\_CAPTION\_INGEST\_ON\_CAPTION\_INFO\_STRING**

```
public static final java.lang.String PROP_LIVE_CAPTION_INGEST_ON_CAPTION_INFO_STRING
```

Timed Text Property "captionLiveIngestOnCaptionInfoString": to change AMF identifier from default "onCaptionInfo" to something else

Type String. The default value is "onCaptionInfo".

Add the property to Application/TimedText/Properties in Application.xml  
Constant value: **captionLiveIngestOnCaptionInfoString**

## **PROP\_LIVE\_CAPTION\_INGEST\_ON\_CAPTION\_CEA608\_IDLE\_TIME\_TO\_FLUSH\_DECODER**

```
public static final java.lang.String
```

```
PROP_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608_IDLE_TIME_TO_FLUSH_DECODER
```

Timed Text Property "captionLiveIngestOnCaptionCEA608IdleTimeToFlushDecoder" Controls allowed amount of idle time when parsing CEA-608 captions when ingested from AMF 708 data events

Type Integer. The default value is 250 ms.

Add the property to Application/TimedText/Properties in Application.xml  
Constant value: **captionLiveIngestOnCaptionCEA608IdleTimeToFlushDecoder**

## **PROP\_LIVE\_CAPTION\_INGEST\_ON\_CAPTION\_CEA608\_ENABLE\_FIELD1**

```
public static final java.lang.String
```

```
PROP_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608_ENABLE_FIELD1
```

Timed Text Property "captionLiveIngestOnCaptionCEA608EnableField2": When ingesting AMF onCaption 708 data events (actually CEA-608), controls whether CEA-608 field 1 is enabled for decoding

Type Boolean. Default is true;

Add the property to Application/TimedText/Properties in Application.xml  
Constant value: **captionLiveIngestOnCaptionCEA608EnableField1**

## **PROP\_LIVE\_CAPTION\_INGEST\_ON\_CAPTION\_CEA608\_ENABLE\_FIELD2**

```
public static final java.lang.String
```

```
PROP_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608_ENABLE_FIELD2
```

(continued from last page)

Timed Text Property "captionLiveIngestOnCaptionCEA608EnableField2": When ingesting AMF onCaption 708 data events (actually CEA-608), controls whether CEA-608 field 2 is enabled for decoding

Type Boolean. Default is false;

Add the property to Application/TimedText/Properties in Application.xml  
Constant value: **captionLiveIngestOnCaptionCEA608EnableField2**

## **PROP\_LIVE\_CAPTION\_INGEST\_ON\_CAPTION\_CEA608\_MAX\_WARN\_MESSAGES**

```
public static final java.lang.String
PROP_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608_MAX_WARN_MESSAGES
```

Timed Text Property "captionLiveIngestOnCaptionCEA608MaxWarnMessages": When ingesting AMF onCaption 708 data events (actually CEA-608), controls the maximum number of warning messages that are logged by the decoder

Type Integer. Default is 25.

Add the property to Application/TimedText/Properties in Application.xml  
Constant value: **captionLiveIngestOnCaptionCEA608MaxWarnMessages**

## **PROP\_LIVE\_CAPTION\_INGEST\_ON\_CAPTION\_CEA608\_TRACK\_NUMBER**

```
public static final java.lang.String
PROP_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608_TRACK_NUMBER
```

Timed Text Property "captionLiveIngestOnCaptionCEA608TrackNumber": When ingesting AMF onCaption 708 data events (actually CEA-608), controls the track number used to add captions as onTextData

Type Integer. Default is 99.

Add the property to Application/TimedText/Properties in Application.xml  
Constant value: **captionLiveIngestOnCaptionCEA608TrackNumber**

## **PROP\_LIVE\_CAPTION\_INGEST\_ON\_CAPTION\_CEA608\_LANGUAGE\_ID**

```
public static final java.lang.String
PROP_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608_LANGUAGE_ID
```

Timed Text Property "captionLiveIngestOnCaptionCEA608LanguageID": When ingesting AMF onCaption 708 data events (actually CEA-608), controls the language ID used to add captions as onTextData

Type String. Default is language code of Java's locale.

Add the property to Application/TimedText/Properties in Application.xml  
Constant value: **captionLiveIngestOnCaptionCEA608LanguageID**

## **PROP\_DEBUG\_LIVE\_CAPTION\_INGEST\_ON\_CAPTION\_CEA608\_VERBOSE**

```
public static final java.lang.String
PROP_DEBUG_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608_VERBOSE
```

Timed Text Property "debugCaptionLiveIngestOnCaptionCEA608Verbose" : Adds verbose logging to AMF onCaption events of type "708" (CEA-608)

Type Boolean. The default value is false.

Add the property to Application/TimedText/Properties in Application.xml  
Constant value: **debugCaptionLiveIngestOnCaptionCEA608Verbose**

## **PROP\_DEBUG\_LIVE\_CAPTION\_INGEST\_ON\_CAPTION\_CEA608\_LOG\_AS\_SCC**

```
public static final java.lang.String
PROP_DEBUG_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608_LOG_AS_SCC
```

(continued from last page)

Timed Text Property "debugCaptionLiveIngestOnCaptionCEA608LogAsSCC" : Adds additional logging of decoding AMF onCaption events of type "708" (CEA-608) such that they are logged in SCC format

Type Boolean. The default value is false.

Add the property to Application/TimedText/Properties in Application.xml  
Constant value: **debugCaptionLiveIngestOnCaptionCEA608LogAsSCC**

## **PROP\_DEBUG\_LIVE\_CAPTION\_INGEST\_ON\_CAPTION\_CEA608\_LOG\_AS\_RAW**

```
public static final java.lang.String
PROP_DEBUG_LIVE_CAPTION_INGEST_ON_CAPTION_CEA608_LOG_AS_RAW
```

Timed Text Property "debugCaptionLiveIngestOnCaptionCEA608LogAsRaw" : Adds additional logging of decoding AMF onCaption events of type "708" (CEA-608) such that they are logged in a raw binary state

Type Boolean. The default value is false.

Add the property to Application/TimedText/Properties in Application.xml  
Constant value: **debugCaptionLiveIngestOnCaptionCEA608LogAsRaw**

## **ON\_TEXT\_DATA**

```
public static final java.lang.String ON_TEXT_DATA
```

Constant value: **onTextData**

## **ON\_CAPTION\_INFO**

```
public static final java.lang.String ON_CAPTION_INFO
```

Constant value: **onCaptionInfo**

## **ON\_CAPTION**

```
public static final java.lang.String ON_CAPTION
```

Constant value: **onCaption**

## **SMIL\_PARAM\_WOWZA\_CAPTION\_INGEST\_TYPE**

```
public static final java.lang.String SMIL_PARAM_WOWZA_CAPTION_INGEST_TYPE
```

Constant value: **wowzacaptioningesttype**

## **SMIL\_PARAM\_IS\_WOWZA\_CAPTION\_STREAM**

```
public static final java.lang.String SMIL_PARAM_IS_WOWZA_CAPTION_STREAM
```

Constant value: **iswowzacaptionstream**

---

Package

**com.wowza.wms.transcoder.model**

## com.wowza.wms.transcoder.model Interface ILiveStreamTranscoderActionNotify

public interface **ILiveStreamTranscoderActionNotify**  
extends

ILiveStreamTranscoderActionNotify: listener interface for live stream transcoder events.

### Method Summary

void	<a href="#"><u>onCalculateSourceAudioBitrate</u></a> (LiveStreamTranscoder liveStreamTranscoder, long bitrate) Called when the bitrate of the source audio stream is calculated
void	<a href="#"><u>onCalculateSourceVideoBitrate</u></a> (LiveStreamTranscoder liveStreamTranscoder, long bitrate) Called when the bitrate of the source video stream is calculated
void	<a href="#"><u>onInitAfterLoadTemplate</u></a> (LiveStreamTranscoder liveStreamTranscoder) Called just after transcoder template is loaded.
void	<a href="#"><u>onInitBeforeLoadTemplate</u></a> (LiveStreamTranscoder liveStreamTranscoder) Called just before transcoder template is loaded
void	<a href="#"><u>onInitStart</u></a> (LiveStreamTranscoder liveStreamTranscoder, String streamName, String transcoderName, <a href="#"><u>IApplicationInstance</u></a> appInstance, LiveStreamTranscoderItem liveStreamTranscoderItem) Call when live stream transcoder is first created
void	<a href="#"><u>onInitStop</u></a> (LiveStreamTranscoder liveStreamTranscoder) At the end of the initialization process
void	<a href="#"><u>onRegisterStreamNameGroup</u></a> (LiveStreamTranscoder liveStreamTranscoder, TranscoderStreamNameGroup streamNameGroup) Called after a stream name group is resolved and registered with MediaStreamMap
void	<a href="#"><u>onResetStream</u></a> (LiveStreamTranscoder liveStreamTranscoder) Called when the stream feeding the live stream transcoder switches.
void	<a href="#"><u>onSessionAudioDecodeCodecInfo</u></a> (LiveStreamTranscoder liveStreamTranscoder, com.wowza.wms.media.model.MediaCodecInfoAudio codecInfoAudio) Called when audio decoding information is available.
void	<a href="#"><u>onSessionAudioEncodeCodecInfo</u></a> (LiveStreamTranscoder liveStreamTranscoder, TranscoderSessionAudioEncode sessionAudioEncode, com.wowza.wms.media.model.MediaCodecInfoAudio codecInfoAudio) Called when audio encoding information is available.
void	<a href="#"><u>onSessionAudioEncodeCreate</u></a> (LiveStreamTranscoder liveStreamTranscoder, TranscoderSessionAudioEncode sessionAudioEncode) Called when audio encoder session is created.
void	<a href="#"><u>onSessionAudioEncodeInit</u></a> (LiveStreamTranscoder liveStreamTranscoder, TranscoderSessionAudioEncode sessionAudioEncode) Called after audio session is initialized.

void	<a href="#">onSessionAudioEncodeSetup</a> (LiveStreamTranscoder liveStreamTranscoder, TranscoderSessionAudioEncode sessionAudioEncode) Called after native audio encoder is created and initialized.
void	<a href="#">onSessionDataEncodeCreate</a> (LiveStreamTranscoder liveStreamTranscoder, TranscoderSessionDataEncode sessionDataEncode) Called when data encoder session is created.
void	<a href="#">onSessionDataEncodeInit</a> (LiveStreamTranscoder liveStreamTranscoder, TranscoderSessionDataEncode sessionDataEncode) Called after data session is initialized.
void	<a href="#">onSessionDestinationCreate</a> (LiveStreamTranscoder liveStreamTranscoder, TranscoderSessionDestination sessionDestination) Called when transcoding destination is created
void	<a href="#">onSessionVideoDecodeCodecInfo</a> (LiveStreamTranscoder liveStreamTranscoder, com.wowza.wms.media.model.MediaCodecInfoVideo codecInfoVideo) Called when video decoding information is available.
void	<a href="#">onSessionVideoEncodeCodecInfo</a> (LiveStreamTranscoder liveStreamTranscoder, TranscoderSessionVideoEncode sessionVideoEncode, com.wowza.wms.media.model.MediaCodecInfoVideo codecInfoVideo) Called when video encoding information is available.
void	<a href="#">onSessionVideoEncodeCreate</a> (LiveStreamTranscoder liveStreamTranscoder, TranscoderSessionVideoEncode sessionVideoEncode) Called when video encoder session is created.
void	<a href="#">onSessionVideoEncodeInit</a> (LiveStreamTranscoder liveStreamTranscoder, TranscoderSessionVideoEncode sessionVideoEncode) Called after video session is initialized.
void	<a href="#">onSessionVideoEncodeSetup</a> (LiveStreamTranscoder liveStreamTranscoder, TranscoderSessionVideoEncode sessionVideoEncode) Called after native video encoder is created and initialized.
void	<a href="#">onShutdownStart</a> (LiveStreamTranscoder liveStreamTranscoder) Called when the live stream transcoder starts to shutdown.
void	<a href="#">onShutdownStop</a> (LiveStreamTranscoder liveStreamTranscoder) Called when the live stream transcoder is shutdown.
void	<a href="#">onUnregisterStreamNameGroup</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="#"><u>getAndClearPendingFrameGrabs()</u></a> Returns a list of all grab frame requests that are pending and clears the pending list.
void	<a href="#"><u>grabFrame( ITranscoderFrameGrabResult grabResult )</u></a> Call to grab a frame.
void	<a href="#"><u>grabFrame( ITranscoderFrameGrabResult grabResult, int width, int height )</u></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="#"><u>onGrabFrame</u></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 ITranscoderOverlayImages

public interface ITranscoderOverlayImages  
extends

### Method Summary

void	<a href="#">addOverlayImage(int index, TranscoderVideoOverlayFrame overlay)</a>
------	---

java.util.Map	<a href="#">getOverlayImages()</a>
---------------	------------------------------------

### Methods

#### **addOverlayImage**

```
public void addOverlayImage(int index,  
                           TranscoderVideoOverlayFrame overlay)
```

#### **getOverlayImages**

```
public java.util.Map getOverlayImages()
```

## com.wowza.wms.transcoder.model Interface ITranscoderOverlayProvider

public interface ITranscoderOverlayProvider  
extends

ITranscoderOverlayProvider: interface to add and remove video overlays.

### Method Summary

void	<a href="#"><u>addOverlay(int index, TranscoderVideoOverlayFrame overlay)</u></a> Add an overlay to a video stream.
void	<a href="#"><u>clearOverlay(int index)</u></a> Clear video overlay
java.util.Map	<a href="#"><u>getAndClearPendingOverlays()</u></a> Get a list of pending overlay requests and clear the pending overlay queue.
boolean	<a href="#"><u>isOverlayAvailable()</u></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="#">onAfterDecodeFrame</a> (TranscoderSessionVideo sessionVideo, TranscoderStreamSourceVideo sourceVideo, long frameCount)
void	<a href="#">onAfterScaleFrame</a> (TranscoderSessionVideo sessionVideo, TranscoderStreamSourceVideo sourceVideo, long frameCount)
void	<a href="#">onBeforeDecodeFrame</a> (TranscoderSessionVideo sessionVideo, TranscoderStreamSourceVideo sourceVideo, long frameCount)
void	<a href="#">onBeforeScaleFrame</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

void	<a href="#">destroy()</a>
String	<a href="#">getContextStr()</a>
long	<a href="#">getFrameCount()</a>
boolean	<a href="#">isEncoderInUse()</a>
boolean	<a href="#">isRunning()</a> Returns true if transcoder worker is running (internal use)
void	<a href="#">setFrameCount(long frameCount)</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()**

(continued from last page)

---

### **getContextStr**

```
public String getContextStr()
```

---

### **destroy**

```
public void destroy()
```

## com.wowza.wms.transcoder.model Interface ITranscoderWorkerSorterSender

---

public interface ITranscoderWorkerSorterSender  
extends

For internal use.

### Method Summary

void	<a href="#"><u>addSorterPacket</u></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(String rangeStr)</a> Decode RTP range header, Internal use.
static <a href="#">RTPStream</a>	<a href="#">decodeStreamInfo(IApplicationInstance appInstance, RTPContext context, String streamId, String streamInfo)</a> Decode SDP info and create RTP stream
static <a href="#">RTPStream</a>	<a href="#">decodeStreamInfo(IApplicationInstance appInstance, RTPStreamContext rtpStreamContext, RTPContext context, String streamId, String streamInfo)</a> Decode SDP info and create RTP stream
static <a href="#">RTPStream</a>	<a href="#">decodeStreamInfo(RTPContext context, String streamId, String streamInfo)</a> Decode SDP info and create RTP stream
static byte[]	<a href="#">formatH264CodecConfig(byte[] sps, java.util.List ppss, byte[] profileLevel)</a> Format codec config info, Internal use.
static byte[]	<a href="#">formatH264CodecConfigPacket(byte[] sps, java.util.List ppss, byte[] profileLevel)</a> Format codec config info, Internal use.
static boolean	<a href="#">loadConfigFile(RTPContext rtpContext, String fileURL)</a> Load config file, Internal use.
static <a href="#">RTPPushPublishSession</a>	<a href="#">startRTPPull(IApplicationInstance appInstance, String streamName, boolean streamPacketizer, String ipAddress, int streamPort)</a> Start pushing an RTP stream
static <a href="#">RTPPushPublishSession</a>	<a href="#">startRTPPull(IApplicationInstance appInstance, String streamName, boolean streamPacketizer, String ipAddress, int streamPort, boolean isRTPWrapped)</a> Start pushing an RTP stream
static <a href="#">RTPPushPublishSession</a>	<a href="#">startRTPPull(IApplicationInstance appInstance, String streamName, boolean streamPacketizer, String ipAddress, int audioPort, int videoPort)</a> Start pushing an RTP stream

<code>static RTPPushPublishSession</code>	<code>startRTPPull(IApplicationInstance appInstance, String streamName, boolean streamPacketizer, String ipAddress, int audioPort, int videoPort, boolean isRTPWrapped)</code> Start pushing an RTP stream
<code>static RTPPushPublishSession</code>	<code>startRTPPull(IApplicationInstance appInstance, String streamName, RTPDestination rtpDestination)</code> Start pushing an RTP stream
<code>static void</code>	<code>stopRTPPull(RTPPushPublishSession rtpPushPublishSession)</code> Stop pushing an RTP stream
<code>static String</code>	<code>updateSDPDestination(RTPDestination rtpDestination, String sdpData)</code> Update SDP data information with RTP destination information
<code>static void</code>	<code>writeCodecConfig(RTPTrack rtpTrack, int codecId, long adjTimecode, byte[] codecConfig)</code>
<code>static void</code>	<code>writeCodecConfig(RTPTrack rtpTrack, long adjTimecode, byte[] codecConfig)</code> Write codec config information, Internal use.
<code>static void</code>	<code>writeCodecConfig(RTPTrack rtpTrack, long adjTimecode, byte[] codecConfig, int codec)</code>

**Methods inherited from class java.lang.Object**

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait`

## Constructors

### RTPUtils

```
public RTPUtils()
```

## Methods

### updateSDPDestination

```
public static String updateSDPDestination(RTPDestination rtpDestination,
                                         String sdpData)
```

Update SDP data information with RTP destination information

**Parameters:**

`rtpDestination` - RTP destination  
`sdpData` - SDP data

**Returns:**

modified SDP data

(continued from last page)

## decodeRangeHeader

```
public static double[] decodeRangeHeader(String rangeStr)
```

Decode RTP range header, Internal use.

**Parameters:**

rangeStr

**Returns:**

range values

---

## stopRTPPull

```
public static void stopRTPPull(RTPPushPublishSession rtpPushPublishSession)
```

Stop pushing an RTP stream

**Parameters:**

rtpPushPublishSession - RTP push session

---

## startRTPPull

```
public static RTPPushPublishSession startRTPPull(IAplicationInstance 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(IAplicationInstance appInstance,  
String streamName,  
boolean streamPacketizer,  
String ipAddress,  
int streamPort)
```

Start pushing an RTP stream

**Parameters:**

appInstance - application instance  
streamName - stream name  
streamPacketizer - stream packetizer  
ipAddress - IP address  
streamPort - stream port

(continued from last page)

**Returns:**

RTP push session

---

**startRTPPull**

```
public static RTPPushPublishSession startRTPPull(IApplicationInstance appInstance,  
    String streamName,  
    boolean streamPacketizer,  
    String ipAddress,  
    int audioPort,  
    int videoPort)
```

Start pushing an RTP stream

**Parameters:**

appInstance - application instance  
streamName - stream name  
streamPacketizer - stream packetizer  
ipAddress - IP address  
audioPort - audio port  
videoPort - video port

**Returns:**

RTP push session

---

**startRTPPull**

```
public static RTPPushPublishSession startRTPPull(IApplicationInstance appInstance,  
    String streamName,  
    boolean streamPacketizer,  
    String ipAddress,  
    int audioPort,  
    int videoPort,  
    boolean isRTPWrapped)
```

Start pushing an RTP stream

**Parameters:**

appInstance - application instance  
streamName - stream name  
streamPacketizer - stream packetizer  
ipAddress - IP address  
audioPort - audio port  
videoPort - video port  
isRTPWrapped - is RTP wrapped

**Returns:**

RTP push session

---

**startRTPPull**

```
public static RTPPushPublishSession startRTPPull(IApplicationInstance appInstance,  
    String streamName,  
    RTPDestination rtpDestination)
```

Start pushing an RTP stream

**Parameters:**

appInstance - application instance  
streamName - stream name  
rtpDestination - RTP destination

(continued from last page)

**Returns:**

RTP push session

**writeCodecConfig**

```
public static void writeCodecConfig(RTPTrack rtpTrack,
    int codecId,
    long adjTimecode,
    byte[] codecConfig)
```

**writeCodecConfig**

```
public static void writeCodecConfig(RTPTrack rtpTrack,
    long adjTimecode,
    byte[] codecConfig)
```

Write codec config information, Internal use.

**Parameters:**

rtpTrack - RTP track  
adjTimecode - timecode (milliseconds)  
codecConfig - codec config

**writeCodecConfig**

```
public static void writeCodecConfig(RTPTrack rtpTrack,
    long adjTimecode,
    byte[] codecConfig,
    int codec)
```

**loadConfigFile**

```
public static boolean loadConfigFile(RTPContext rtpContext,
    String fileURL)
```

Load config file, Internal use.

**Parameters:**

rtpContext  
fileURL

**decodeStreamInfo**

```
public static RTPStream decodeStreamInfo(RTPContext context,
    String streamId,
    String streamInfo)
```

Decode SDP info and create RTP stream

**Parameters:**

context - RTP context  
streamId - stream id  
streamInfo - SDP data

**Returns:**

(continued from last page)

RTP stream

---

## decodeStreamInfo

```
public static RTPStream decodeStreamInfo(IAplicationInstance appInstance,  
    RTPContext context,  
    String streamId,  
    String streamInfo)
```

Decode SDP info and create RTP stream

**Parameters:**

appInstance - app instance  
context - rtp context  
streamId - stream id  
streamInfo - stream info (SDP data)

**Returns:**

rtp stream

---

## decodeStreamInfo

```
public static RTPStream decodeStreamInfo(IAplicationInstance appInstance,  
    RTPStreamContext rtpStreamContext,  
    RTPContext context,  
    String streamId,  
    String streamInfo)
```

Decode SDP info and create RTP stream

**Parameters:**

appInstance - app instance  
rtpStreamContext - rtp stream context  
context - rtp context  
streamId - stream id  
streamInfo - stream info (SDP data)

**Returns:**

rtp stream

---

## formatH264CodecConfigPacket

```
public static byte[] formatH264CodecConfigPacket(byte[] sps,  
    java.util.List ppss,  
    byte[] profileLevel)
```

Format codec config info, Internal use.

**Parameters:**

sps  
ppss  
profileLevel

**Returns:**

bytes

---

(continued from last page)

## formatH264CodecConfig

```
public static byte[] formatH264CodecConfig(byte[] sps,  
                                         java.util.List ppss,  
                                         byte[] profileLevel)
```

Format codec config info, Internal use.

**Parameters:**

- sps
- ppss
- profileLevel

**Returns:**

- bytes

## com.wowza.wms.util Class StreamUtils

```
java.lang.Object
+-com.wowza.wms.util.StreamUtils
```

```
public class StreamUtils
extends Object
```

### Constructor Summary

public	<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="#">IAplicationInstance</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/second.
static double	<a href="#">getStreamLength</a> ( <a href="#">IAplicationInstance</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 boolean	<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/second. This will work correctly with all RandomAccessReaders and caching technology.

**Parameters:**

stream - stream

**Returns:**

bitrate in bits/second

#### getStreamLength

```
public static double getStreamLength(IApplicationInstance appInstance,  
String streamName)
```

Get the duration of a media file in seconds. This will work correctly with all RandomAccessReaders and caching technology

**Parameters:**

appInstance - application instance

streamName - stream name

**Returns:**

duration in seconds

#### getStreamBitrate

```
public static double getStreamBitrate(IApplicationInstance appInstance,  
String streamName)
```

(continued from last page)

## directOutput

```
public static int directOutput(byte[] dataBuffer,
    int[] headerValues,
    boolean isAbsTimecode,
    int src,
    java.io.OutputStream out,
    AMFObj wmsObj,
    byte[] workBuffer,
    int chunkSize)
```

Direct output, Internal use.

### Parameters:

- dataBuffer
- headerValues
- isAbsTimecode
- src
- out
- wmsObj
- workBuffer
- chunkSize

### Returns:

c

---

## packetOutput

```
public static int packetOutput(java.io.OutputStream out,
    IMediaStream stream,
    AMFPacket packet,
    long timecode,
    AMFObj wmsObj,
    byte[] workBuffer,
    int chunkSize)
```

Packet output. Internal use.

### Parameters:

- out
- stream
- packet
- timecode
- wmsObj
- workBuffer
- chunkSize

### Returns:

bytes

---

## packetOutput

```
public static int packetOutput(java.io.OutputStream out,
    IMediaStream stream,
    AMFPacket packet,
    long timecode,
    AMFObj wmsObj,
    byte[] workBuffer,
    int chunkSize,
    boolean referenceWrite)
```

Packet output. Internal use.

(continued from last page)

**Parameters:**

out  
stream  
packet  
timecode  
wmsObj  
workBuffer  
chunkSize  
referenceWrite

**Returns:**

bytes

**loadConfigFile**

```
public static boolean loadConfigFile(StreamList streamDefs,  
String fileURL)
```

Load Streams.xml, Internal use.

**Parameters:**

streamDefs  
fileURL

---

Package

**com.wowza.wms.util.crontab**

## com.wowza.wms.util.crontab

### Class CrontabEvent

```
java.lang.Object
+-com.wowza.wms.util.crontab.CrontabEvent
```

All Implemented Interfaces:  
Comparable

```
public class CrontabEvent
extends Object
implements Comparable
```

Holds an event's schedule and is responsible for calculating event activation.

#### Field Summary

public static final	<a href="#">DAY</a> Value: 2
public static	<a href="#">DT_FORMAT</a>
public static final	<a href="#">HOUR</a> Value: 1
public static final	<a href="#">MINUTE</a> Value: 0
public static final	<a href="#">MONTH</a> Value: 3
public static final	<a href="#">WEEKDAY</a> Value: 5
public static final	<a href="#">YEAR</a> Value: 4

#### Constructor Summary

public	<a href="#">CrontabEvent()</a>
--------	--------------------------------

#### Method Summary

int	<a href="#">compareTo(CrontabEvent o)</a> Allows CronEvents to be sorted via Java Sort Class.
void	<a href="#">dump()</a> dumps this event's information to the console

String	<a href="#">getError()</a>
int	<a href="#">getErrorField()</a>
<a href="#">ICrontabEventHandler</a>	<a href="#">getEventHandler()</a>
String	<a href="#">getEventStr()</a>
CrontabField	<a href="#">getField(int whichField)</a>
org.joda.time.MutableDateTime	<a href="#">getLastRun()</a>
org.joda.time.MutableDateTime	<a href="#">getNextRun()</a>
String	<a href="#">getTarget()</a>
boolean	<a href="#">isDebugOn()</a>
boolean	<a href="#">isExpired()</a> Indicates whether or not all execution dates for this event occur in the past
boolean	<a href="#">isMatch(org.joda.time.MutableDateTime date)</a> Determines if this event will fire on the date passed in
void	<a href="#">setDebug(boolean debug)</a> turns on/off debug logging
void	<a href="#">setError(String error)</a>
void	<a href="#">setErrorField(int errorField)</a>
void	<a href="#">setEventHandler(<a href="#">ICrontabEventHandler</a> handler)</a>
void	<a href="#">setEventStr(String entry)</a>
void	<a href="#">setField(int whichField, CrontabField field)</a>
void	<a href="#">setLastRun()</a> Sets the last run time to now
void	<a href="#">setLastRun(org.joda.time.MutableDateTime date)</a>
void	<a href="#">setTarget(String target)</a>
boolean	<a href="#">start()</a> Starts this events timer
void	<a href="#">stop()</a> stops this events timer

Methods inherited from class `java.lang.Object`

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait,  
wait
```

Methods inherited from interface java.lang.Comparable

```
compareTo
```

## Fields

### DT\_FORMAT

```
public static org.joda.time.format.DateTimeFormatter DT_FORMAT
```

---

### MINUTE

```
public static final int MINUTE
```

Constant value: 0

---

### HOUR

```
public static final int HOUR
```

Constant value: 1

---

### DAY

```
public static final int DAY
```

Constant value: 2

---

### MONTH

```
public static final int MONTH
```

Constant value: 3

---

### YEAR

```
public static final int YEAR
```

Constant value: 4

---

### WEEKDAY

```
public static final int WEEKDAY
```

Constant value: 5

## Constructors

### CrontabEvent

```
public CrontabEvent()
```

## Methods

### start

```
public boolean start()
```

Starts this events timer

**Returns:**

true - if timer is started, else false

---

### stop

```
public void stop()
```

stops this events timer

---

### isMatch

```
public boolean isMatch(org.joda.time.MutableDateTime date)
```

Determines if this event will fire on the date passed in

**Parameters:**

date

**Returns:**

boolean

---

### setField

```
public void setField(int whichField,  
                     CrontabField field)
```

---

### getField

```
public CrontabField getField(int whichField)
```

---

### setTarget

```
public void setTarget(String target)
```

---

(continued from last page)

**getTarget**

```
public String getTarget()
```

---

**setEventStr**

```
public void setEventStr(String entry)
```

---

**getEventStr**

```
public String getEventstr()
```

---

**getNextRun**

```
public org.joda.time.MutableDateTime getNextRun()
```

---

**getLastRun**

```
public org.joda.time.MutableDateTime getLastRun()
```

---

**setLastRun**

```
public void setLastRun(org.joda.time.MutableDateTime date)
```

---

**getError**

```
public String getError()
```

---

**setError**

```
public void setError(String error)
```

---

**getErrorField**

```
public int getErrorField()
```

---

**setErrorField**

```
public void setErrorField(int errorField)
```

---

(continued from last page)

---

**setLastRun**

```
public void setLastRun()
```

Sets the last run time to now

---

**setDebug**

```
public void setDebug(boolean debug)
```

turns on/off debug logging

**Parameters:**

debug -- true/false

---

**isDebugOn**

```
public boolean isDebugOn()
```

---

**setEventHandler**

```
public void setEventHandler(ICrontabEventHandler handler)
```

---

**getEventHandler**

```
public ICrontabEventHandler getEventHandler()
```

---

**isExpired**

```
public boolean isExpired()
```

Indicates whether or not all execution dates for this event occur in the past

**Returns:**

true/false

---

**compareTo**

```
public int compareTo(CrontabEvent o)
```

Allows CronEvents to be sorted via Java Sort Class. Compares nextRun dates of two CronEvents where earlier events are considered "less" than later events, (e.g. today < tomorrow) and nulls are "greater" than events with dates (e.g. tomorrow < null) This provides the logic to sort events by soonest to latest with nulls at the end

**Parameters:**

o -- CronEvent to compare this event to

**Returns:**

-1,0,1 - if equal, if this event is greater than

---

(continued from last page)

## **dump**

`public void dump()`

dumps this event's information to the console

## com.wowza.wms.util.crontab Interface ICrontabEventHandler

public interface **ICrontabEventHandler**  
extends

EventHandler functions are called from a Java timer task context and therefore should execute very quickly so as not to delay execution of subsequent timers. If there is heavy processing which needs to take place, the implementation for these functions should spawn a thread to do it.

### Method Summary

void	<a href="#">onCronEvent(CrontabEvent event)</a>
------	---

### Methods

#### onCronEvent

public void [onCronEvent\(CrontabEvent event\)](#)

---

**Package**

**com.wowza.wms.vhost**

## com.wowza.wms.vhost

### Class HostPort

```
java.lang.Object
+-com.wowza.wms.vhost.HostPort
```

---

```
public class HostPort
extends Object
```

HostPort: data object that describes a socket connection. The address can be defined by ipAddress or by domainName. It can also contain a reference to an SSLFactory class that can be used to create a secure connection to the server.

#### Field Summary

<code>public static final</code>	<a href="#">TYPE_ADMIN</a>
	Value: <b>Admin</b>
<code>public static final</code>	<a href="#">TYPE_STREAMING</a>
	Value: <b>Streaming</b>

#### Constructor Summary

<code>public</code>	<a href="#">HostPort()</a> Create an empty HostPort object
---------------------	---

#### Method Summary

<code>void</code>	<a href="#">addHttpProvider(<a href="#">IHTTPProvider</a> httpProvider)</a>
<code>void</code>	<a href="#">addHttpProvider(<a href="#">IHTTPProvider2</a> httpProvider)</a>
<code>void</code>	<a href="#">addHttpProvider(int index, <a href="#">IHTTPProvider2</a> httpProvider)</a>
<code>void</code>	<a href="#">addHTTPStreamerAdapterID(String ID)</a>
<code>void</code>	<a href="#">configureSocketAcceptor(org.apache.mina.transport.socket.nio.SocketAcceptorConfig socketConfig)</a> Configure a socketAcceptoport
<code>java.net.InetAddress</code>	<a href="#">getAddress()</a> Get the ipAddress as an InetAddress object
<code>String</code>	<a href="#">getAddressRawStr()</a>
<code>String</code>	<a href="#">getAddressStr()</a> Get a String representation of the address
<code>HostPortConfig</code>	<a href="#">getConfiguration()</a> Get the socket configuration

String	<a href="#">getHTTPIdent2Response()</a>
String	<a href="#">getHTTPIdentResponse()</a>
java.util.List	<a href="#">getHttpProviders()</a>
java.util.List	<a href="#">getHTTPStreamerAdapterIDs()</a>
String	<a href="#">getName()</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
String	<a href="#">getTypeStr()</a>
boolean	<a href="#">isAdmin()</a>
boolean	<a href="#">isStreaming()</a>
boolean	<a href="#">isSuspended()</a>
void	<a href="#">setDomainName(String domainName)</a> Set domainName.
void	<a href="#">setHTTPIdent2Response(String httpIdent2Response)</a>
void	<a href="#">setHTTPIdentResponse(String httpIdentResponse)</a>
void	<a href="#">setIpAddress(String ipAddress)</a> Set ipAddress for object.
void	<a href="#">setName(String name)</a>
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>
void	<a href="#">setTypeStr(String typeStr)</a>
int	<a href="#">size(int index, IHTTPPProvider2 httpProvider)</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

## Fields

**TYPE\_ADMIN**

```
public static final java.lang.String TYPE_ADMIN
```

Constant value: **Admin**

**TYPE\_STREAMING**

```
public static final java.lang.String TYPE_STREAMING
```

Constant value: **Streaming**

## Constructors

**HostPort**

```
public HostPort()
```

Create an empty HostPort object

## Methods

**setDomainName**

```
public void setDomainName(String domainName)
```

Set domainName. A domainName of \* defines a local connection.

**Parameters:**

domainName - domainName like www.mycompany.com or \* for local

(continued from last page)

## getAddress

```
public java.net.InetAddress getAddress()
```

Get the ipAddress as an InetAddress object

**Returns:**

ipAddress as an InetAddress object. At this point the domainName (if specified) will be resolved.

---

## getAddressStr

```
public String getAddressStr()
```

Get a String representation of the address

**Returns:**

String representation of the address/domainName. If \* it will return [any]

---

## getAddressRawStr

```
public String getAddressRawStr()
```

---

## setIpAddress

```
public void setIpAddress(String ipAddress)
```

Set ipAddress for object. A ipAddress of \* defines a local connection.

**Parameters:**

ipAddress - ipAddress like 127.0.0.1 or \* for local

---

## getPort

```
public int getPort()
```

Get port

**Returns:**

port

---

## setPort

```
public void setPort(int port)
```

Set port

**Parameters:**

port - port

---

## getSslFactoryClass

```
public String getSslFactoryClass()
```

Get full class name or SSLFactory class

**Returns:**

(continued from last page)

full class name or SSLFactory or "" if not specified

---

## setSslFactoryClass

```
public void setSslFactoryClass(String sslFactoryClass)
```

Set full class name of SSLFactory class

**Parameters:**

sslFactoryClass - full class name or SSLFactory or "" if not specified

---

## getProcessorCount

```
public int getProcessorCount()
```

Get the number of threads to use to service this incoming port

**Returns:**

number of processor threads

---

## setProcessorCount

```
public void setProcessorCount(int processorCount)
```

Set the number of threads to use to service this incoming port

**Parameters:**

processorCount - number of processor threads

---

## toString

```
public String toString(boolean mBeanSafe)
```

Return object as formatted string

**Parameters:**

mBeanSafe - make the name safe for JMX management interface

**Returns:**

formmatted string

---

## toString

```
public String tostring()
```

Return object as formatted string

**Returns:**

formmatted string

---

## getConfiguration

```
public HostPortConfig getConfiguration()
```

Get the socket configuration

**Returns:**

socket configuration

---

**configureSocketAcceptor**

```
public void  
configureSocketAcceptor(org.apache.mina.transport.socket.nio.SocketAcceptorConfig  
socketConfig)
```

Configure a socketAccepport

**Parameters:**

socketConfig - socket acceptor

---

**getHttpProviders**

```
public java.util.List getHttpProviders()
```

---

**addHttpProvider**

```
public void addHttpProvider(IHTTPProvider httpProvider)
```

---

**addHttpProvider**

```
public void addHttpProvider(IHTTPProvider2 httpProvider)
```

---

**addHttpProvider**

```
public void addHttpProvider(int index,  
IHTTPProvider2 httpProvider)
```

---

**size**

```
public int size(int index,  
IHTTPProvider2 httpProvider)
```

---

**isSuspended**

```
public boolean isSuspended()
```

---

**setSuspended**

```
public void setSuspended(boolean isSuspended)
```

---

(continued from last page)

**getHTTPStreamerAdapterIDs**

```
public java.util.List getHTTPStreamerAdapterIDs()
```

---

**addHTTPStreamerAdapterID**

```
public void addHTTPStreamerAdapterID(String ID)
```

---

**getSSLConfig**

```
public HostPortSSLConfig getSSLConfig()
```

---

**setSSLConfig**

```
public void setSSLConfig(HostPortSSLConfig sslConfig)
```

---

**getHTTPIdentResponse**

```
public String getHTTPIdentResponse()
```

---

**setHTTPIdentResponse**

```
public void setHTTPIdentResponse(String httpIdentResponse)
```

---

**getHTTPIdent2Response**

```
public String getHTTPIdent2Response()
```

---

**setHTTPIdent2Response**

```
public void setHTTPIdent2Response(String httpIdent2Response)
```

---

**getTypeStr**

```
public String getTypeStr()
```

---

**setTypeStr**

```
public void setTypeStr(String typeStr)
```

(continued from last page)

---

**getName**

```
public String getName()
```

---

**setName**

```
public void setName(String name)
```

---

**isAdmin**

```
public boolean isAdmin()
```

---

**isStreaming**

```
public boolean isStreaming()
```

## com.wowza.wms.vhost

### Class HostPortList

```
java.lang.Object
+-com.wowza.wms.vhost.HostPortList
```

---

```
public class HostPortList
extends Object
```

HostPortList: data object that contains a collection of HostPort objects.

#### Constructor Summary

public	<a href="#">HostPortList()</a>
	Create empty HostPortList

#### Method Summary

void	<a href="#">add(HostPort 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,
```

#### 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="#"><u>onAcceptorCreate</u></a> ( <a href="#"><u>HostPort</u></a> hostPort, java.util.Map acceptorMap) Triggered when a new acceptor is created
void	<a href="#"><u>onAcceptorDestroy</u></a> ( <a href="#"><u>HostPort</u></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_AC3</a>  Value: <b>1</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="#"><u>CODEC_AUDIO_PCM_LE</u></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_H265</u></a>
	Value: <b>12</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="#"><u>CODEC_VIDEO_VP6A</u></a> Value: <b>5</b>
public static final	<a href="#"><u>CODEC_VIDEO_VP8</u></a> Value: <b>8</b>
public static final	<a href="#"><u>CONTENTTYPE_ACKBANDWIDTH</u></a> AMF Content type: set acknowledge bandwidth size Value: <b>5</b>
public static final	<a href="#"><u>CONTENTTYPE_AUDIO</u></a> AMF Content type: audio packet Value: <b>8</b>
public static final	<a href="#"><u>CONTENTTYPE_BUFFERSIZE</u></a> AMF Content type: set buffer size Value: <b>4</b>
public static final	<a href="#"><u>CONTENTTYPE_DATA</u></a> AMF Content type: data packet Value: <b>18</b>
public static final	<a href="#"><u>CONTENTTYPE_DATA0</u></a> AMF Content type: data packet (AMF0) Value: <b>18</b>
public static final	<a href="#"><u>CONTENTTYPE_DATA3</u></a> AMF Content type: data packet (AMF3) Value: <b>15</b>
public static final	<a href="#"><u>CONTENTTYPE_FUNCTION</u></a> AMF Content type: function data (AMF0) Value: <b>20</b>
public static final	<a href="#"><u>CONTENTTYPE_FUNCTION0</u></a> AMF Content type: function data (AMF0) Value: <b>20</b>
public static final	<a href="#"><u>CONTENTTYPE_FUNCTION3</u></a> AMF Content type: function data (AMF3) Value: <b>17</b>
public static final	<a href="#"><u>CONTENTTYPE_MEDIACHUNK</u></a> AMF Content type: media chunk Value: <b>22</b>
public static final	<a href="#"><u>CONTENTTYPE_PLAYCALLBACK</u></a> AMF Content type: play callback Value: <b>127</b>

public static final	<a href="#"><u>CONTENTTYPE_SETBANDWIDTH</u></a> AMF Content type: set bandwidth size Value: <b>6</b>
public static final	<a href="#"><u>CONTENTTYPE_SETCUNKSIZE</u></a> AMF Content type: set packet chunk size Value: <b>1</b>
public static final	<a href="#"><u>CONTENTTYPE_SHAREDOBJECTS</u></a> AMF Content type: shared object packet (AMF0) Value: <b>19</b>
public static final	<a href="#"><u>CONTENTTYPE_SHAREDOBJECTS0</u></a> AMF Content type: shared object packet (AMF0) Value: <b>19</b>
public static final	<a href="#"><u>CONTENTTYPE_SHAREDOBJECTS3</u></a> AMF Content type: shared object packet (AMF3) Value: <b>16</b>
public static final	<a href="#"><u>CONTENTTYPE_UNKNOWN</u></a> AMF Content type: unknown Value: <b>0</b>
public static final	<a href="#"><u>CONTENTTYPE_VIDEO</u></a> AMF Content type: video packet Value: <b>9</b>
public static final	<a href="#"><u>CONTENTTYPE_WATCHDOG</u></a> AMF Content type: watch dog Value: <b>3</b>
public static final	<a href="#"><u>COUNTER_HTTPCUPERTINO</u></a>  Value: <b>2</b>
public static final	<a href="#"><u>COUNTER_HTTPDVRCHUNKS</u></a>  Value: <b>7</b>
public static final	<a href="#"><u>COUNTER_HTTPMPPEGDASH</u></a>  Value: <b>6</b>
public static final	<a href="#"><u>COUNTER_HTTPSANJOSE</u></a>  Value: <b>4</b>
public static final	<a href="#"><u>COUNTER_HTTPSMOOTH</u></a>  Value: <b>3</b>
public static final	<a href="#"><u>COUNTER_HTTPWEBM</u></a>  Value: <b>5</b>
public static final	<a href="#"><u>COUNTER_RTMP</u></a>  Value: <b>0</b>

public static final	<a href="#"><u>COUNTER_RTP</u></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="#">LICENSECOUNTER_TRANSCODE_DECODECOUNTAUDIO</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="#">LICENSECOUNTERS_DRM</a>
public static final	<a href="#">VHOST_DEFAULT</a> Value: <b>_defaultVHost</b>

## Method Summary

void	<a href="#">addAcceptorListener(IAcceptorNotify acceptorListener)</a> Add acceptor listener.
void	<a href="#">addApplicationListener(IApplicationNotify applicationListener)</a> Add application listener.
void	<a href="#">addIdleWorkerListener(IIIdleWorkerNotify idleWorkerListener)</a> Add idleWorker listener.
void	<a href="#">addStartupStream(StartupStream startupStream)</a> Add a stream to the list of streams to start and virtual host startup
boolean	<a href="#">applicationExists(String name)</a> Return true if an application folder exists for this application name

void	<a href="#"><code>closeHostPort(HostPort hostPort, boolean isSuspend)</code></a> Close an individual HostPort
<a href="#"><u>IApplication</u></a>	<a href="#"><code>getApplication(String applicationName)</code></a> Get application by name.
java.util.List	<a href="#"><code>getApplicationFolderNames()</code></a> Get a list of application folder names
edu.emory.mathcs.backport.java.util.concurrent.locks.WMSReadWriteLock	<a href="#"><code>getApplicationLock()</code></a> Get the object used in synchronized statements to lock and application.
java.util.List	<a href="#"><code>getApplicationNames()</code></a> Get a list of application names
int	<a href="#"><code>getApplicationTimeout()</code></a> Get application time out (milliseconds).
AuthenticationList	<a href="#"><code>getAuthenticationList()</code></a> Get the list of available authentication methods
CEACaptionConversionContext	<a href="#"><code>getCEACaptionConversionContext()</code></a> Get CEACaptionConversionContextMP3 tag map attached to vHost (CEACaptionConversion.xml).
<a href="#"><u>IClient</u></a>	<a href="#"><code>getClient(int clientId)</code></a> Get client by client id.
<a href="#"><u>IClient</u></a>	<a href="#"><code>getClient(int clientId, boolean create)</code></a> Get client by client id and create if does not exist.
int	<a href="#"><code>getClientCount()</code></a> Get number of clients connected to this vHost.
int	<a href="#"><code>getClientIdleFrequency()</code></a> Get default client idle frequency (milliseconds)
int	<a href="#"><code>getClientTimeout()</code></a> Get client timeout.
<a href="#"><u>ConnectionCounter</u></a>	<a href="#"><code>getConnectionCounter()</code></a> Get vHost connection counter.
ConnectionCounterSimple	<a href="#"><code>getConnectionCounter(int counterIndex)</code></a> Get vHost connection counter for a specific technology (see IVHost.COUNTER_*)
int	<a href="#"><code>getConnectionLimit()</code></a> Get vHost connection limit.
int	<a href="#"><code>getCoreHandlerPoolSize()</code></a> Get the handler core thread pool size.
int	<a href="#"><code>getCoreTransportPoolSize()</code></a> Get the transport core thread pool size.
String	<a href="#"><code>getDateStarted()</code></a> Get date and time the server was started.

DvrRecorderList	<a href="#">getDvrRecorderList()</a> Get the DvrRecorderList
DvrStoreList	<a href="#">getDvrStoreList()</a> Get the list of DVR Stores
java.util.Properties	<a href="#">getDynamicLogProperties()</a> Get the dynamic log properties defined at the vhost level in conf/log4j.properties
int	<a href="#">getFileIOPoolSize()</a> Get the default file io pool size.
ThreadPool	<a href="#">getHandlerThreadPool()</a> Get the VHost handler thread pool.
String	<a href="#">getHomePath()</a> Get vHost configuration path.
HostPortList	<a href="#">getHostPortsList()</a> Get list of host port definitions for vHost.
IHTTPStreamerAdapter	<a href="#">getHTTPStreamerAdapter(String ID)</a> Get an HTTPStreamerAdapter by ID
java.util.List	<a href="#">getHTTPStreamerAdapterIDs()</a> Get a list of HTTPStreamerAdapter IDs
HTTPStreamerContext	<a href="#">getHTTPStreamerContext()</a> Get the HTTPStreamer (Cupertino Streaming and Silverlight Smooth Streaming) host context
HTTPStreamerList	<a href="#">getHTTPStreamerList()</a> Get the list of HTTPStreamers
int	<a href="#">getHTTPStreamerMaxPathLen()</a> Get the max HTTP streamer path length
IVHostHTTPStreamerRequestValidator	<a href="#">getHTTPStreamerRequestValidator()</a> Get the HTTP streamer request validator
int	<a href="#">getIdleCheckFrequency()</a> Get idle check frequency (milliseconds)
int	<a href="#">getIdleMinimumWaitTime()</a> Get the minimum time (milliseconds) the idle worker thread will sleep before generating idle events
int	<a href="#">getIdleWorkerCount()</a> Get number of threads used to generate idle events
IdleWorkersUtil	<a href="#">getIdleWorkers()</a> Get the idle worker utility
IOPerformanceCounter	<a href="#">getIoPerformanceCounter()</a> Get vHost IO performance counter.
IOPerformanceCounter	<a href="#">getIoPerformanceCounter(int counterIndex)</a> Get vHost IO performance counter for a specific technology (see IVHost.COUNTER_*)
IOScheduler	<a href="#">getIOScheduler()</a> Get IO scheduler for vHost.

int	<a href="#"><code>getKeepAliveTimeout()</code></a> Get the RTMPT connection keep alive timeout
LiveStreamPacketizerList	<a href="#"><code>getLiveStreamPacketizerList()</code></a> Get the LiveStreamPacketizerList
ILiveStreamRecordManager	<a href="#"><code>getLiveStreamRecordManager()</code></a> Returns the LiveStreamRecordManager associated with this vhost This object contains the active LiveStreamRecorders
LiveStreamTranscoderList	<a href="#"><code>getLiveStreamTranscoderList()</code></a> Get the LiveStreamTranscoderList
long	<a href="#"><code>getLiveStreamTranscoderSessionCount()</code></a> Get the current number of live stream transcoder sessions.
WMSProperties	<a href="#"><code>getManagerProperties()</code></a> Get VHost's manager's properties collection
int	<a href="#"><code>getMaximumPendingReadBytes()</code></a> Set maximum number of bytes a client connection can have waiting to be written before the connection is terminated.
int	<a href="#"><code>getMaximumPendingWriteBytes()</code></a> Get maximum number a bytes a client connection can have waiting to be sent before the connection is terminated.
int	<a href="#"><code>getMaximumSetBufferTime()</code></a> Get maximum number of milliseconds allowed for the NetStream.setBufferTime(secs) call.
MediaCasterList	<a href="#"><code>getMediaCasterList()</code></a> Get the list of media caster definitions (MediaCaster.xml)
MediaCasterSettings	<a href="#"><code>getMediaCasterSettings()</code></a> Get the media caster settings
int	<a href="#"><code>getMediaReaderContentType(String mediaType)</code></a> Get the content type of a media stream name prefix (see IMediaReader.CONTENTTYPE_*)
MediaReaderList	<a href="#"><code>getMediaReaders()</code></a> Get the media readers attached to vHost (MediaReaders.xml).
MediaWriterList	<a href="#"><code>getMediaWriters()</code></a> Get the media writers attached to vHost (MediaWriters.xml).
java.util.Map	<a href="#"><code>getMp3TagMap()</code></a> Get MP3 tag map attached to vHost (MP3Tags.xml).
String	<a href="#"><code>getName()</code></a> Get vHost name
HostPortConfig	<a href="#"><code>getNetConnectionHostPortConfig()</code></a> Get the socket configuration for server to server connections
int	<a href="#"><code>getNetConnectionIdleFrequency()</code></a> Get server to server idle frequency (milliseconds)
int	<a href="#"><code>getNetConnectionProcessorCount()</code></a> Get net connection processor count.

int	<a href="#">getNextNetConnectionId()</a> Get next connection id.
int	<a href="#">getPingTimeout()</a> Get ping timeout (milliseconds)
<a href="#">WMSProperties</a>	<a href="#">getProperties()</a> Get properties attached to this vHost.
String	<a href="#">getProperty(String key)</a> Get virtual host property.
void	<a href="#">getProtocolUsage(boolean[] protocolsInUse)</a> Get the protocols in use by this application instance (see IApplicationInstance.PROTOCOLUSAGE_*)
<a href="#">PushPublishContext</a>	<a href="#">getPushPublishContext()</a> Get the push publish context
<a href="#">RTPContext</a>	<a href="#">getRTPContext()</a> Get the RTP (real time protocol) virtual host context
com.wowza.wms.rtp.tra nsport.RTPUDPDatagram Config	<a href="#">getRTPDatagramConfigIncoming()</a> Get the RTP Datagram Socket configuration
com.wowza.wms.rtp.tra nsport.RTPUDPDatagram Config	<a href="#">getRTPDatagramConfigOutgoing()</a> Get the RTP Datagram Socket configuration
int	<a href="#">getRTPIIdleFrequency()</a> Get default RTP idle frequency (milliseconds)
int	<a href="#">getRTSPMaxPathLen()</a> Get the max RTSP stream path length
<a href="#">IVHostRTSPRequestVali dator</a>	<a href="#">getRTSPRequestValidator()</a> Get the RTSP request validator
java.util.List	<a href="#">getStartupStreams()</a> Get the list of streams to start at virtual host startup
int	<a href="#">getStartupStreamsDelayTime()</a> Get startup streams delay time (milliseconds).
<a href="#">StreamList</a>	<a href="#">getStreamTypes()</a> Get default stream type.
<a href="#">ThreadPool</a>	<a href="#">getThreadPool()</a> Get the VHost handler thread pool.
TimedTextProviderConf igure	<a href="#">getTimedTextProviderConfig()</a> Get the config of TimedTextProviders
String	<a href="#">getTimeRunning()</a> Get the time vHost has been running.
double	<a href="#">getTimeRunningSeconds()</a> Get time running in seconds

com.wowza.wms.transcoder.vod.TranscoderVODContext	<a href="#"><code>getTranscoderVODContext()</code></a> Get the VOD transcoder context
<a href="#"><code>ThreadPool</code></a>	<a href="#"><code>getTransportThreadPool()</code></a> Get the VHost transport thread pool.
com.wowza.wms.rtp.transport.UDPTransportManager	<a href="#"><code>getUDPTransportManager()</code></a> Get the UDP transport manager.
int	<a href="#"><code>getUnidentifiedSessionTimeout()</code></a> Get the unidentified session timeout (TCP session that we are not able to identify)
int	<a href="#"><code>getValidationFrequency()</code></a> Get time between validation pings (milliseconds)
void	<a href="#"><code>init(String basePath)</code></a> Initialize vHost.
boolean	<a href="#"><code>isApplicationLoaded(String applicationName)</code></a> Return true if the application is loaded
boolean	<a href="#"><code>isShuttingDown()</code></a> Is the VHost shutting down
boolean	<a href="#"><code>isStartStarupStreams()</code></a> Returns true if the startup streams are to start and vhost startup
boolean	<a href="#"><code>isSuspended()</code></a> Returns true if all HostPorts connected to this VHost are suspended
void	<a href="#"><code>killClient(int clientId)</code></a> Remove client from vHost and send disconnect message.
void	<a href="#"><code>killRTSPSession(String rtspSessionId)</code></a> Kill an RTSP connection by the RTSP session id
void	<a href="#"><code>putHTTPStreamerAdapter(String ID, IHTTPStreamerAdapter adapter)</code></a> Add an HTTPStreamerAdapter
void	<a href="#"><code>removeAcceptorListener(IAcceptorNotify acceptorListener)</code></a> Remove acceptor listener.
void	<a href="#"><code>removeApplicationListener(IApplicationNotify applicationListener)</code></a> Remove application listener.
void	<a href="#"><code>removeClient(int clientId)</code></a> Remove client from vHost.
void	<a href="#"><code>removeIdleWorkerListener(IIIdleWorkerNotify idleWorkerListener)</code></a> Remove idleWorker listener
void	<a href="#"><code>reparentClient(IClient client)</code></a> Move a client object to a new vhost.
void	<a href="#"><code>setAdminInterfaceHostPort(HostPort adminInterfaceHostPort)</code></a> Set admin interface host port (not used)
void	<a href="#"><code>setApplicationTimeout(int applicationTimeout)</code></a> Set application time out (milliseconds).

void	<a href="#"><u>setClientIdleFrequency(int clientIdleFrequency)</u></a> Set default client idle frequency (milliseconds)
void	<a href="#"><u>setClientTimeout(int clientTimeout)</u></a> Set client timeout.
void	<a href="#"><u>setCoreHandlerPoolSize(int corePoolSize)</u></a> Set the handler core thread pool size.
void	<a href="#"><u>setCoreTransportPoolSize(int corePoolSize)</u></a> Set the transport core thread pool size.
void	<a href="#"><u>setDynamicLogProperties(java.util.Properties dynamicLogProperties)</u></a> Set the dynamic log properties set at the vhost level
void	<a href="#"><u>setFileIOPoolSize(int fileIOPoolSize)</u></a> Set default file io thread pool size.
void	<a href="#"><u>setHTTPStreamerMaxPathLen(int httpStreamerMaxPathLen)</u></a> Set the max HTTP streamer path length
void	<a href="#"><u>setHTTPStreamerRequestValidator(IVHostHTTPStreamerRequestValidator httpRequestValidator)</u></a> Set the HTTP streamer request validator
void	<a href="#"><u>setIdleCheckFrequency(int idleCheckFrequency)</u></a> Set idle check frequency (milliseconds)
void	<a href="#"><u>setIdleMinimumWaitTime(int idleMinimumWaitTime)</u></a> Set the minimum time (milliseconds) the idle worker thread will sleep before generating idle events
void	<a href="#"><u>setIdleWorkerCount(int idleWorkerCount)</u></a> Set number of threads used to generate idle events
void	<a href="#"><u>setKeepAliveTimeout(int keepAliveTimeout)</u></a> Set the RTMPT connection keep alive timeout
void	<a href="#"><u>setMaximumPendingReadBytes(int maximumPendingReaderBytes)</u></a> Get maximum number of bytes a client connection can have waiting to be written before the connection is terminated.
void	<a href="#"><u>setMaximumPendingWriteBytes(int maximumPendingWriteBytes)</u></a> Set maximum number a bytes a client connection can have waiting to be sent before the connection is terminated.
void	<a href="#"><u>setMaximumSetBufferTime(int maximumSetBufferTime)</u></a> Set maximum number of milliseconds allowed for the NetStream.setBufferTime(secs) call.
void	<a href="#"><u>setNetConnectionIdleFrequency(int netConnectionIdleFrequency)</u></a> Set the server to server idle frequency (milliseconds)
void	<a href="#"><u>setNetConnectionProcessorCount(int netConnectionProcessorCount)</u></a> Set net connection processor count.
void	<a href="#"><u>setPingTimeout(int pingTimeout)</u></a> Set ping timeout (milliseconds)
void	<a href="#"><u>setRTPIdleFrequency(int rtpIdleFrequency)</u></a> Set default RTP idle frequency (milliseconds)

void	<a href="#"><code>setRTSPMaxPathLen(int rtspStreamerMaxPathLen)</code></a> Set the max RTSP stream path length
void	<a href="#"><code>setRTSPRequestValidator(IVHostRTSPRequestValidator rtspRequestValidator)</code></a> Set the RTSP request validator
void	<a href="#"><code>setShuttingDown(boolean shuttingDown)</code></a> Set VHost shutting down flag
void	<a href="#"><code>setStartStarupStreams(boolean startStarupStreams)</code></a> Set to true to startup startup stream as vhost startup
void	<a href="#"><code>setStartupStreamsDelayTime(int startupStreamsDelayTime)</code></a> Set startup streams delay time (milliseconds).
void	<a href="#"><code>setUnidentifiedSessionTimeout(int unidentifiedSessionTimeout)</code></a> Set the unidentified session timeout (TCP session that we are not able to identify)
void	<a href="#"><code>setValidationFrequency(int validationFrequency)</code></a> Set time between validation pings (milliseconds)
void	<a href="#"><code>shutdown()</code></a> Shutdown.
void	<a href="#"><code>shutdownApplication(String appName)</code></a> Shutdown an application by name.
void	<a href="#"><code>shutdownApplication(String appName, boolean deleteFolder)</code></a> Shutdown an application by name.
boolean	<a href="#"><code>startApplicationInstance(String appName)</code></a> Start an application instance.
boolean	<a href="#"><code>startApplicationInstance(String appName, String appInstanceName)</code></a> Start an application instance.
void	<a href="#"><code>startStartupStreams()</code></a> Method to start startup streams
void	<a href="#"><code>stopStartupStreams()</code></a> Method to stop startup streams
void	<a href="#"><code>suspendAllHostPorts()</code></a> Suspend all HostPorts from accepting new connections.
boolean	<a href="#"><code>touchApplicationInstance(String appName)</code></a> 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="#"><code>touchApplicationInstance(String appName, String appInstanceName)</code></a> 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="#"><code>unbindAllHostPorts()</code></a> Unbind all HostPorts and drop all connections
void	<a href="#"><code>updateLoggingDuration()</code></a> Internal: update the internal logging values.

## Fields

### VHOST\_DEFAULT

```
public static final java.lang.String VHOST_DEFAULT
```

Constant value: `_defaultVHost_`

### LICENSECOUNTER\_PUBLISHER

```
public static final int LICENSECOUNTER_PUBLISHER
```

Constant value: `0`

### LICENSECOUNTER\_TRANSCODE\_DECODE

```
public static final int LICENSECOUNTER_TRANSCODE_DECODE
```

Constant value: `1`

### LICENSECOUNTER\_TRANSCODE\_ENCODE

```
public static final int LICENSECOUNTER_TRANSCODE_ENCODE
```

Constant value: `2`

### LICENSECOUNTER\_NDVR

```
public static final int LICENSECOUNTER_NDVR
```

Constant value: `3`

### LICENSECOUNTER\_DRM\_EZDRM\_LIVE

```
public static final int LICENSECOUNTER_DRM_EZDRM_LIVE
```

Constant value: `4`

### LICENSECOUNTER\_DRM\_EZDRM\_VOD

```
public static final int LICENSECOUNTER_DRM_EZDRM_VOD
```

Constant value: `5`

### LICENSECOUNTER\_DRM\_VERIMATRIX\_LIVE

```
public static final int LICENSECOUNTER_DRM_VERIMATRIX_LIVE
```

Constant value: `6`

---

**LICENSECOUNTER\_DRM\_VERIMATRIX\_VOD**

```
public static final int LICENSECOUNTER_DRM_VERIMATRIX_VOD
```

Constant value: 7

---

**LICENSECOUNTER\_PUBLISHERTRANSCODER**

```
public static final int LICENSECOUNTER_PUBLISHERTRANSCODER
```

Constant value: 8

---

**LICENSECOUNTER\_DRM\_BUYDRM\_LIVE**

```
public static final int LICENSECOUNTER_DRM_BUYDRM_LIVE
```

Constant value: 9

---

**LICENSECOUNTER\_DRM\_BUYDRM\_VOD**

```
public static final int LICENSECOUNTER_DRM_BUYDRM_VOD
```

Constant value: 10

---

**LICENSECOUNTER\_TRANSCODE\_DECODEPOLLING**

```
public static final int LICENSECOUNTER_TRANSCODE_DECODEPOLLING
```

Constant value: 11

---

**LICENSECOUNTER\_TRANSCODE\_ENCODEPOLLING**

```
public static final int LICENSECOUNTER_TRANSCODE_ENCODEPOLLING
```

Constant value: 12

---

**LICENSECOUNTER\_TRANSCODE\_STREAMNAMES**

```
public static final int LICENSECOUNTER_TRANSCODE_STREAMNAMES
```

Constant value: 13

---

**LICENSECOUNTER\_TRANSCODE\_DECODECOUNTAUDIO**

```
public static final int LICENSECOUNTER_TRANSCODE_DECODECOUNTAUDIO
```

Constant value: 14

---

(continued from last page)

**LICENSECOUNTER\_TRANSCODE\_DECODECOUNTVIDEO**public static final int **LICENSECOUNTER\_TRANSCODE\_DECODECOUNTVIDEO**Constant value: **15****LICENSECOUNTER\_TRANSCODE\_DECODECOUNTAUDIOVIDEO**public static final int **LICENSECOUNTER\_TRANSCODE\_DECODECOUNTAUDIOVIDEO**Constant value: **16****LICENSECOUNTER\_TRANSCODE\_ENCODECOUNTAUDIO**public static final int **LICENSECOUNTER\_TRANSCODE\_ENCODECOUNTAUDIO**Constant value: **17****LICENSECOUNTER\_TRANSCODE\_ENCODECOUNTVIDEO**public static final int **LICENSECOUNTER\_TRANSCODE\_ENCODECOUNTVIDEO**Constant value: **18****LICENSECOUNTER\_TRANSCODE\_ENCODECOUNTAUDIOVIDEO**public static final int **LICENSECOUNTER\_TRANSCODE\_ENCODECOUNTAUDIOVIDEO**Constant value: **19****LICENSECOUNTER\_TOTAL**public static final int **LICENSECOUNTER\_TOTAL**Constant value: **20****LICENSECOUNTERS\_DRM**public static final int **LICENSECOUNTERS\_DRM****COUNTER\_RTMP**public static final int **COUNTER\_RTMP**Constant value: **0****COUNTER\_RTP**public static final int **COUNTER\_RTP**

---

(continued from last page)

Constant value: **1**

---

## COUNTER\_HTTPCUPERTINO

```
public static final int COUNTER_HTTPCUPERTINO
```

Constant value: **2**

---

## COUNTER\_HTTPSMOOTH

```
public static final int COUNTER_HTTPSMOOTH
```

Constant value: **3**

---

## COUNTER\_HTTPSANJOSE

```
public static final int COUNTER_HTTPSANJOSE
```

Constant value: **4**

---

## COUNTER\_HTTPWEBM

```
public static final int COUNTER_HTTPWEBM
```

Constant value: **5**

---

## COUNTER\_HTTPMPGEGDASH

```
public static final int COUNTER_HTTPMPGEGDASH
```

Constant value: **6**

---

## COUNTER\_HTTPDVRCHUNKS

```
public static final int COUNTER_HTTPDVRCHUNKS
```

Constant value: **7**

---

## COUNTER\_TOTAL

```
public static final int COUNTER_TOTAL
```

Constant value: **8**

---

## CODEC\_VIDEO\_UNKNOWN

```
public static final int CODEC_VIDEO_UNKNOWN
```

Constant value: **-1**

---

---

(continued from last page)

## **CODEC\_VIDEO\_SPARK**

```
public static final int CODEC_VIDEO_SPARK
```

Constant value: **2**

---

## **CODEC\_VIDEO\_SCREEN**

```
public static final int CODEC_VIDEO_SCREEN
```

Constant value: **3**

---

## **CODEC\_VIDEO\_VP6**

```
public static final int CODEC_VIDEO_VP6
```

Constant value: **4**

---

## **CODEC\_VIDEO\_VP6A**

```
public static final int CODEC_VIDEO_VP6A
```

Constant value: **5**

---

## **CODEC\_VIDEO\_SCREEN2**

```
public static final int CODEC_VIDEO_SCREEN2
```

Constant value: **6**

---

## **CODEC\_VIDEO\_H264**

```
public static final int CODEC_VIDEO_H264
```

Constant value: **7**

---

## **CODEC\_VIDEO\_VP8**

```
public static final int CODEC_VIDEO_VP8
```

Constant value: **8**

---

## **CODEC\_VIDEO\_H263**

```
public static final int CODEC_VIDEO_H263
```

Constant value: **9**

---

## **CODEC\_VIDEO\_MPEG4**

```
public static final int CODEC_VIDEO_MPEG4
```

---

(continued from last page)

Constant value: **10**

---

## **CODEC\_VIDEO\_MPEG2**

```
public static final int CODEC_VIDEO_MPEG2
```

Constant value: **11**

---

## **CODEC\_VIDEO\_H265**

```
public static final int CODEC_VIDEO_H265
```

Constant value: **12**

---

## **CODEC\_AUDIO\_UNKNOWN**

```
public static final int CODEC_AUDIO_UNKNOWN
```

Constant value: **-1**

---

## **CODEC\_AUDIO\_PCM\_BE**

```
public static final int CODEC_AUDIO_PCM_BE
```

Constant value: **0**

---

## **CODEC\_AUDIO\_PCM\_SWF**

```
public static final int CODEC_AUDIO_PCM_SWF
```

Constant value: **1**

---

## **CODEC\_AUDIO\_AC3**

```
public static final int CODEC_AUDIO_AC3
```

Constant value: **1**

---

## **CODEC\_AUDIO\_MP3**

```
public static final int CODEC_AUDIO_MP3
```

Constant value: **2**

---

## **CODEC\_AUDIO\_PCM\_LE**

```
public static final int CODEC_AUDIO_PCM_LE
```

Constant value: **3**

---

---

**CODEC\_AUDIO\_NELLYMOSER\_16MONO**

```
public static final int CODEC_AUDIO_NELLYMOSER_16MONO
```

Constant value: **4**

---

**CODEC\_AUDIO\_NELLYMOSER\_8MONO**

```
public static final int CODEC_AUDIO_NELLYMOSER_8MONO
```

Constant value: **5**

---

**CODEC\_AUDIO\_NELLYMOSER**

```
public static final int CODEC_AUDIO_NELLYMOSER
```

Constant value: **6**

---

**CODEC\_AUDIO\_G711\_ALAW**

```
public static final int CODEC_AUDIO_G711_ALAW
```

Constant value: **7**

---

**CODEC\_AUDIO\_G711\_MULAW**

```
public static final int CODEC_AUDIO_G711_MULAW
```

Constant value: **8**

---

**CODEC\_AUDIO\_RESERVED**

```
public static final int CODEC_AUDIO_RESERVED
```

Constant value: **9**

---

**CODEC\_AUDIO\_VORBIS**

```
public static final int CODEC_AUDIO_VORBIS
```

Constant value: **9**

---

**CODEC\_AUDIO\_AAC**

```
public static final int CODEC_AUDIO_AAC
```

Constant value: **10**

---

---

(continued from last page)

## **CODEC\_AUDIO\_SPEEX**

```
public static final int CODEC_AUDIO_SPEEX
```

Constant value: **11**

---

## **CODEC\_AUDIO\_MP3\_8**

```
public static final int CODEC_AUDIO_MP3_8
```

Constant value: **15**

---

## **CODEC\_STREAM\_UNKNOWN**

```
public static final int CODEC_STREAM_UNKNOWN
```

Constant value: **-1**

---

## **CODEC\_STREAM\_MP2T**

```
public static final int CODEC_STREAM_MP2T
```

Constant value: **0**

---

## **FILEFORMAT\_UNKNOWN**

```
public static final int FILEFORMAT_UNKNOWN
```

Constant value: **-1**

---

## **FILEFORMAT\_FLV**

```
public static final int FILEFORMAT_FLV
```

Constant value: **1**

---

## **FILEFORMAT\_MP4**

```
public static final int FILEFORMAT_MP4
```

Constant value: **2**

---

## **CONTENTTYPE\_UNKNOWN**

```
public static final byte CONTENTTYPE_UNKNOWN
```

AMF Content type: unknown  
Constant value: **0**

---

## **CONTENTTYPE\_SETCUNKSIZE**

```
public static final byte CONTENTTYPE_SETCUNKSIZE
```

---

(continued from last page)

AMF Content type: set packet chunk size  
Constant value: **1**

---

## **CONTENTTYPE\_WATCHDOG**

```
public static final byte CONTENTTYPE_WATCHDOG
```

AMF Content type: watch dog  
Constant value: **3**

---

## **CONTENTTYPE\_BUFFERSIZE**

```
public static final byte CONTENTTYPE_BUFFERSIZE
```

AMF Content type: set buffer size  
Constant value: **4**

---

## **CONTENTTYPE\_ACKBANDWIDTH**

```
public static final byte CONTENTTYPE_ACKBANDWIDTH
```

AMF Content type: set acknowledge bandwidth size  
Constant value: **5**

---

## **CONTENTTYPE\_SETBANDWIDTH**

```
public static final byte CONTENTTYPE_SETBANDWIDTH
```

AMF Content type: set bandwidth size  
Constant value: **6**

---

## **CONTENTTYPE\_AUDIO**

```
public static final byte CONTENTTYPE_AUDIO
```

AMF Content type: audio packet  
Constant value: **8**

---

## **CONTENTTYPE\_VIDEO**

```
public static final byte CONTENTTYPE_VIDEO
```

AMF Content type: video packet  
Constant value: **9**

---

## **CONTENTTYPE\_DATA**

```
public static final byte CONTENTTYPE_DATA
```

AMF Content type: data packet  
Constant value: **18**

---

## **CONTENTTYPE\_DATA0**

```
public static final byte CONTENTTYPE_DATA0
```

AMF Content type: data packet (AMF0)  
Constant value: **18**

---

---

## CONTENTTYPE\_DATA3

```
public static final byte CONTENTTYPE_DATA3
```

AMF Content type: data packet (AMF3)  
Constant value: **15**

---

## CONTENTTYPE\_SHAREDOBJECTS

```
public static final byte CONTENTTYPE_SHAREDOBJECTS
```

AMF Content type: shared object packet (AMF0)  
Constant value: **19**

---

## CONTENTTYPE\_SHAREDOBJECTS0

```
public static final byte CONTENTTYPE_SHAREDOBJECTS0
```

AMF Content type: shared object packet (AMF0)  
Constant value: **19**

---

## CONTENTTYPE\_SHAREDOBJECTS3

```
public static final byte CONTENTTYPE_SHAREDOBJECTS3
```

AMF Content type: shared object packet (AMF3)  
Constant value: **16**

---

## CONTENTTYPE\_FUNCTION

```
public static final byte CONTENTTYPE_FUNCTION
```

AMF Content type: function data (AMF0)  
Constant value: **20**

---

## CONTENTTYPE\_FUNCTION0

```
public static final byte CONTENTTYPE_FUNCTION0
```

AMF Content type: function data (AMF0)  
Constant value: **20**

---

## CONTENTTYPE\_MEDIACHUNK

```
public static final byte CONTENTTYPE_MEDIACHUNK
```

AMF Content type: media chunk  
Constant value: **22**

---

## CONTENTTYPE\_FUNCTION3

```
public static final byte CONTENTTYPE_FUNCTION3
```

AMF Content type: function data (AMF3)  
Constant value: **17**

---

(continued from last page)

## CONTENTTYPE\_PLAYCALLBACK

```
public static final byte CONTENTTYPE_PLAYCALLBACK
```

AMF Content type: play callback  
Constant value: **127**

## ACCEPTORS\_ACCEPTOR

```
public static final java.lang.String ACCEPTORS_ACCEPTOR
```

Acceptor types: acceptor  
Constant value: **acceptor**

## ACCEPTORS\_HANDLERADAPTER

```
public static final java.lang.String ACCEPTORS_HANDLERADAPTER
```

Acceptor types: handler  
Constant value: **handlerAdapter**

## Methods

### init

```
public void init(String basePath)
```

Initialize vHost.

#### Parameters:

basePath - base path to configuration files

### shutdown

```
public void shutdown()
```

Shutdown.

### getStreamTypes

```
public StreamList getStreamTypes()
```

Get default stream type.

#### Returns:

default stream type

### getProperty

```
public String getProperty(String key)
```

Get virtual host property.

#### Parameters:

key - key

#### Returns:

property value or null if does not exist

---

## getClientTimeout

```
public int getClientTimeout()
```

Get client timeout.

**Returns:**

client timeout

---

## getNextNetConnectionId

```
public int getNextNetConnectionId()
```

Get next connection id.

**Returns:**

next connection id

---

## getClient

```
public IClient getClient(int clientId)
```

Get client by client id.

**Parameters:**

clientId - client id

**Returns:**

client

---

## getClient

```
public IClient getClient(int clientId,  
                           boolean create)
```

Get client by client id and create if does not exist.

**Parameters:**

clientId - client id

create - create if does not exist

**Returns:**

client

---

## removeClient

```
public void removeClient(int clientId)
```

Remove client from vHost.

**Parameters:**

clientId - client id

---

## killClient

```
public void killClient(int clientId)
```

Remove client from vHost and send disconnect message.

(continued from last page)

**Parameters:**

clientId - client id

**killRTSPSession**

```
public void killRTSPSession(String rtspSessionId)
```

Kill an RTSP connection by the RTSP session id

**Parameters:**

rtspSessionId - RTSP session id

**getApplication**

```
public IApplication getApplication(String applicationName)
```

Get application by name.

**Parameters:**

applicationName - application name

**Returns:**

application

**isApplicationLoaded**

```
public boolean isApplicationLoaded(String applicationName)
```

Return true if the application is loaded

**Parameters:**

applicationName - application name

**Returns:**

true if application is loaded

**getHomePath**

```
public String getHomePath()
```

Get vHost configuration path.

**Returns:**

configuration path

**getProperties**

```
public WMSProperties getProperties()
```

Get properties attached to this vHost.

**Returns:**

properties attached to this vHost

**getManagerProperties**

```
public WMSProperties getManagerProperties()
```

(continued from last page)

Get VHost's manager's properties collection

**Returns:**

manager's property collection

---

**getCoreTransportPoolSize**

```
public int getCoreTransportPoolSize()
```

Get the transport core thread pool size.

**Returns:**

default core thread pool size

---

**setCoreTransportPoolSize**

```
public void setCoreTransportPoolSize(int corePoolSize)
```

Set the transport core thread pool size.

**Parameters:**

corePoolSize - core thread pool size

---

**getCoreHandlerPoolSize**

```
public int getCoreHandlerPoolSize()
```

Get the handler core thread pool size.

**Returns:**

default core thread pool size

---

**setCoreHandlerPoolSize**

```
public void setCoreHandlerPoolSize(int corePoolSize)
```

Set the handler core thread pool size.

**Parameters:**

corePoolSize - core thread pool size

---

**getFileIOPoolSize**

```
public int getFileIOPoolSize()
```

Get the default file io pool size.

**Returns:**

default file io pool size

---

**setFileIOPoolSize**

```
public void setFileIOPoolSize(int fileIOPoolSize)
```

Set default file io thread pool size.

**Parameters:**

fileIOPoolSize - default file io thread pool size

---

## setClientTimeout

```
public void setClientTimeout(int clientTimeout)
```

Set client timeout. An inactive client connected by RTMPT protocol will be deleted after this timeout.

**Parameters:**

clientTimeout - client timeout

---

## getHostPortsList

```
public HostPortList getHostPortsList()
```

Get list of host port definitions for vHost.

**Returns:**

list of host port definitions for vHost

---

## getThreadPool

```
public ThreadPool getThreadPool()
```

Get the VHost handler thread pool. Same as getHandlerThreadPool.

**Returns:**

VHost handler thread pool

---

## getTransportThreadPool

```
public ThreadPool getTransportThreadPool()
```

Get the VHost transport thread pool. This thread pool is used to read/write data from the transports sockets.

**Returns:**

VHost transport thread pool

---

## getHandlerThreadPool

```
public ThreadPool getHandlerThreadPool()
```

Get the VHost handler thread pool. This thread pool is used to process the incoming events.

**Returns:**

VHost handler thread pool

---

## getName

```
public String getName()
```

Get vHost name

**Returns:**

vHost name

---

## setAdminInterfaceHostPort

```
public void setAdminInterfaceHostPort(HostPort adminInterfaceHostPort)
```

(continued from last page)

Set admin interface host port (not used)

**Parameters:**

adminInterfaceHostPort - admin interface host port

---

**getClientCount**

```
public int getClientCount()
```

Get number of clients connected to this vHost.

**Returns:**

number of clients connected to this vHost

---

**getNetConnectionProcessorCount**

```
public int getNetConnectionProcessorCount()
```

Get net connection processor count. Number of threads used for server to server communication (not finished).

**Returns:**

net connection processor count

---

**setNetConnectionProcessorCount**

```
public void setNetConnectionProcessorCount(int netConnectionProcessorCount)
```

Set net connection processor count. Number of threads used for server to server communication (not finished).

**Parameters:**

netConnectionProcessorCount - net connection processor count

---

**addApplicationListener**

```
public void addApplicationListener(IAplicationNotify applicationListener)
```

Add application listener. An application listener will receive the following events: onApplicationCreate, onApplicationDestroy.

**Parameters:**

applicationListener - application listener

---

**removeApplicationListener**

```
public void removeApplicationListener(IAplicationNotify 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 fly serving.

**Returns:**

IO scheduler for vHost

---

## getIoPerformanceCounter

```
public IOPerformanceCounter getIoPerformanceCounter()
```

Get vHost IO performance counter.

**Returns:**

io performance counter

---

## getIoPerformanceCounter

```
public IOPerformanceCounter getIoPerformanceCounter(int counterIndex)
```

Get vHost IO performance counter for a specific technology (see IVHost.COUNTER\_\*)

**Parameters:**

counterIndex - counter index (see IVHost.COUNTER\_\*)

**Returns:**

io performance counter

---

## getConnectionCounter

```
public ConnectionCounter getConnectionCounter()
```

Get vHost connection counter.

**Returns:**

connection counter

---

## getConnectionCounter

```
public ConnectionCounterSimple getConnectionCounter(int counterIndex)
```

Get vHost connection counter for a specific technology (see IVHost.COUNTER\_\*)

**Parameters:**

counterIndex - counter index (see IVHost.COUNTER\_\*)

(continued from last page)

**Returns:**

connection counter

---

**getDateStarted**

```
public String getDateStarted()
```

Get date and time the server was started.

**Returns:**

date and time the server was started

---

**getTimeRunning**

```
public String getTimeRunning()
```

Get the time vHost has been running.

**Returns:**

formatted string with vHost uptime

---

**getTimeRunningSeconds**

```
public double getTimeRunningSeconds()
```

Get time running in seconds

**Returns:**

time running in seconds

---

**getConnectionLimit**

```
public int getConnectionLimit()
```

Get vHost connection limit.

**Returns:**

vHost connection limit

---

**getMediaReaders**

```
public MediaReaderList getMediaReaders()
```

Get the media readers attached to vHost (MediaReaders.xml).

**Returns:**

media readers attached to vHost

---

**getMediaWriters**

```
public MediaWriterList getMediaWriters()
```

Get the media writers attached to vHost (MediaWriters.xml).

**Returns:**

media writers attached to vHost

(continued from last page)

## getMp3TagMap

```
public java.util.Map getMp3TagMap()
```

Get MP3 tag map attached to vHost (MP3Tags.xml).

**Returns:**

MP3 tag map attached to vHost

---

## getCEACaptionConversionContext

```
public CEACaptionConversionContext getCEACaptionConversionContext()
```

Get CEACaptionConversionContextMP3 tag map attached to vHost (CEACaptionConversion.xml).

**Returns:**

MP3 tag map attached to vHost

---

## updateLoggingDuration

```
public void updateLoggingDuration()
```

Internal: update the internal logging values.

---

## getApplicationTimeout

```
public int getApplicationTimeout()
```

Get application time out (milliseconds). Time from last client disconnect to application destruction.

**Returns:**

application time out (milliseconds)

---

## setApplicationTimeout

```
public void setApplicationTimeout(int applicationTimeout)
```

Set application time out (milliseconds).

**Parameters:**

applicationTimeout - application time out (milliseconds)

---

## getPingTimeout

```
public int getPingTimeout()
```

Get ping timeout (milliseconds)

**Returns:**

ping timeout (milliseconds)

---

## setPingTimeout

```
public void setPingTimeout(int pingTimeout)
```

Set ping timeout (milliseconds)

**Parameters:**

(continued from last page)

`pingTimeout - ping timeout (milliseconds)`

## **getValidationFrequency**

`public int getValidationFrequency()`

Get time between validation pings (milliseconds)

**Returns:**

time between validation pings (milliseconds)

## **setValidationFrequency**

`public void setValidationFrequency(int validationFrequency)`

Set time between validation pings (milliseconds)

**Parameters:**

`validationFrequency - time between validation pings (milliseconds)`

## **getMaximumPendingWriteBytes**

`public int getMaximumPendingWriteBytes()`

Get maximum number a bytes a client connection can have waiting to be sent before the connection is terminated. If set to zero this feature is turned off.

**Returns:**

maximum number a bytes a client connection can have waiting to be sent before the connection is terminated

## **setMaximumPendingWriteBytes**

`public void setMaximumPendingWriteBytes(int maximumPendingWriteBytes)`

Set maximum number a bytes a client connection can have waiting to be sent before the connection is terminated. If set to zero this feature is turned off.

**Parameters:**

`maximumPendingWriteBytes - maximum number a bytes a client connection can have waiting to be sent before the connection is terminated`

## **getMaximumPendingReadBytes**

`public int getMaximumPendingReadBytes()`

Set maximum number of bytes a client connection can have waiting to be written before the connection is terminated. If set to zero this feature is off.

**Returns:**

maximum number of bytes a client connection can have waiting to be written before the connection is terminated

## **setMaximumPendingReadBytes**

`public void setMaximumPendingReadBytes(int maximumPendingReaderBytes)`

Get maximum number of bytes a client connection can have waiting to be written before the connection is terminated. If set to zero this feature is off.

**Parameters:**

---

(continued from last page)

`maximumPendingReaderBytes` - maximum number of bytes a client connection can have waiting to be written before the connection is terminated

---

## getMaximumSetBufferTime

```
public int getMaximumSetBufferTime()
```

Get maximum number of milliseconds allowed for the NetStream.setBufferTime(secs) call. If set to zero this feature is turned off.

**Returns:**

maximum number of milliseconds allowed for the NetStream.setBufferTime(secs) call

---

## setMaximumSetBufferTime

```
public void setMaximumSetBufferTime(int maximumSetBufferTime)
```

Set maximum number of milliseconds allowed for the NetStream.setBufferTime(secs) call. If set to zero this feature is turned off.

**Parameters:**

`maximumSetBufferTime` - maximum number of milliseconds allowed for the NetStream.setBufferTime(secs) call

---

## getApplicationLock

```
public edu.emory.mathcs.backport.java.util.concurrent.locks.WMSReadWriteLock  
getApplicationLock()
```

Get the object used in synchronized statements to lock and application.

**Returns:**

object used in synchronized statements to lock and application

---

## getClientIdleFrequency

```
public int getClientIdleFrequency()
```

Get default client idle frequency (milliseconds)

**Returns:**

default client idle frequency (milliseconds)

---

## setClientIdleFrequency

```
public void setClientIdleFrequency(int clientIdleFrequency)
```

Set default client idle frequency (milliseconds)

**Parameters:**

`clientIdleFrequency` - default client idle frequency (milliseconds)

---

## getRTPIIdleFrequency

```
public int getRTPIIdleFrequency()
```

Get default RTP idle frequency (milliseconds)

**Returns:**

default RTP idle frequency (milliseconds)

---

---

## setRTPIdleFrequency

```
public void setRTPIdleFrequency(int rtpIdleFrequency)
```

Set default RTP idle frequency (milliseconds)

**Parameters:**

rtpIdleFrequency - RTP default client idle frequency (milliseconds)

---

## getNetConnectionIdleFrequency

```
public int getNetConnectionIdleFrequency()
```

Get server to server idle frequency (milliseconds)

**Returns:**

server to server idle frequency (milliseconds)

---

## setNetConnectionIdleFrequency

```
public void setNetConnectionIdleFrequency(int netConnectionIdleFrequency)
```

Set the server to server idle frequency (milliseconds)

**Parameters:**

netConnectionIdleFrequency - server to server idle frequency (milliseconds)

---

## getIdleCheckFrequency

```
public int getIdleCheckFrequency()
```

Get idle check frequency (milliseconds)

**Returns:**

idle check frequency (milliseconds)

---

## setIdleCheckFrequency

```
public void setIdleCheckFrequency(int idleCheckFrequency)
```

Set idle check frequency (milliseconds)

**Parameters:**

idleCheckFrequency - idle check frequency (milliseconds)

---

## getIdleWorkerCount

```
public int getIdleWorkerCount()
```

Get number of threads used to generate idle events

**Returns:**

number of threads used to generate idle events

---

## setIdleWorkerCount

```
public void setIdleWorkerCount(int idleWorkerCount)
```

(continued from last page)

Set number of threads used to generate idle events

**Parameters:**

idleWorkerCount - number of threads used to generate idle events

---

**getKeepAliveTimeout**

```
public int getKeepAliveTimeout()
```

Get the RTMPT connection keep alive timeout

**Returns:**

RTMPT connection keep alive timeout

---

**setKeepAliveTimeout**

```
public void setKeepAliveTimeout(int keepAliveTimeout)
```

Set the RTMPT connection keep alive timeout

**Parameters:**

keepAliveTimeout - RTMPT connection keep alive timeout

---

**addIdleWorkerListener**

```
public void addIdleWorkerListener(IIdleWorkerNotify idleWorkerListener)
```

Add idleWorker listener. An idleWorker listener will receive the following events: onIdleWorkerCreate, onIdleWorkerDestroy.

**Parameters:**

idleWorkerListener - idleWorker listener

---

**removeIdleWorkerListener**

```
public void removeIdleWorkerListener(IIdleWorkerNotify idleWorkerListener)
```

Remove idleWorker listener

**Parameters:**

idleWorkerListener - idleWorker listener

---

**getNetConnectionHostPortConfig**

```
public HostPortConfig getNetConnectionHostPortConfig()
```

Get the socket configuration for server to server connections

**Returns:**

socket configuration for server to server connections

---

**getMediaCasterSettings**

```
public MediaCasterSettings getMediaCasterSettings()
```

Get the media caster settings

**Returns:**

media caster settings

---

## getMediaCasterList

```
public MediaCasterList getMediaCasterList()
```

Get the list of media caster definitions (MediaCaster.xml)

**Returns:**

list of media caster definitions

---

## getApplicationNames

```
public java.util.List getApplicationNames()
```

Get a list of application names

**Returns:**

list of application names

---

## getApplicationFolderNames

```
public java.util.List getApplicationFolderNames()
```

Get a list of application folder names

**Returns:**

list of application folder names

---

## applicationExists

```
public boolean applicationExists(String name)
```

Return true if an application folder exists for this application name

**Parameters:**

name - application name

**Returns:**

true if an application folder exists for this application name

---

## getRTPContext

```
public RTPContext getRTPContext()
```

Get the RTP (real time protocol) virtual host context

**Returns:**

RTP (real time protocol) virtual host context

---

## getHTTPStreamerContext

```
public HTTPStreamerContext getHTTPStreamerContext()
```

Get the HTTPStreamer (Cupertino Streaming and Silverlight Smooth Streaming) host context

**Returns:**

HTTPStreamer (Cupertino Streaming and Silverlight Smooth Streaming) host context

---

---

## getRTPDatagramConfigIncoming

```
public com.wowza.wms.rtp.transport.RTPUDPDatagramConfig getRTPDatagramConfigIncoming()
```

Get the RTP Datagram Socket configuration

**Returns:**

RTP Datagram Socket configuration

---

## getRTPDatagramConfigOutgoing

```
public com.wowza.wms.rtp.transport.RTPUDPDatagramConfig getRTPDatagramConfigOutgoing()
```

Get the RTP Datagram Socket configuration

**Returns:**

RTP Datagram Socket configuration

---

## getAuthenticationList

```
public AuthenticationList getAuthenticationList()
```

Get the list of available authentication methods

**Returns:**

list of available authentication methods

---

## getIdleMinimumWaitTime

```
public int getIdleMinimumWaitTime()
```

Get the minimum time (milliseconds) the idle worker thread will sleep before generating idle events

**Returns:**

minimum time (milliseconds) the idle worker thread will sleep before generating idle events

---

## setIdleMinimumWaitTime

```
public void setIdleMinimumWaitTime(int idleMinimumWaitTime)
```

Set the minimum time (milliseconds) the idle worker thread will sleep before generating idle events

**Parameters:**

idleMinimumWaitTime - minimum time (milliseconds) the idle worker thread will sleep before generating idle events

---

## getIdleWorkers

```
public IdleWorkersUtil getIdleWorkers()
```

Get the idle worker utility

**Returns:**

idle worker utility

---

## isShuttingDown

```
public boolean isShuttingDown()
```

(continued from last page)

Is the VHost shutting down

**Returns:**

true if the vhost is shutting down

---

**setShuttingDown**

```
public void setShuttingDown(boolean shuttingDown)
```

Set VHost shutting down flag

**Parameters:**

shuttingDown - true if the vhost is shutting down

---

**reparentClient**

```
public void reparentClient(IClient client)
```

Move a client object to a new vhost. This can only be done right after the handshake process has completed. See IVHostNotify.onVHostClientConnect.

**Parameters:**

client - client object to move

---

**getDynamicLogProperties**

```
public java.util.Properties getDynamicLogProperties()
```

Get the dynamic log properties defined at the vhost level in conf/log4j.properties

**Returns:**

dynamic log properties defined at the vhost level

---

**setDynamicLogProperties**

```
public void setDynamicLogProperties(java.util.Properties dynamicLogProperties)
```

Set the dynamic log properties set at the vhost level

**Parameters:**

dynamicLogProperties - dynamic log properties defined at the vhost level

---

**shutdownApplication**

```
public void shutdownApplication(String appName)
```

Shutdown an application by name. This will disconnect all clients connected to all child application instances.

**Parameters:**

appName - application name

---

**shutdownApplication**

```
public void shutdownApplication(String appName,
                               boolean deleteFolder)
```

Shutdown an application by name. This will disconnect all clients connected to all child application instances.

(continued from last page)

**Parameters:**

appName - application name  
deleteFolder - delete the application folder

---

**unbindAllHostPorts**

```
public void unbindAllHostPorts()
```

Unbind all HostPorts and drop all connections

---

**suspendAllHostPorts**

```
public void suspendAllHostPorts()
```

Suspend all HostPorts from accepting new connections. Current connections will continue to be serviced

---

**closeHostPort**

```
public void closeHostPort(HostPort hostPort, boolean isSuspend)
```

Close an individual HostPort

**Parameters:**

hostPort - host port to close  
isSuspend - if true will just suspend the HostPort from accepting new connections, if false will unbind and drop all connections

---

**isSuspended**

```
public boolean isSuspended()
```

Returns true is all HostPorts connected to this VHost are suspended

---

**touchApplicationInstance**

```
public boolean touchApplicationInstance(String appName)
```

Start an application instance if it is not already started then touch it so it stays loaded for at least 3 seconds.

**Parameters:**

appName - application name

**Returns:**

true is successful

---

**touchApplicationInstance**

```
public boolean touchApplicationInstance(String appName, String appInstanceName)
```

Start an application instance if it is not already started then touch it so it stays loaded for at least 3 seconds. The default appInstanceName \_definst\_ will be used.

**Parameters:**

appName - application name  
appInstanceName - app instance name

**Returns:**

(continued from last page)

true is successful

---

## startApplicationInstance

```
public boolean startApplicationInstance(String appName)
```

Start an application instance. The default appInstanceName \_definst\_ will be used.

**Parameters:**

appName - application name

**Returns:**

true is successful

---

## startApplicationInstance

```
public boolean startApplicationInstance(String appName,  
String appInstanceName)
```

Start an application instance.

**Parameters:**

appName - application name

appInstanceName - app instance name

**Returns:**

true is successful

---

## getUDPTransportManager

```
public com.wowza.wms.rtp.transport.UDPTransportManager getUDPTransportManager()
```

Get the UDP transport manager.

**Returns:**

UDP transport manager

---

## getHTTPStreamerList

```
public HTTPStreamerList getHTTPStreamerList()
```

Get the list of HTTPStreamers

**Returns:**

list of HTTPStreamers

---

## getHTTPStreamerAdapter

```
public IHTTPStreamerAdapter getHTTPStreamerAdapter(String ID)
```

Get an HTTPStreamerAdapter by ID

**Parameters:**

ID - HTTPStreamerAdapter ID

**Returns:**

HTTPStreamerAdapter

---

## putHTTPStreamerAdapter

```
public void putHTTPStreamerAdapter(String ID,  
        IHTTPStreamerAdapter adapter)
```

Add an HTTPStreamerAdapter

**Parameters:**

ID - HTTPStreamerAdapter ID  
adapter - HTTPStreamerAdapter

---

## getHTTPStreamerAdapterIDs

```
public java.util.List getHTTPStreamerAdapterIDs()
```

Get a list of HTTPStreamerAdapter IDs

**Returns:**

list of HTTPStreamerAdapter IDs

---

## getLiveStreamPacketizerList

```
public LiveStreamPacketizerList getLiveStreamPacketizerList()
```

Get the LiveStreamPacketizerList

**Returns:**

LiveStreamPacketizerList

---

## getLiveStreamTranscoderList

```
public LiveStreamTranscoderList getLiveStreamTranscoderList()
```

Get the LiveStreamTranscoderList

**Returns:**

LiveStreamTranscoderList

---

## getTimedTextProviderConfig

```
public TimedTextProviderConfiguration getTimedTextProviderConfig()
```

Get the config of TimedTextProviders

**Returns:**

TimedTextProviders config

---

## getStartupStreams

```
public java.util.List getStartupStreams()
```

Get the list of streams to start at virtual host startup

**Returns:**

list of streams to start at virtual host startup

---

(continued from last page)

## **addStartupStream**

```
public void addstartupStream(StartupStream startupStream)
```

Add a stream to the list of streams to start and virtual host startup

**Parameters:**

startupStream - startup stream

---

## **isStartStarupStreams**

```
public boolean isStartStarupStreams()
```

Returns true if the startup streams are to start and vhost startup

**Returns:**

true if the startup streams are to start and vhost startup

---

## **setStartStarupStreams**

```
public void setStartStarupStreams(boolean startStarupStreams)
```

Set to true to startup startup stream as vhost startup

**Parameters:**

startStarupStreams - true if the startup streams are to start and vhost startup

---

## **startStartupStreams**

```
public void startStartupStreams()
```

Method to start startup streams

---

## **stopStartupStreams**

```
public void stopStartupStreams()
```

Method to stop startup streams

---

## **getDvrRecorderList**

```
public DvrRecorderList getDvrRecorderList()
```

Get the DvrRecorderList

**Returns:**

list of DVR Recorders

---

## **getDvrStoreList**

```
public DvrStoreList getDvrStoreList()
```

Get the list of DVR Stores

**Returns:**

list of DVR Stores

(continued from last page)

## getLiveStreamRecordManager

```
public ILiveStreamRecordManager getLiveStreamRecordManager()
```

Returns the LiveStreamRecordManager associated with this vhost This object contains the active LiveStreamRecorders

**Returns:**

ILiveStreamRecordManager associated with this vhost

---

## getMediaReaderContentType

```
public int getMediaReaderContentType(String mediaType)
```

Get the content type of a media stream name prefix (see IMediaReader.CONTENTTYPE\_\*)

**Parameters:**

mediaType - mediaType (such as flv or smil)

**Returns:**

content type (see IMediaReader.CONTENTTYPE\_\*)

---

## getProtocolUsage

```
public void getProtocolUsage(boolean[] protocolsInUse)
```

Get the protocols in use by this application instance (see IApplicationInstance.PROTOCOLUSAGE\_\*)

---

## getTranscoderVODContext

```
public com.wowza.wms.transcoder.vod.TranscoderVODContext getTranscoderVODContext()
```

Get the VOD transcoder context

**Returns:**

VOD transcoder context

---

## getHTTPStreamerRequestValidator

```
public IVHostHTTPStreamerRequestValidator getHTTPStreamerRequestValidator()
```

Get the HTTP streamer request validator

**Returns:**

HTTP streamer request validator

---

## setHTTPStreamerRequestValidator

```
public void setHTTPStreamerRequestValidator(IVHostHTTPStreamerRequestValidator httpRequestValidator)
```

Set the HTTP streamer request validator

**Parameters:**

httpRequestValidator - HTTP streamer request validator

---

## getHTTPStreamerMaxPathLen

```
public int getHTTPStreamerMaxPathLen()
```

(continued from last page)

Get the max HTTP streamer path length

**Returns:**

max HTTP streamer path length

---

**setHTTPStreamerMaxPathLen**

```
public void setHTTPStreamerMaxPathLen(int httpStreamerMaxPathLen)
```

Set the max HTTP streamer path length

**Parameters:**

httpStreamerMaxPathLen - max HTTP streamer path length

---

**getRTSPRequestValidator**

```
public IVHostRTSPRequestValidator getRTSPRequestValidator()
```

Get the RTSP request validator

**Returns:**

RTSP request validator

---

**setRTSPRequestValidator**

```
public void setRTSPRequestValidator(IVHostRTSPRequestValidator rtspRequestValidator)
```

Set the RTSP request validator

**Parameters:**

rtspRequestValidator - RTSP request validator

---

**getRTSPMaxPathLen**

```
public int getRTSPMaxPathLen()
```

Get the max RTSP stream path length

**Returns:**

max RTSP stream path length

---

**setRTSPMaxPathLen**

```
public void setRTSPMaxPathLen(int rtspStreamerMaxPathLen)
```

Set the max RTSP stream path length

**Parameters:**

rtspStreamerMaxPathLen - max RTSP stream path length

---

**getLiveStreamTranscoderSessionCount**

```
public long getLiveStreamTranscoderSessionCount()
```

Get the current number of live stream transcoder sessions.

**Returns:**

current number of live stream transcoder sessions

---

## getPushPublishContext

```
public PushPublishContext getPushPublishContext()
```

Get the push publish context

**Returns:**

push publish context

---

## getUnidentifiedSessionTimeout

```
public int getUnidentifiedSessionTimeout()
```

Get the unidentified session timeout (TCP session that we are not able to identify)

**Returns:**

timeout in milliseconds

---

## setUnidentifiedSessionTimeout

```
public void setUnidentifiedSessionTimeout(int unidentifiedSessionTimeout)
```

Set the unidentified session timeout (TCP session that we are not able to identify)

**Parameters:**

unidentifiedSessionTimeout - timeout in milliseconds

---

## getStartupStreamsDelayTime

```
public int getStartupStreamsDelayTime()
```

Get startup streams delay time (milliseconds). Time between items started in StartupStreams.xml.

**Returns:**

delay time

---

## setStartupStreamsDelayTime

```
public void setStartupStreamsDelayTime(int startupStreamsDelayTime)
```

Set startup streams delay time (milliseconds). Time between items started in StartupStreams.xml.

**Parameters:**

startupStreamsDelayTime - delay time

---

**com.wowza.wms.vhost**  
**Interface IVHostHTTPStreamerRequestValidator**

public interface IVHostHTTPStreamerRequestValidator  
extends

---

## Method Summary

boolean	<a href="#">validateHTTPStreamerRequest</a> (RtmpRequestMessage request, <a href="#">HostPort</a> hostPort, String path)
---------	---

---

## Methods

### [validateHTTPStreamerRequest](#)

public boolean [validateHTTPStreamerRequest](#)(RtmpRequestMessage request,  
[HostPort](#) hostPort,  
String path)

## com.wowza.wms.vhost Interface IVHostItemNotify

public interface IVHostItemNotify  
extends

IVHostItemNotify: listener interface used by VHostList addVHostItemListener

VHostList is the list vHost definitions in VHosts.xml not the vHosts themselves. With this listener interface you can monitor the vHost definition changes.

### Method Summary

void	<a href="#">onVHostItemCreate(VHostItem vhostItem)</a> Triggered when vHostItem created
void	<a href="#">onVHostItemDestroy(VHostItem vhostItem)</a> Triggered when vHostItem destroyed
void	<a href="#">onVHostItemUpdate(VHostItem vhostItem)</a> 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(IVHost vhost, IClient inClient, com.wowza.wms.request.RequestFunction function, AMFDataList params)</a> Triggered before a client connects to this virtual host.
void	<a href="#">onVHostCreate(IVHost vhost)</a> Triggered when vHost created
void	<a href="#">onVHostInit(IVHost vhost)</a> Triggered when vHost initialized
void	<a href="#">onVHostShutdownComplete(IVHost vhost)</a> Triggered at the end of vhost shutdown
void	<a href="#">onVHostShutdownStart(IVHost vhost)</a> Triggered at the beginning of vhost shutdown

### Methods

#### onVHostCreate

public void [onVHostCreate\(IVHost vhost\)](#)

Triggered when vHost created

**Parameters:**  
vhost

#### onVHostInit

public void [onVHostInit\(IVHost vhost\)](#)

Triggered when vHost initialized

**Parameters:**  
vhost - vhost

#### onVHostShutdownStart

public void [onVHostShutdownStart\(IVHost vhost\)](#)

Triggered at the beginning of vhost shutdown

(continued from last page)

**Parameters:**

vhost - vhost

---

**onVHostShutdownComplete**

```
public void onVHostShutdownComplete(IVHost vhost)
```

Triggered at the end of vhost shutdown

**Parameters:**

vhost - vhost

---

**onVHostClientConnect**

```
public void onVHostClientConnect(IVHost vhost,  
      IClient inClient,  
      com.wowza.wms.request.RequestFunction function,  
      AMFDataList params)
```

Triggered before a client connects to this virtual host. Provides an opportunity to rewrite the information that is being used to connect.

**Parameters:**

vhost

inClient - client object of the connection

function - function

params - parameters

**com.wowza.wms.vhost**  
**Interface IVHostRTSPRequestValidator**

public interface IVHostRTSPRequestValidator  
extends

---

## Method Summary

boolean	<a href="#">validateRTSPRequest</a> (com.wowza.wms.rtsp.RTSPRequestMessage req)
---------	---

---

## Methods

### validateRTSPRequest

public boolean **validateRTSPRequest**(com.wowza.wms.rtsp.RTSPRequestMessage req)

## com.wowza.wms.vhost Interface IWorkerThreadClear

---

public interface IWorkerThreadClear  
extends

IWorkerThreadClear: Internal use.

---

### Method Summary

void	<a href="#"><u>clear()</u></a>
void	<a href="#"><u>reset()</u></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

<code>public</code>	<a href="#"><code>StreamItem(String name, String baseClass, String playClass)</code></a> Create a new streamItem
---------------------	---

### Method Summary

<code>void</code>	<a href="#"><code>clearProperty(String name)</code></a> Clear property.
<code>String</code>	<a href="#"><code>getBaseClass()</code></a> Get base class path.
<code>String</code>	<a href="#"><code>getDescription()</code></a> Get streamType description.
<code>String</code>	<a href="#"><code>getName()</code></a> Get streamType name.
<code>String</code>	<a href="#"><code>getPlayClass()</code></a> Get play class path.
<a href="#"><code>WMSProperties</code></a>	<a href="#"><code>getProperties()</code></a> Get properties.
<code>String</code>	<a href="#"><code>getProperty(String name)</code></a> Get property by name.
<code>void</code>	<a href="#"><code>setBaseClass(String baseClass)</code></a> Set base class path.
<code>void</code>	<a href="#"><code>setDescription(String description)</code></a> Set streamType description.
<code>void</code>	<a href="#"><code>setName(String name)</code></a> Set streamType name.
<code>void</code>	<a href="#"><code>setPlayClass(String playClass)</code></a> Set play class path.
<code>void</code>	<a href="#"><code>setProperty(String name, String value)</code></a> 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

## 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.
<a href="#">java.util.Map</a>	<a href="#">getStreamDefs()</a> Get Map of streamItems (by streamType names).
<a href="#">java.util.List</a>	<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
--

### 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

<code>public</code>	<code><a href="#">ThreadPool(IVHost vhost, String name)</a></code>
	Create a new thread pool attached to a vHost.

#### Method Summary

<code>void</code>	<code><a href="#">execute(Runnable command)</a></code> Execute a runnable object.
<code>int</code>	<code><a href="#">getActiveCount()</a></code> Get number of active threads.
<code>java.util.concurrent.Executor</code>	<code><a href="#">getExecutor()</a></code> Get the underlying Executor pool.
<code>int</code>	<code><a href="#">getQueueSize()</a></code> Get the number of command objects in the LinkedBlockingQueue.
<code>void</code>	<code><a href="#">init(int corePoolSize)</a></code> Initialize threadPool.
<code>void</code>	<code><a href="#">terminate()</a></code> Terminate all threads and cleanup threadPool.

#### Methods inherited from class java.lang.Object

<code>clone</code> , <code>equals</code> , <code>finalize</code> , <code>getClass</code> , <code>hashCode</code> , <code>notify</code> , <code>notifyAll</code> , <code>toString</code> , <code>wait</code> , <code>wait</code>
---

#### 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

<code>public</code>	<a href="#"><code>VHostItem()</code></a> Create empty vHostItem
---------------------	--

### Method Summary

<code>String</code>	<a href="#"><code>getConfigDir()</code></a> Get configuration path for vHost.
<code>int</code>	<a href="#"><code>getConnectionLimit()</code></a> Get connection limit of this vHost item.
<code>String</code>	<a href="#"><code>getName()</code></a> Get vHost name
<code>WMSProperties</code>	<a href="#"><code>getProperties()</code></a> Get properties
<code>boolean</code>	<a href="#"><code>isVisited()</code></a> Has this vHostItem been visited during load of VHhosts.xml file.
<code>void</code>	<a href="#"><code>reset()</code></a> Reset vHostItem to empty state
<code>void</code>	<a href="#"><code>setConfigDir(String configDir)</code></a> Set configuration path for vHost.
<code>void</code>	<a href="#"><code>setConnectionLimit(int connectionLimit)</code></a> Set connection limit of this vHost item.
<code>void</code>	<a href="#"><code>setName(String name)</code></a> Set vHost name
<code>void</code>	<a href="#"><code>setProperties(WMSProperties properties)</code></a> Set properties
<code>void</code>	<a href="#"><code>setVisited(boolean visited)</code></a> Set is visited.
<code>String</code>	<a href="#"><code>toString()</code></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 VHhosts.xml file. Internally used to track and delete vHost definitions on reload of VHhosts.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="#">VHostList()</a>
	Create empty vHostList

### Method Summary

void	<a href="#">addVHostItemListener(IVHostItemNotify vHostItemListener)</a>
	Add vHostItem listener.
java.util.List	<a href="#">getVHostItems()</a>
	Get a list of vHostItems.
java.util.Map	<a href="#">getVHostMap()</a>
	Get the Map of vHostItem definitions.
java.util.List	<a href="#">getVHostNames()</a>
	Get a list of vHost names.
void	<a href="#">loadConfig()</a>
void	<a href="#">notifyVHostItemCreate(VHostItem vhostItem)</a>
	Notify vHostItem listener of item create.
void	<a href="#">notifyVHostItemDestroy(VHostItem vhostItem)</a>
	Notify vHostItem listener of item destroy.
void	<a href="#">notifyVHostItemUpdate(VHostItem vhostItem)</a>
	Notify vHostItem listener of item update.
void	<a href="#">reloadConfig()</a>
	Reload VHosts.xml file.
void	<a href="#">removeVHostItemListener(IVHostItemNotify vHostItemListener)</a>
	Remove vHostItem listener.

### Methods inherited from class java.lang.Object

<code>clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait</code>	
---	--

## 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(IVHostNotify vhostListener)</a> Add a vHost listener.
static <a href="#">IVHost</a>	<a href="#">getInstance(String vhostName)</a> Get vHost by name.
static <a href="#">IVHost</a>	<a href="#">getInstance(String vhostName, boolean doCreate)</a> Get vHost by name.
static java.util.List	<a href="#">getVHostNames()</a> Get a list of vHost names.
static void	<a href="#">init(String vhostName, String configHome)</a> Initialize vhost by name.
static void	<a href="#">notifyVHostClientConnect(IVHost vhost, IClient inClient, com.wowza.wms.request.RequestFunction function, AMFDataList params)</a> Notify vhost client connect
static void	<a href="#">notifyVHostCreate(IVHost vhost)</a> Notify vHost listener of create.
static void	<a href="#">notifyVHostInit(IVHost vhost)</a> Notify vHost listener of ini.
static void	<a href="#">notifyVHostShutdownComplete(IVHost vhost)</a> Notify vHost listener of destruction.
static void	<a href="#">notifyVHostShutdownStart(IVHost vhost)</a> Notify vHost listener of destruction.
static void	<a href="#">remove(String vhostName)</a> Remove a vHost from list of running vHosts.
static void	<a href="#">removeApplicationListener(IVHostNotify vhostListener)</a> Remove vHost listener

static void	<a href="#"><u>setServer(IIServer server)</u></a> Set a reference to the current running server.
static void	<a href="#"><u>shutdown(String vhostName)</u></a> Shutdown a vHost by name.

**Methods inherited from class** java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, 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 707  
 ABSOLUTE\_TIME\_PATTERN 707  
 acceptConnection 341, 358, 359  
 ACCEPTORS\_ACCEPTOR 1735  
 ACCEPTORS\_HANDLERADAPTER 1735  
 acceptSession 612, 1221  
 acquire 893, 900, 901, 902, 1336, 1360  
 acquireAndRelease 893, 901  
 acquireMediaCaster 907  
 acquireSocketAddress 1189  
 acquireUDPPortPair 1189, 1190  
 actionListeners 1219  
 activateOptions 716  
 add 81, 142, 143, 144, 177, 178, 179, 1175, 1709  
 addAcceptConnectionAttribute 373  
 addAcceptorListener 1740  
 addActionListener 1230  
 addApplicationInstanceListener 243  
 addApplicationListener 1740  
 addAudioCodecConfigPacket 1437  
 addAudioData 1422, 1564, 1565  
 addAudioDataInc 1564  
 addAudioExtraData 1407  
 addAuthQueryString 1048, 1112, 1119  
 addBody 1172  
 addBouncyCastleSecurityProvider 104  
 addBytes 1172  
 addCallbackListener 1445  
 addChunk 210  
 addClient 1336, 1360  
 addClientListener 274, 1428, 1429  
 addConnectionListener 340  
 addConnectMetaData 1103  
 addContentDescriptor 809  
 addData 222  
 addDataData 1422, 1565  
 addDataDataInc 1566  
 addDataEx 225  
 addDataExtraData 1407  
 addDifference 81  
 addDvrChunkListener 534

addDvrRecorderListener 311  
 addDvrRecordingListener 523  
 addDvrStoreListener 522  
 addDvrStreamManagerListener 311  
 addFirstABSTRequest 645  
 addHTTPListener 314  
 addHttpProvider 1706  
 addHTTPStreamerAdapterID 1707  
 addHTTPStreamerSession 299  
 addIdleWorkerListener 1747  
 addIOPerformance 608, 1227  
 addIOPerformance2 608, 1227  
 addItem 1551  
 addLicense 1511  
 addListener 662, 676, 1088, 1584  
 addLivePacketListener 1446  
 addLiveStreamPacketizerListener 307  
 addLiveStreamTranscoderListener 308  
 addManifestEntries 522, 533  
 addMediaCasterListener 277  
 addMediaReaderListener 304  
 addMediaStreamListener 275, 1508  
 addMediaWriterListener 302  
 addMember 204  
 addMetaDataListener 915  
 addModuleListener 281  
 addNameGroup 1509  
 addObject 159  
 addOverlay 1671  
 addOverlayImage 1670  
 addPendingWrites 841  
 addPlayReadyDecryptorKey 1489  
 addPlayStreamByName 279  
 addProperties 829  
 addPublisher 296  
 addPushPublishSession 316  
 addPushPublishSessionListener 316  
 addQueryStrRemoveList 593  
 addRange 412  
 addRendition 937  
 addRepeaterHeartBeatItem 525  
 addRTPIIncomingDatagramPortAll 299  
 addRTPIIncomingDatagramPortRange 299  
 addRTPListener 314  
 addRTPSession 291

addRTSPStream 1222  
addSegment 921  
addServerListener 1297, 1319  
addSession 1082, 1241  
addSessionListener 1241  
addSharedObjectListener 276, 1348  
addSlotListener 1340, 1364  
addSorterPacket 1678  
addStartupStream 1753  
addStreamAttribute 1261  
addStreamDomainStr 616  
addStreamDomainStrs 616  
addStreamInfo 1261  
addStreamSrcToMediaCaster 903  
addString 158  
addToCache 425  
addToChunk 490, 630  
addToEnvMap 830  
addToManifest 477  
addToPlaylist 1581, 1582  
addTrack 1257  
addTrackId 1259  
addTrackInternal 1258  
addTrailingZero 813  
addTrait 159  
addUserHTTPHeaders 620  
addVHostItemListener 1777  
addVHostListener 1781  
addVideoCodecConfigPacket 1437  
addVideoData 1421, 1562, 1563  
addVideoDataInc 1564  
addVideoExtraData 1407  
addVideoH264SEILListener 1445  
adjustFirstPacketTCs 67  
akamaiOriginIp 1076  
akamaiOriginPort 1076  
akamaiPassword 1075  
akamaiUserName 1075  
alphas 32  
AMF3Utils 119  
AMF\_LEVEL0 130  
AMF\_LEVEL3 130  
AMF\_TEXTTYPE\_ON\_TEXT\_DATA 1640  
AMFData 131  
AMFdataArray 140, 141  
AMFByteArray 151, 152  
AMFDataContextDeserialize 157  
AMFDataContextSerialize 162  
AMFDataItem 168, 169, 170  
AMFDataList 176, 177  
AMFDataMixedArray 188, 189  
AMFDataObj 196, 197  
AMFDataTrait 204  
AMFDEBUGHEADERSIZE 209  
AMFObj 209  
AMFObjChunk 216  
AMFPacket 220  
AMFUtils 13  
announce 1265  
append 717  
APPEND\_FILE 686  
appendParamsToUrl 109  
appInstance 1004, 1027, 1217, 1501  
appInstanceShutdown 1217  
APPLICATION\_KEY 706  
applicationExists 1748  
ARCHIVE\_STRATEGY\_APPEND 443  
ARCHIVE\_STRATEGY\_DELETE 443  
ARCHIVE\_STRATEGY\_VERSION 443  
artifact 393, 415  
assembleQueryStr 74  
attachToWMSSession 1265  
audioCodecId 928  
audioCodecStringToId 59  
audioCodecToMetaDataString 60  
audioCodecToString 60  
audioCodecTypeToString 94  
audioHost 1153  
audioPacketizers 1220  
audioPort 1153  
AUDIOSAMPLE\_ACCESS\_ALL 354  
AUDIOSAMPLE\_ACCESS\_NONE 354  
AUDIOSAMPLEACCESS 1405  
authenticateHandler 546  
authenticateHTTPProvider 331  
authenticateHTTPProviderHandler 546  
authenticatePlayHandler 1217  
authenticatePublishHandler 1217  
authenticateRTSP 332  
authenticateSIP 333

AuthenticateUsernamePasswordProviderBase 327  
 authenticationMethod 546  
 AUTHMETHOD\_PLAY 1216  
 AUTHMETHOD\_PUBLISH 1216  
 AUTHMETHOD\_UNKNOWN 1216  
 authMode 1111, 1118  
 AVSYNCMETHODS\_RTPTIMECODE 1255  
 AVSYNCMETHODS\_SENDERREPORT 1255  
 AVSYNCMETHODS\_SYSTEMCLOCK 1255  
 AVSYNCMETHODS\_UNKNOWN 1255  
 avSyncNameToId 1271

**B**

badStreams 1005  
 bind 1202  
 bitrateAudio 928  
 bitrateTotal 927  
 bitrateVideo 928  
 blockSize 840  
 booleanValue 171  
 broadcastPlayMessage 1507  
 broadcastGetObjectEncoding 1507  
 broadcastMsg 278  
 buffer 216  
 BufferUtils 32  
 byteArrayToFile 44  
 byteArrayToInt 34, 35  
 byteArrayToIntSafeSync 818  
 byteArrayToLong 33, 34  
 byteArrayToLongSafeSync 818  
 byteArrayToShort 35, 36  
 byteArrayToString 33  
 bytesAvg 1079  
 byteStringLen 787  
 byteValue 171

**C**

cacheRTPStream 1190  
 calcBitrate 492  
 calcParity 1628  
 calcTotalPacketSize 225  
 calculateChunkGroupTime 424  
 call 361, 362

CALLBACK\_PARAM1 968  
 CALLBACK\_PARAM10 969  
 CALLBACK\_PARAM2 968  
 CALLBACK\_PARAM3 968  
 CALLBACK\_PARAM4 968  
 CALLBACK\_PARAM5 968  
 CALLBACK\_PARAM6 968  
 CALLBACK\_PARAM7 968  
 CALLBACK\_PARAM8 969  
 CALLBACK\_PARAM9 969  
 callConnect 1084, 1146  
 canHandle 547, 553, 586, 848, 1288  
 canPlay 530  
 canRecord 520, 530  
 canRecordAudio 490, 1519  
 canRecordData 490, 1520  
 canRecordVideo 490, 1520  
 CAT\_ALL 759  
 CAT\_application 748  
 CAT\_cupertino 748  
 CAT\_dvrchunk 749  
 CAT\_mpegdash 749  
 CAT\_rtsp 748  
 CAT\_sanjose 748  
 CAT\_server 747  
 CAT\_session 748  
 CAT\_smoothstreaming 748  
 CAT\_stream 748  
 CAT\_transcoder 749  
 CAT\_vhost 748  
 CAT\_webm 749  
 CEA608\_EXTENDED\_FR\_SP\_MAP 1627  
 CEA608\_EXTENDED\_NA\_MAP 1627  
 CEA608\_EXTENDED\_PORT\_GERMAN\_DANISH\_MAP  
 1628  
 CEA608\_MUSICAL\_NOTE 1627  
 CEA608\_SOLID\_BLOCK 1627  
 cea608CharToCharUTF8 1628  
 cea608ExtendedCharacterToUTF8 1629  
 challenge 1111  
 CHANNEL1 1610  
 CHANNEL2 1610  
 charUTF8ToCAE608 1628  
 checkAndSetPlayLogged 607  
 checkParity 1628, 1629

checkSendMetadata 1266  
CHUNK\_FILENAME\_FORMAT\_AUDIO 501  
CHUNK\_FILENAME\_FORMAT\_DATA 501  
CHUNK\_FILENAME\_FORMAT\_METADATA 501  
CHUNK\_FILENAME\_FORMAT\_VIDEO 501  
CHUNK\_FOLDER\_FORMAT 502  
CLASS 1026, 1036, 1072, 1107, 1111, 1118, 1135, 1145, 1152, 1161  
CLASSNAME 1026, 1037, 1072, 1107, 1111, 1118, 1135, 1145, 1152, 1161  
cleanUp 716  
clear 81, 764, 1176, 1341, 1359, 1435, 1763  
clearByteContainer 213  
clearFastPlaySettings 364, 1425  
clearIntData 158  
clearLoggingValues 568, 610, 637, 644, 651, 1230, 1434  
clearOverlay 1671  
clearProperty 881, 1766  
clearRTSPSessionExtraLines 1283  
clearSegments 922, 937  
clearStreamName 1504, 1505  
clearStreamSrcToMediaCaster 903  
clearTracks 1258  
client 327  
clone 81, 204, 221, 238, 472, 1373  
cloneProperties 321  
close 9, 29, 716, 824, 840, 846, 1336, 1365, 1377, 1384, 1424, 1471, 1538, 1546, 1566, 1571, 1575, 1583  
closeAndWait 1583  
ClosedCaptionCEA608Utils 388, 1628  
closeHostPort 1751  
closeQuietly 51  
closeStream 988  
CODEC\_AUDIO\_AAC 1731  
CODEC\_AUDIO\_AC3 1730  
CODEC\_AUDIO\_G711\_ALAW 1731  
CODEC\_AUDIO\_G711\_MULAW 1731  
CODEC\_AUDIO\_MP3 1730  
CODEC\_AUDIO\_MP3\_8 1732  
CODEC\_AUDIO\_NELLYMOSER 1731  
CODEC\_AUDIO\_NELLYMOSER\_16MONO 1731  
CODEC\_AUDIO\_NELLYMOSER\_8MONO 1731  
CODEC\_AUDIO\_PCM\_BE 1730  
CODEC\_AUDIO\_PCM\_LE 1730  
CODEC\_AUDIO\_PCM\_SWF 1730  
CODEC\_AUDIO\_RESERVED 1731  
CODEC\_AUDIO\_SPEEX 1731  
CODEC\_AUDIO\_UNKNOWN 1730  
CODEC\_AUDIO\_VORBIS 1731  
CODEC\_STREAM\_MP2T 1732  
CODEC\_STREAM\_UNKNOWN 1732  
CODEC\_TYPE 476  
CODEC\_VIDEO\_H263 1729  
CODEC\_VIDEO\_H264 1729  
CODEC\_VIDEO\_H265 1730  
CODEC\_VIDEO\_MPEG2 1730  
CODEC\_VIDEO\_MPEG4 1729  
CODEC\_VIDEO\_SCREEN 1729  
CODEC\_VIDEO\_SCREEN2 1729  
CODEC\_VIDEO\_SPARK 1728  
CODEC\_VIDEO\_UNKNOWN 1728  
CODEC\_VIDEO\_VP6 1729  
CODEC\_VIDEO\_VP6A 1729  
CODEC\_VIDEO\_VP8 1729  
CODECSTR\_FORMAT\_VIDEO\_POSTIOS4 94  
CODECSTR\_FORMAT\_VIDEO\_PREIOS4 94  
CODECSTR\_FORMAT\_VIDEO\_UNKNOWN 93  
CODES\_HREF 707  
COL0\_BLUE 1609  
COL0\_CYAN 1610  
COL0\_GREEN 1609  
COL0\_HIBYTE 1615  
COL0\_LOWBYTE 1616  
COL0\_MAGENTA 1610  
COL0\_RED 1610  
COL0\_WHITE 1609  
COL0\_YELLOW 1610  
COL\_0 1616  
COL\_12 1616  
COL\_16 1616  
COL\_20 1616  
COL\_24 1616  
COL\_28 1617  
COL\_4 1616  
COL\_5 1616  
COLN\_LOWBYTE 1617  
compareTo 1697  
compress 154  
conce 1118  
CONFIGURATOR\_CLASS\_KEY 706

configureSocketAcceptor 1706  
connect 1014, 1084, 1154  
connect\_Out 1136  
ConnectionCounter 340  
connectionFlashVersion 1075  
connectionHolder 1218  
connectionPageURL 1075  
connectionQueryStr 1075  
connectionSwfURL 1075  
connectionTimeout 1076  
connectLastAttempt 1076  
connectLastSuccess 1076  
connectMetaData 1079  
containsDvrRecorder 295  
containsExtraDataKey 227  
containsHeader 562  
containsHTTPStreamer 294  
containsIndex 567, 636, 643, 650  
containsKey 197, 229  
containsLiveStreamPacketizer 295  
containsLiveStreamTranscoder 309  
containsProperty 1341, 1361  
containsSlot 1342, 1361  
containsStreamDomainStr 616  
containsStreamNameParts 614  
containsTime 542  
CONTENTTYPE\_ACKBANDWIDTH 1733  
CONTENTTYPE\_AUDIO 1733  
CONTENTTYPE\_BUFFERSIZE 1733  
CONTENTTYPE\_CHORD 806  
CONTENTTYPE\_DATA 1733  
CONTENTTYPE\_DATA0 1733  
CONTENTTYPE\_DATA3 1734  
CONTENTTYPE\_EVENTS 806  
CONTENTTYPE\_FUNCTION 1734  
CONTENTTYPE\_FUNCTION0 1734  
CONTENTTYPE\_FUNCTION3 1734  
CONTENTTYPE\_LYRICS 806  
CONTENTTYPE\_MEDIA 1381  
CONTENTTYPE\_MEDIACHUNK 1734  
CONTENTTYPE\_MEDIALIST 1381  
CONTENTTYPE\_MOVEMENT 806  
CONTENTTYPE\_OTHER 806  
CONTENTTYPE\_PLAYCALLBACK 1734  
CONTENTTYPE\_SETBANDWIDTH 1733  
CONTENTTYPE\_SETCUNKSIZE 1732  
CONTENTTYPE\_SHAREDOBJECTS 1734  
CONTENTTYPE\_SHAREDOBJECTS0 1734  
CONTENTTYPE\_SHAREDOBJECTS3 1734  
CONTENTTYPE\_TRANSCRPTION 806  
CONTENTTYPE\_TRIVIA 807  
CONTENTTYPE\_UNKNOWN 1732  
CONTENTTYPE\_URLIMAGES 807  
CONTENTTYPE\_URLWEBPAGES 807  
CONTENTTYPE\_VIDEO 1733  
CONTENTTYPE\_WATCHDOG 1733  
CONTROLCODES\_AOF 1611  
CONTROLCODES\_AON 1611  
CONTROLCODES\_BAO 1614  
CONTROLCODES\_BAS 1614  
CONTROLCODES\_BBO 1613  
CONTROLCODES\_BBS 1613  
CONTROLCODES\_BCO 1613  
CONTROLCODES\_BCS 1613  
CONTROLCODES\_BGO 1613  
CONTROLCODES\_BGS 1613  
CONTROLCODES\_BMO 1614  
CONTROLCODES\_BMS 1614  
CONTROLCODES\_BRO 1613  
CONTROLCODES\_BRS 1613  
CONTROLCODES\_BS 1611  
CONTROLCODES\_BT 1614  
CONTROLCODES\_BWO 1612  
CONTROLCODES\_BWS 1613  
CONTROLCODES\_BYO 1614  
CONTROLCODES\_BYS 1614  
CONTROLCODES\_CR 1612  
CONTROLCODES\_DER 1611  
CONTROLCODES\_EDM 1612  
CONTROLCODES\_ENM 1612  
CONTROLCODES\_EOC 1612  
CONTROLCODES\_FA 1614  
CONTROLCODES\_FAU 1614  
CONTROLCODES\_FON 1611  
CONTROLCODES\_RCL 1610  
CONTROLCODES\_RDC 1611  
CONTROLCODES\_RTD 1612  
CONTROLCODES\_RU2 1611  
CONTROLCODES\_RU3 1611  
CONTROLCODES\_RU4 1611

CONTROLCODES\_TO1 1612  
CONTROLCODES\_TO2 1612  
CONTROLCODES\_TO3 1612  
CONTROLCODES\_TR 1612  
convertDataEvenToAMFData 1098  
convertOnMetadataToSetDataFrame 1098  
convertParams 13  
COOKIEDATEFORMAT 73  
cookieStr 1217  
copyFile 50  
copyFile2 50  
COUNT\_COLS 1615  
COUNT\_ROWS 1615  
COUNTER\_HTTPCUPERTINO 1728  
COUNTER\_HTTPDVRCHUNKS 1728  
COUNTER\_HTTPMPPEGDASH 1728  
COUNTER\_HTTPSANJOSE 1728  
COUNTER\_HTTPSMOOTH 1728  
COUNTER\_HTTPWEBM 1728  
COUNTER\_RTMP 1727  
COUNTER\_RTP 1727  
COUNTER\_TOTAL 1728  
createAkamaiStreamName 1109  
createBroadcastMessage 1171  
createConnectMessage 1171  
createContextDeserialize 134  
createContextSerialize 134  
createDefaultMessage 1171  
createDvrAudioChunk 487  
createDvrDataChunk 488  
createDvrOnMetadataChunk 488  
createDvrVideoChunk 487  
createEnhancedSeekMessage 1172  
createHTTPSessionInstance 583  
createIndexLive 637, 644  
createInstance 827, 1561, 1580  
createPlayStatusMessage 1171  
createPushPublishSession 1016, 1034, 1103, 1159  
createRecorder 668  
createSeekMessage 1171  
createSOMessage 1172  
createStream 986, 1262, 1562  
CrontabEvent 1694  
CTRL\_playlist\_node 753  
CURRENTFLASHVERSION 1072  
CURRENTFMLEVERSION 1072  
  
D  
  
data 415  
DATA\_TYPE\_AMF3 128  
DATA\_TYPE\_AMF3\_ARRAY 130  
DATA\_TYPE\_AMF3\_BOOLEAN\_FALSE 129  
DATA\_TYPE\_AMF3\_BOOLEAN\_TRUE 129  
DATA\_TYPE\_AMF3\_BYTEARRAY 130  
DATA\_TYPE\_AMF3\_DATE 130  
DATA\_TYPE\_AMF3\_INTEGER 129  
DATA\_TYPE\_AMF3\_NULL 129  
DATA\_TYPE\_AMF3\_NUMBER 129  
DATA\_TYPE\_AMF3\_OBJECT 130  
DATA\_TYPE\_AMF3\_STRING 129  
DATA\_TYPE\_AMF3\_UNDEFINED 129  
DATA\_TYPE\_AMF3\_XML\_LEGACY 130  
DATA\_TYPE\_AMF3\_XML\_TOP 130  
DATA\_TYPE\_ARRAY 127  
DATA\_TYPE\_AS\_OBJECT 128  
DATA\_TYPE\_BOOLEAN 126  
DATA\_TYPE\_BYTEARRAY 128  
DATA\_TYPE\_CUSTOM\_CLASS 128  
DATA\_TYPE\_DATE 127  
DATA\_TYPE\_INTEGER 128  
DATA\_TYPE\_LONG\_STRING 128  
DATA\_TYPE\_MIXED\_ARRAY 127  
DATA\_TYPE\_MOVIE\_CLIP 127  
DATA\_TYPE\_NULL 127  
DATA\_TYPE\_NUMBER 126  
DATA\_TYPE\_OBJECT 127  
DATA\_TYPE\_OBJECT\_END 127  
DATA\_TYPE\_RECORDSET 128  
DATA\_TYPE\_REFERENCE\_OBJECT 127  
DATA\_TYPE\_STRING 126  
DATA\_TYPE\_UNDEFINED 127  
DATA\_TYPE\_UNKNOWN 126  
DATA\_TYPE\_XML 128  
DATA\_TYPE\_XML\_TOP 129  
DATE\_AND\_TIME\_FORMAT 707  
DATE\_AND\_TIME\_PATTERN 707  
DATEFORMAT 168  
dateValue 171  
DAY 1694

debug 727, 728  
 debugLogLastPacketCount 1073  
 debugPackets 1073  
 debugRTSPSession 1220  
 DebugUtils 41  
 debugWOWZProtocol 1315  
 DECODE 18  
 decode 21  
 DECODE\_OBJ\_REF 195  
 DECODE\_TRAITS 196  
 DECODE\_TRAITS\_EXT 195  
 DECODE\_TRAITS\_REF 195  
 DECODE\_UNDEFINED 195  
 decodeToFile 23  
 decodeFromFile 22  
 decodeHexString 33  
 decodeRangeHeader 1681  
 decodeS 1324  
 decodeSS 1324  
 decodeStorageDir 294, 1316  
 decodeStreamInfo 1684, 1685  
 decodeString 23  
 decodeToFile 22  
 decodeToObject 21  
 decodeValue 109  
 decompress 155  
 decPendingWriteCount 841  
 decrement 342  
 DEFAULT\_AMF\_CONVERTER\_TEXT\_TYPE 1650  
 DEFAULT\_AMF\_CONVERTER\_TRACK\_INDEX 1651  
 DEFAULT\_APPINSTANCE\_NAME 270  
 DEFAULT\_APPLICATION\_NAME 241  
 DEFAULT\_CAPTION\_CHARACTER\_SET 1648  
 DEFAULT\_CHUNK\_MEMORY\_CACHESIZE 451  
 DEFAULT\_CONFIGURATION\_FILE 706  
 DEFAULT\_CONFIGURATION\_KEY 706  
 DEFAULT\_CUPERTINO\_PLAYLIST\_GZIP\_THRESHOLD 464  
 DEFAULT\_MAXIMUM\_CAPTION\_DURATION 1647  
 DEFAULT\_PLAYLIST\_REQUEST\_DELEGATE 460  
 DEFAULT\_PORT 716  
 DEFAULT\_PROPERTY\_ALLOWABLE\_AV\_PACKET\_DELTA 447  
 DEFAULT\_PROPERTY\_APPEND\_DISCONTINUITY\_DELTA 441  
 DEFAULT\_PROPERTY\_AUDIO\_GROUP\_COUNT 500  
 DEFAULT\_PROPERTY\_AUDIO\_ONLY\_CHUNK\_TARGET\_DURATION 449  
 DEFAULT\_PROPERTY\_BREAK\_ON PTS 449  
 DEFAULT\_PROPERTY\_CAPION\_LANGUAGES 1642  
 DEFAULT\_PROPERTY\_CAPTION\_FILE\_NAMING\_RULE 1646  
 DEFAULT\_PROPERTY\_CAPTION\_PATH\_NAMING\_RULE 1647  
 DEFAULT\_PROPERTY\_CHUNK\_CACHE\_CLASS 445  
 DEFAULT\_PROPERTY\_CHUNK\_DURATION\_MINIMUM 450  
 DEFAULT\_PROPERTY\_CHUNK\_GROUPING\_SECONDS 441  
 DEFAULT\_PROPERTY\_CHUNK\_READER\_CLASS 443  
 DEFAULT\_PROPERTY\_CHUNK\_WRITER\_CLASS 444  
 DEFAULT\_PROPERTY\_DEBUG\_MAX\_INVALID\_CHUNKS\_LOGGED 454  
 DEFAULT\_PROPERTY\_DEBUG\_MAX\_RAW\_PACKETS 456  
 DEFAULT\_PROPERTY\_DEBUG\_MAX\_VALID\_CHUNKS\_LOGGED 455  
 DEFAULT\_PROPERTY\_DEBUG\_RAW\_PACKETS 456  
 DEFAULT\_PROPERTY\_DVR\_MAX\_CHUNK\_LOG 457  
 DEFAULT\_PROPERTY\_FILE\_SYSTEM\_CLASS 444  
 DEFAULT\_PROPERTY\_MANIFEST\_PERSISTER\_CLASS 444  
 DEFAULT\_PROPERTY\_MANIFEST\_PURGE\_SIZE 497  
 DEFAULT\_PROPERTY\_MAX\_ALLOWABLE\_CHUNK\_DURATION 498  
 DEFAULT\_PROPERTY\_MAX\_RECALC\_DURATION\_LOG 457  
 DEFAULT\_PROPERTY\_MAX\_RECORDING\_LENGTH 453  
 DEFAULT\_PROPERTY\_MAX\_SAVE HOLDER\_SIZE 499  
 DEFAULT\_PROPERTY\_MBR\_ALTERNATIVE\_MATCH\_DELTA 451  
 DEFAULT\_PROPERTY\_MBR\_MINIMUM\_PACKETTIME\_GAP\_SIZE 452  
 DEFAULT\_PROPERTY\_MBR\_MINIMUM\_UTCTIME\_GAP\_SIZE 452  
 DEFAULT\_PROPERTY\_MEDIACACHE\_READER\_CLASS 453  
 DEFAULT\_PROPERTY\_MINIMUM\_AVAILABLE\_CHUNKS 497

DEFAULT\_PROPERTY\_PACKET\_DELTA\_TO\_NOTIFY 448  
DEFAULT\_PROPERTY\_PACKET\_DELTA\_TO\_RESET\_TIMER 447  
DEFAULT\_PROPERTY\_PACKET\_DURATION\_MAXIMUM 452  
DEFAULT\_PROPERTY\_PACKET\_SORT\_TIME 449  
DEFAULT\_PROPERTY\_PURGE\_CONTROL\_CLASS 498  
DEFAULT\_PROPERTY\_RECALC\_DURATION\_MAX\_AV\_DIFFERENCE\_TRIGGER 499  
DEFAULT\_PROPERTY\_RECALC\_DURATION\_TRIGGER\_SIZE 499  
DEFAULT\_PROPERTY\_RECORDINGS\_LOADER\_CLASS 445  
DEFAULT\_PROPERTY\_REPEATER\_HEARTBEAT\_DURATION 450  
DEFAULT\_PROPERTY\_SANJOSE\_ABST\_DURATION\_TOLERANCE 463  
DEFAULT\_PROPERTY\_SANJOSE\_ABST\_TIMESCALE 463  
DEFAULT\_PROPERTY\_SMOOTH\_MANIFEST\_LIVE\_CAN\_PAUSE 465  
DEFAULT\_PROPERTY\_SMOOTH\_MANIFEST\_LIVE\_CAN\_SEEK 465  
DEFAULT\_PROPERTY\_SMOOTH\_MANIFEST\_MAJOR\_VERSION 466  
DEFAULT\_PROPERTY\_SMOOTH\_MANIFEST\_MINOR\_VERSION 466  
DEFAULT\_PROPERTY\_SMOOTH\_MANIFEST\_RECORD  
ED\_CAN\_PAUSE 465  
DEFAULT\_PROPERTY\_SMOOTH\_MANIFEST\_RECORD  
ED\_CAN\_SEEK 465  
DEFAULT\_PROPERTY\_SMOOTH\_MANIFEST\_RECORD  
ED\_SPECIFY\_DURATION 466  
DEFAULT\_PROPERTY\_STORAGE\_DIRECTORY 442  
DEFAULT\_PROPERTY\_STREAM\_IDLE\_TIMEOUT 500  
DEFAULT\_PROPERTY\_STREAM\_STARTUP\_TIMEOUT 501  
DEFAULT\_PROPERTY\_TEXT\_READER\_CLASS 446  
DEFAULT\_PROPERTY\_TEXT\_WRITER\_CLASS 446  
DEFAULT\_PROPERTY\_UTCAUTHORITY 496  
DEFAULT\_PROPERTY\_UTC\_USE\_WOWZA\_PACKET\_AR  
RIVAL 496  
DEFAULT\_PROPERTY\_WAIT\_FOR\_CODEC\_TIME 500  
DEFAULT\_RANDOMACCESSREADER 1381  
DEFAULT\_RANDOMACCESSREADEROPTIMIZER 1381  
DEFAULT\_REPOSITORY\_NAME 705  
DEFAULT RTPIGNOREPROFILELEVELID 1257  
DEFAULT RTPIGNORERESPROPARPARAMETERSETS 1257  
DEFAULT\_RTPSESSIONNAME 1257  
DEFAULT\_XML\_CONFIGURATION\_FILE 706  
DefaultDvrStreamVersionHandler 390  
defaultLocale 103  
defaultTimeZone 103  
deleteArchivedStore 521  
deleteDirectory 51  
deleteFolder 51  
deleteSlot 1337, 1361  
deleteStream 987  
describe 1274, 1275  
deserialize 136, 148, 154, 171, 183, 189, 201, 237, 407, 410, 412, 486  
deserializeBody 771, 791, 794, 797, 800, 802, 808, 813, 816, 821  
deserializeDate 119  
deserializeFrame 788  
deserializeInnerObject 134  
deserializeInt 119  
deserializePacket 226  
deserializeString 120, 788  
destroy 1677  
destroyInstance 827  
detachFromWMSession 1265  
determineChunkGroupIdentifier 424  
determineExistingStoreForPlaying 391, 537  
determineExistingStoreForRecording 391, 536  
determineStreamVersion 492  
determineUtcTime 427  
directOutput 1688  
disconnect 342, 1014, 1098, 1154, 1339, 1348, 1364  
doBreak 1169  
doCRC32 38  
doesStackContainMethod 45  
doHTTPAuthentication 548  
doIdle 910, 1238  
DONOTHING\_BYTE 1610  
DONOTHING\_SHORT 1610  
DONT\_BREAK\_LINES 18  
doSendOnMetaData 1076  
doSessionRedirect 620  
doSet 82

doubleValue 170  
 doWatchdog 867, 892, 899, 1038, 1187, 1317, 1511  
 dstAppInstanceName 1073  
 dstApplicationName 1073  
 dstStreamName 1027, 1073  
 DT\_FORMAT 1694  
 dummy 87  
 dump 1697  
 DVR\_DEFAULT\_FILESTORE 440  
 DVR\_DEFAULT\_RECORDER\_ID 440  
 DVR\_REPEAT\_PACKETIZER\_ID 440  
 DVR\_STREAMING\_PACKETIZER\_ID 440  
 DVR\_TIME 468  
 DVR\_WINDOW\_DURATION\_UNLIMITED 442  
 DvrManifestChunkEntry 393  
 DvrManifestCodecEntry 397  
 DvrManifestEntry 404  
 DvrManifestEntryFactory 407  
 DvrManifestEntryRange 408  
 DvrManifestEntryRangeGroup 411  
 DvrManifestOnMetadataEntry 415  
 DvrManifestTimeMapEntry 418  
 dvrRecorders 1501  
 dvrStart 403  
 dvrStop 403  
 dvrStreamStorageDeleted 513  
 dvrStreamStorageLoaded 513  
 dvrStreamStoreCreate 512  
 dvrStreamStoreDestroy 512  
 dvrStreamStoreInit 512  
 dvrToPt 542  
 dvrToUtc 542

**E**

elapsedTime 1219  
 ElapsedTimer 47  
 ENCODE 17  
 encodeBytes 19, 20, 406  
 encodeToFile 23  
 encodeFromFile 22  
 encodeHexString 32  
 encodeObject 18, 19  
 encodeString 23  
 encodeToFile 22  
 encodeValue 109  
 ENCRYPTION\_TYPE\_CENC 471  
 ENCRYPTION\_TYPE\_CUPERTINO 470  
 ENCRYPTION\_TYPE\_PLAYREADY 470  
 encryptions 393  
 end 106  
 endChunk 490, 629  
 equals 101  
 error 725, 726  
 EVT\_ALL 760  
 EVT\_announce 752  
 EVT\_app\_start 752  
 EVT\_app\_stop 752  
 EVT\_comment 752  
 EVT\_connect 749  
 EVT\_connect\_burst 749  
 EVT\_connect\_pending 749  
 EVT\_create 750  
 EVT\_decoderaudiostart 752  
 EVT\_decoderaudiostop 752  
 EVT\_decodervideostart 752  
 EVT\_decodervideostop 753  
 EVT\_describe 752  
 EVT\_destroy 750  
 EVT\_disconnect 749  
 EVT\_encoderaudiostart 753  
 EVT\_encoderaudiostop 753  
 EVT\_encodervideostart 753  
 EVT\_encodervideostop 753  
 EVT\_pause 750  
 EVT\_play 750  
 EVT\_publish 750  
 EVT\_record 751  
 EVT\_recordstop 751  
 EVT\_seek 751  
 EVT\_server\_start 751  
 EVT\_server\_stop 751  
 EVT\_setbuffertime 750  
 EVT\_setstreamtype 750  
 EVT\_stop 751  
 EVT\_unpause 750  
 EVT\_unpublish 750  
 EVT\_vhost\_start 751  
 EVT\_vhost\_stop 751  
 execute 1771

exists 10, 843, 1347  
 existsRTSPTunnelingSession 1191  
 expandEndTime 423, 485  
 expandEnvironmentVariables 103, 104  
 expandStartTime 423, 485  
 expandToPortPair 1190  
 extractCodecConfigFromTrackInfo 1266  
 extractHTTPRequestInfo 618  
 extraDataClone 227

F

fastDateFormat 168  
 FastPlaySettings 1369  
 fatal 726, 727  
 FCPublish 992  
 FCSubscribe 993  
 fcSubscribe 379  
 FCUnPublish 992  
 FCUnpublish 993  
 FCUnSubscribe 994  
 FCUnsubscribe 994  
 fcUnSubscribe 379  
 fcUnSubscribeAll 379  
 FD\_ALL 759  
 FD\_c\_client\_id 745  
 FD\_c\_ip 744  
 FD\_c\_proto 744  
 FD\_c\_referrer 744  
 FD\_c\_user\_agent 744  
 FD\_cs\_bytes 745  
 FD\_cs\_bytes\_inc 712  
 FD\_cs\_stream\_bytes 745  
 FD\_cs\_stream\_bytes\_inc 712  
 FD\_cs\_uri\_query 746  
 FD\_cs\_uri\_stem 746  
 FD\_date 743  
 FD\_s\_ip 746  
 FD\_s\_port 747  
 FD\_s\_uri 744  
 FD\_sc\_bytes 745  
 FD\_sc\_bytes\_inc 712  
 FD\_sc\_stream\_bytes 746  
 FD\_sc\_stream\_bytes\_inc 712  
 FD\_time 743  
 FD\_tz 743  
 FD\_x\_app 744  
 FD\_x\_appinst 744  
 FD\_x\_category 743  
 FD\_x\_comment 747  
 FD\_x\_ctx 743  
 FD\_x\_ctx\_override 744  
 FD\_x\_duration 746  
 FD\_x\_duration\_inc 712  
 FD\_x\_event 743  
 FD\_x\_file\_ext 747  
 FD\_x\_file\_length 745  
 FD\_x\_file\_name 746  
 FD\_x\_file\_size 745  
 FD\_x\_severity 747  
 FD\_x\_sname 745  
 FD\_x\_sname\_query 746  
 FD\_x\_spos 745  
 FD\_x\_status 746  
 FD\_x\_stream\_id 747  
 FD\_x\_suri 747  
 FD\_x\_suri\_query 747  
 FD\_x\_suri\_stem 747  
 FD\_x\_vhost 744  
 FILEEXTENSION 1333  
 fileFormat 697  
 FILEFORMAT\_FLV 1732  
 FILEFORMAT\_MP4 1732  
 FILEFORMAT\_UNKNOWN 1732  
 fileTemplate 700  
 fileToByteArray 50  
 FileUtils 50  
 fileVersionDelegate 700  
 filterMetadata 1575  
 filterQueryStr 593  
 filters 546  
 FLAGS\_DEFAULT 787  
 floatValue 170  
 flush 1337, 1348, 1365, 1438, 1566  
 flushBase64 29  
 FLV\_CHUNKHEADER\_BUFFERSIZE 58  
 FLV\_CHUNKHEADER\_FIRSTBYTE 58  
 FLV\_CHUNKHEADER\_HEADERSIZE 58  
 FLV\_CHUNKHEADER\_ISIZE 57  
 FLV\_CHUNKHEADER\_ITIMECODE 57

FLV\_CHUNKHEADER\_ITYPE 57  
FLV\_CHUNKHEADER\_SECONDBYTE 58  
FLV\_CHUNKHEADER\_VALUESIZE 58  
FLV\_DFRAME 58  
FLV\_KFRAME 58  
FLV\_PFRAME 58  
FLV\_TCINDEXAUDIO 59  
FLV\_TCINDEXDATA 59  
FLV\_TCINDEXVIDEO 59  
FLV\_UFRAME 58  
FLVUtils 59  
forceNewLoggerInstance 731  
forceReset 867  
FORMAT\_COLOR 1617  
FORMAT\_FLASH 1617  
FORMAT\_FLV 685  
FORMAT\_ITALICS 1617  
FORMAT\_ITALICS\_UNDERLINE 1617  
FORMAT\_MP4 685  
formatBytes 42  
formatBytesInSingleRow 43  
formatBytesShort 41  
formatBytesStruct 43  
formatDeleteCookie 75  
formatH264CodecConfig 1685  
formatH264CodecConfigPacket 1685  
formatMilliseconds 44  
formatRTPInfo 1276  
formatSeconds 44  
formatSetCookie 76  
formatUtcTime 45  
FORWARD 8  
frameTypeToString 61  
fs 1324

## G

generateAuth 89, 90, 91  
generateAuthHTTP 90  
generateHash 92  
generateHashBytes 91, 92  
get 145, 181, 198, 231, 232, 1346, 1710  
getAbsTimecode 212, 223  
getAccess 1366, 1436  
getActiveCount 1771

getAdapterName 587  
getAddress 1703  
getAddressCount 1205  
getAddressRawStr 1704  
getAddressStr 1704  
getAdminAgent 1315  
getAdminGUID 1322  
getAdminInterfaceObjectList 1298, 1320  
getAkamaiOriginIp 1086  
getAkamaiOriginPort 1087  
getAkamaiPassword 1086  
getAkamaiSessionKey 1084  
getAkamaiUserName 1086  
getAllAsStrings 323  
getAllowDomains 284  
getAndClearNotifyCreate 623  
getAndClearPendingFrameGrabs 1668  
getAndClearPendingOverlays 1672  
getAndSetStartStream 1531  
getAppInstance 242, 355, 529, 591, 609, 669, 865, 975, 1031, 1094, 1155, 1225, 1263, 1507, 1518, 1539, 1562  
getAppInstanceName 1267, 1506  
getAppInstanceNames 244  
getAppInstanceProperty 1009  
getApplication 270, 356, 717, 975, 1737  
getApplicationFolderNames 1748  
getApplicationInstance 899, 1531  
getApplicationInstanceTouchTimeout 303  
getApplicationLock 1745  
getApplicationNames 1748  
getApplicationPath 242  
getApplicationProperty 1008  
getApplicationTimeout 281, 1743  
getAppName 1266, 1506  
getArchiveStrategy 391, 519, 537  
getArtifactsTextRepresentation 395, 416  
getAudioCodec 61, 69, 397  
getAudioCodecConfigPacket 1437  
getAudioCodecId 931  
getAudioHost 1156, 1196  
getAudioMissing 1421  
getAudioMP3Layer 69  
getAudioPacketizerItem 1187, 1235  
getAudioPort 1156, 1195  
getAudioSize 1406

getAudioTC 1409  
getAudioTrack 1260  
getAuthenticatePlayHandler 1224  
getAuthenticatePublishHandler 1224  
getAuthenticationList 1749  
getAuthenticationMethod 548, 553  
getAuthMode 1120  
getAutoAllocateInterleavePorts 1278  
getAVSyncMethod 1270  
getBandwidth 1034  
getBaseClass 849, 882, 1765  
getBaseFilePath 674  
getBasePath 10, 830  
getBitrate 627  
getBitrateAudio 930  
getBitrateTotal 930  
getBitrateVideo 930  
getBlockCount 842  
getBlockMask 854  
getBlockPath 854  
getBlockPtr 840  
getBlockSize 842  
getBodySize 771, 792, 795, 797, 800, 802, 808, 814, 816, 821  
getBoolean 147, 183, 199, 200, 234, 236  
getBuffer 581  
getBufferSize 803  
getBufferTime 366, 1405  
getBurstStartStop 1433  
getByte 146, 183, 199, 200, 234, 235  
getByteAllocation 77  
getByteContainerLevel 213  
getCacheControlStatus4xx 593  
getCacheName 1431  
getCallbackParamCount 971  
getCanHandleFilter 830  
getCEACaptionConversionContext 1743  
getChecksum 1373  
getChunkCounter 215  
getChunker 530  
getChunkGroupDuration 424  
getChunkIndex 418  
getChunks 210  
getClassName 202, 205, 237  
getClient 273, 328, 335, 1412, 1736  
getClientById 273  
getClientCount 272, 1740  
getClientCountTotal 272  
getClientID 989  
getClientId 355, 1412  
getClientIdGenerator 1299, 1321  
getClientIdGeneratorRecycleDelaySize 1323  
getClientIdGeneratorRecycleSize 1323  
getClientIdGeneratorTimeout 1323  
getClientIdleFrequency 284, 1745  
getClientProperty 1009  
 getClients 273, 1335, 1360  
getClientsLockObj 292  
getClientTimeout 1736  
getClientUpdates 1363  
getClosestStartTime 422, 484, 535  
getCodecEntryForTime 479  
getCodecId 931  
getCommandInterface 1319  
getCommandInterfaceCommandHandler 1301, 1323  
getCommandInterfaceHostPort 1297, 1319  
getCommittedVirtualMemory 1322  
getCommonInitialTextRepString 406  
getConce 1120  
getConfigDataOrDefault 1029  
getConfigDir 1773  
getConfigPath 242  
getConfiguation 1705  
getConnectionCounter 243, 277, 1296, 1318, 1741  
getConnectionFlashVerion 1084  
getConnectionFlashVersion 1084  
getConnectionHolder 608, 1227  
getConnectionLimit 1742, 1774  
getConnectionPageURL 1085  
getConnectionQueryStr 1083  
getConnectionsMaximum 1304, 1326  
getConnectionSwfURL 1085  
getConnectionTimeout 828, 882, 1080  
getConnectionValidator 1318  
getConnectLastAttempt 869, 1092  
getConnectLastForceReset 869  
getConnectLastSuccess 869, 1093  
getConnectTime 364  
getContentDescriptors 809  
getContentLength 556  
getContentType 557, 808

getContextStr 301, 516, 529, 1016, 1020, 1035, 1043, 1083, 1159, 1442, 1541, 1677  
getCookieStr 613, 1234  
getCoreHandlerPoolSize 1298, 1320, 1738  
getCoreTransportPoolSize 1298, 1320, 1738  
getCount 1505  
getCpuTime 104  
getCrossSession 1037  
getCrossSessionLock 1038  
getCryptoPoolActiveCount 1322  
getCryptoPoolMaxSize 1322  
getCurrent 342  
getCurrentDuration 674  
getCurrentFile 675  
getCurrentHeapSize 1322  
getCurrentItem 1581  
getCurrentSize 674  
getCurrentTime 504  
getData 224, 415, 795, 798, 1351  
getDataBuffer 224  
getDataMissing 1421  
getDataSize 1406  
getDataTC 1410  
getDataType 1411  
getDate 47, 147, 183, 200, 201, 234, 236  
getDateStarted 244, 278, 365, 1296, 1318, 1742  
getDateString 48  
getDefaultBlockSize 849  
getDefaultStreamingStore 518  
getDefaultStreamPrefix 1304, 1327  
getDePacketizerList 1190  
getDePacketizerName 1191  
getDescription 772, 792, 795, 849, 882, 1767  
getDirection 1370  
getDirecton 9  
getDouble 147, 182, 199, 200, 233, 235  
getDstAppInstanceName 1096, 1140  
getDstApplicationName 1096, 1140  
getDstStreamName 1014, 1032, 1095  
getDuration 405, 1263, 1386, 1486, 1572, 1575  
getDvrApplicationContext 310  
getDvrArtifact 394, 415  
getDvrChunkAtTime 534  
getDvrChunkByIndex 534  
getDvrChunkNearTime 535  
getDvrFileSystemClass 519  
getDvrManager 529, 1519  
getDvrPlaylistRequest 492  
getDvrProperties 310  
getDvrRecorder 1439, 1443, 1510  
getDvrRecorderId 1519  
getDvrRecorderList 310, 1439, 1754  
getDvrRecorders 1511  
getDvrRepeater 1443  
getDvrSessionInfo 618  
getDvrStorageDir 518  
getDvrStorageWindowSeconds 518, 528  
getDvrStoreList 1754  
getDvrTime 483  
getDynamicLogProperties 1299, 1321, 1750  
getElapsedTime 378, 617, 1236, 1440  
getEmbeddedFlag 803  
getEncInfo 1390  
getEncMethod 1489  
getEncoding 717  
getEncryptionDelegate 519  
getEncryptionRepeaterSharedSecret 519  
getEncryptions 394  
getEncryptionsTextRepresentation 395  
getEncryptionType 471  
getEndIndex 409  
getEndTime 675  
getEntriesToPurge 479  
getError 1696  
getErrorField 1696  
getErrorString 675  
getEventHandler 1697  
getEventStr 1696  
getExecutor 1771  
getExt 1434  
getExtent 1372  
getExtraData 226  
getExtraDataKeys 227  
getFastPlaySettings 363, 1425  
getField 1695  
getFileExtension 1562  
getFileFormat 671  
getFileInBytes 85  
getFileInBytesRate 87  
getFileInfo 615

getFileIOPoolSize 1738  
getFilename 692, 694, 702  
getFileOutBytes 86  
getFileOutBytesRate 87  
getFilePath 674  
getFilePointer 9, 845  
getFileSystem 530  
getFileVersionDelegate 676  
getFirstByte 225  
getFirstEntry 423, 481  
getFirstIndex 423  
getFirstRendition 937  
getFirstSegment 922  
getFlags 789, 821  
getFlashVer 355  
getFloat 147, 182, 199, 200, 233, 236  
getFps 1369  
getFrameCount 1676  
getFrameIdentifier 795  
getFrameMapIds 765  
getFrames 765  
getFrameType 61, 62  
getGlobalLogValue 733  
getGUID 1321  
getHandler 1202  
getHandlerThreadPool 1299, 1320, 1739  
getHeader 555, 562  
getHeaderBytes 559  
getHeaderFlags 71  
getHeaderMap 555  
getHeaderNames 556  
getHeaders 562  
getHeaderSize 563, 1436  
getHeight 932  
getHighestVersionedStore 517  
getHomePath 1737  
getHost 1195, 1261  
getHostname 1015, 1032, 1095, 1156  
getHostPortsList 1739  
getHostType 1158, 1197  
getHowToPublish 1080  
getHTTPHeader 618  
getHTTPHeaderMap 618  
getHTTPHeaderNames 619  
getHTTPIdent2Response 1707  
getHTTPIdentResponse 1707  
getHTTPIntHeader 618  
getHTTPOriginMode 592  
getHttpProviders 1706  
getHTTPStreamerAdapter 608, 1752  
getHTTPStreamerAdapterIDs 1706, 1753  
getHTTPStreamerApplicationContext 299  
getHTTPStreamerContext 1748  
getHTTPStreamerItem 586  
getHTTPStreamerList 294, 1752  
getHTTPStreamerMaxPathLen 1755  
getHTTPStreamerProperties 293  
getHTTPStreamerRequestValidator 1755  
getHTTPStreamerSession 1378, 1440  
getHTTPStreamerSessionCount 297, 298  
getHTTPStreamerSessionCountsByName 297  
getHTTPStreamerSessions 297, 298  
getId 588  
getId 210  
getIdleCheckFrequency 1746  
getIdleFrequency 369, 587, 1232  
getIdleHandler 1228  
getIdleMinimumWaitTime 1749  
getIdleTimeout 866  
getIdleWorkerCount 1746  
getIdleWorkers 1749  
getIdStr 788, 820  
getIdString 883  
getImplementation 1028  
getIndex 405, 568, 637, 644, 651, 1554  
getIndexLive 637, 644  
getIndexMap 421  
getInfo 840  
getInIpAddress 1201  
getInitialRepeaterItems 524  
getInnerObj 205  
getInPort 1202  
getInputStream 558  
getInstance 732, 1315, 1780  
getInt 146, 182, 199, 200, 232, 235  
getIntData 157  
getIntHeader 556, 563  
getIOPerformanceCounter 279, 615, 1234  
getIoPerformanceCounter 244, 1295, 1296, 1317, 1318, 1741  
getIOScheduler 1741

getIoSession 1237  
getIp 363, 1225  
getIpAddress 609  
getItems 1550  
getJmxRemoteConfig 1320  
getKeepAliveTime 883  
getKeepAliveTimeout 1747  
getKey 198, 231  
getKeys 198, 231  
getLanguage 808, 933  
getLastAccessed 1373  
getLastAudioTC 1100  
getLastAudioTimecode 1567  
getLastAuthenticateMethod 1224  
getLastClientId 1353  
getLastConnectAcceptedStamp 343  
getLastConnectAcceptedStampString 343  
getLastConnectAcceptedTime 343  
getLastConnectAcceptedTimeString 343  
getLastConnectRejectedByReasonStamp 344  
getLastConnectRejectedByReasonStampString 344  
getLastConnectRejectedByReasonTime 344  
getLastConnectRejectedByReasonTimeString 344  
getLastConnectRejectedStamp 343  
getLastConnectRejectedStampString 343  
getLastConnectRejectedTime 343  
getLastConnectRejectedTimeString 344  
getLastDataTC 1100  
getLastDataTimecode 1567  
getLastDisconnectStamp 345  
getLastDisconnectStampString 345  
getLastDisconnectTime 345  
getLastDisconnectTimeString 345  
getLastKeyFrame 1422  
getLastLiveEntry 423, 481  
getLastOperation 1353  
getLastPacket 1423  
getLastPacketsByType 1274  
getLastPurgeTime 504  
getLastRecordedEntry 423, 478  
getLastRecordedIndex 420, 483  
getLastRequest 622  
getLastRun 1696  
getLastSession 1081  
getLastStreamId 987  
getLastTC 68  
getLastTouchTime 303  
getLastValidateTime 374  
getLastVideoTC 1100  
getLastVideoTimecode 1567  
getLength 1387, 1554  
getLicenseCounter 1323  
getLicenseHolder 1522, 1540  
getLicenseTracker 1324  
getLicenseValidator 1318  
getLiveDuration 422, 483  
getLiveEntries 422, 481  
getLiveEntriesWithLimit 422, 482  
getLivePlaylist 651  
getLiveRangeEndingBeforeTime 423  
getLiveRepeaterCapabilities 378  
getLiveStreamDvrRecorderControl 310  
getLiveStreamingPacketizer 609  
getLiveStreamPacketizer 893, 990, 1439, 1440, 1509  
getLiveStreamPacketizerControl 300  
getLiveStreamPacketizerId 1529  
getLiveStreamPacketizerList 294, 377, 1438, 1753  
getLiveStreamPacketizerLock 1509  
getLiveStreamPacketizerProperties 292  
getLiveStreamRecordManager 1754  
getLiveStreamRepeater 894, 1441  
getLiveStreamTranscoder 1444  
getLiveStreamTranscoderControl 309  
getLiveStreamTranscoderItem 1539  
getLiveStreamTranscoderList 309, 378, 1439, 1753  
getLiveStreamTranscoders 1444  
getLiveStreamTranscoderSessionCount 1302, 1325, 1756  
getLiveTailEntries 422  
getLiveThreads 1322  
getLocale 559  
getLocalListeners 1088  
getLocalPath 843  
getLock 226, 606, 764, 870, 890, 899, 922, 929, 938, 1080, 1189, 1580  
getLockCount 839, 892, 906  
getLogger 722, 733, 977  
getLoggerObj 731, 733  
getLong 146, 182, 199, 200, 233, 235  
getMACAddress 104  
getManagerProperties 271, 1299, 1321, 1737

getManifest 529  
getManifestChannel 484  
getManifestRepresentation 394, 397, 405, 416, 418  
getMapFileIndex 1014, 1034  
getMapIdStr 771, 789, 820  
getMaxHeapSize 1322  
getMaximumPendingReadBytes 283, 1744  
getMaximumPendingWriteBytes 282, 377, 1744  
getMaximumSetBufferTime 283, 377, 1745  
getMaxRTCPWaitTime 1270  
getMaxStorageDirDepth 293  
getMaxTimecode 1422, 1566  
getMaxTimeToLive 842, 851  
getMediaCache 850  
getMediaCacheFilters 315  
getMediaCacheItem 829, 846  
getMediaCacheSource 839  
getMediaCaster 891, 900, 1457  
getMediaCasterCount 899  
getMediaCasterDef 866, 886  
getMediaCasterDefs 885  
getMediaCasterHostPortConfig 887  
getMediaCasterId 865, 891  
getMediaCasterList 1748  
getMediaCasterNames 885, 900  
getMediaCasterProcessorCount 888  
getMediaCasterProperties 292  
getMediacasterRTPRTSPRTPTTransportMode 313  
getMediaCasterSettings 1747  
getMediaCasterStreamItem 867  
getMediaCasterStreamItems 899  
getMediaCasterStreams 280  
getMediaCasterType 870, 1262  
getMediaCasterValidator 302  
getMediaExtension 10, 1378, 1387  
getMediaFile 840  
getMediaIOPerformance 1426  
getMediaIOPerformanceCounter 355  
getMediaList 938, 1378  
getMediaListProvider 313  
getMediaListSegment 930  
getMediaName 10, 841  
getMediaNameExpression 831  
getMediaNameFromContext 835  
getMediaPath 853  
getMediaReader 1279  
getMediaReaderContentType 312, 1755  
getMediaReaderProperties 292  
getMediaReaders 1742  
getMediaWriterProperties 292  
getMediaWriters 1742  
getMember 205  
getMemberCount 204  
getMembers 204  
getMessagesInBytes 85  
getMessagesInBytesRate 86  
getMessagesInCount 85  
getMessagesInCountRate 86  
getMessagesLossBytes 86  
getMessagesLossBytesRate 87  
getMessagesLossCount 85  
getMessagesLossCountRate 86  
getMessagesOutBytes 85  
getMessagesOutBytesRate 87  
getMessagesOutCount 85  
getMessagesOutCountRate 86  
getMetadata 1384  
getMetadataEntryForTime 479  
getMetadataPacket 1179, 1266  
getMetaDataProvider 1436  
getMethod 556  
getMethodValue 1120  
getMimeType 772  
getMinimumAvailableChunks 518  
getMinTimeToLive 842, 850  
getMissing 221  
getMode 1275  
getModFunctions 280  
getModuleInstance 281  
getModuleList 281  
getMp3TagMap 1742  
getMPEGTSAudioBitrate 1285  
getMPEGTSAudioLanguage 1271  
getMPEGTSAudioPID 1272  
getMPEGTSProgramID 1271  
getMPEGTSVideoBitrate 1285  
getMPEGTSVideoPID 1271  
getmRtpDestination 1159  
getMsgBytes 560  
getMultiplier 1369

getName 242, 270, 832, 849, 881, 922, 931, 1198, 1339, 1352, 1364, 1408, 1552, 1554, 1583, 1708, 1739, 1765, 1773  
getNameGroupByGroupName 1510  
getNameGroupRemoveDelayTime 1511  
getNameGroups 1510  
getNameGroupStreamNames 1510  
getNCCounter 1121  
getNetConnection 872, 1079, 1123, 1412  
getNetConnectionHostPortConfig 1747  
getNetConnectionIdleFrequency 1746  
getNetConnectionProcessorCount 1740  
getNextChunkIndex 483  
getNextCodecIndex 484  
getNextMapIndex 1037  
getNextMetadataIndex 484  
getNextNameGroupId 1509  
getNextNetConnectionId 1736  
getNextRun 1696  
getNextSessionId 1240  
getNextStreamIndex 1505  
getNextTimeMapIndex 484  
getNodeByTagName 112  
getNodeValue 113  
getNodeValueByTagName 112  
getNonce 1120  
getNormalizedNTPTimecode 1276  
getNumberFromString 1034  
getNumberLiveEntries 422, 480  
getNumberRecordedEntries 422, 480  
getObject 147, 159, 183, 200, 201, 234, 236  
getObjectEncoding 158, 162, 214, 376  
getObjectNames 1346  
getObjectReference 163  
getOffset 581  
getOffsetToNextTag 803  
getOnMetadataData 71  
getOnMetadataFilter 1576  
getOpaqueObject 846  
getOrCreate 1346  
getOriginalTimecodeThreshold 1030, 1093  
getOutHost 1273  
getOutIpAddress 1202  
getOutPort 1202  
getOutputStream 561  
getOverlayImages 1670  
getOwnerIdentifier 797  
getPacketizerName 517  
getPacketsSentAudio 1103  
getPacketsSentData 1103  
getPacketsSentVideo 1103  
getPacketStartTime 404  
getPacketType 582  
getPageUrl 368, 993  
getParam 971  
getParamBoolean 974, 975  
getParamCount 971  
getParamDate 973  
getParamDouble 973, 974  
getParameter 557  
getParameterMap 558  
getParameterNames 557  
getParameterValues 558  
getParamInt 973  
getParamLong 974  
getParamMixedArray 972  
getParamObj 972, 1115  
getParamString 972  
getParamType 971  
getParamValue 110  
getParent 1342, 1366, 1472  
getPassword 334, 1031, 1049, 1115  
getPath 11, 547, 559, 1377, 1387  
getPathDelimiter 831  
getPeakThreads 1322  
getPendingWriteCount 841  
getPendingWrites 841  
getPerformanceCount 850  
getPictureData 772  
getPictureType 772  
getPingRoundTripTime 365  
getPingTimeout 282, 374, 1743  
getPlayClass 1766  
getPlayDuration 619, 933  
getPlayer 1421  
getPlayerCount 892, 906  
getPlaylist 1581  
getPlaylistAlignment 492  
getPlayPackets 1422  
getPlayReadyDecryptorKey 1489  
getPlaySeek 619, 933

getPlayStart 619, 932  
getPlayStreamCount 280, 357  
getPlayStreamCountsByName 280  
getPlayStreams 356  
getPlayStreamsByName 280  
getPollingInterval 1584  
getPort 718, 828, 1015, 1033, 1095, 1704  
getProcessorCount 1705  
getProfile 1019, 1043  
getProfileData 1037  
getProperties 243, 271, 358, 528, 586, 591, 617, 850, 854, 882, 922, 930, 938, 1014, 1029, 1226, 1299, 1320, 1422, 1519, 1528, 1540, 1737, 1766, 1774  
getPropertiesPath 853  
getProperty 322, 881, 1338, 1361, 1735, 1766  
getPropertyBoolean 322  
getPropertyDouble 323  
getPropertyInt 323  
getPropertyLong 323  
getPropertyStr 322  
getProtocol 369, 558  
getProtocolUsage 246, 313, 1324, 1755  
getProtocolUsageTracker 1324  
getPublishAudioCodecId 1441  
getPublisher 1049, 1115, 1119, 1584  
getPublisherCount 296  
getPublishers 296  
getPublishersMaximum 1296, 1327  
getPublishStream 1124  
getPublishStreamCount 357  
getPublishStreamNames 301, 1502  
getPublishStreams 356  
getPublishVideoCodecId 1441  
getPurgeController 530  
getPushPublishContext 1757  
getPushPublishImpl 1019, 1043  
getPushPublishSessionCount 317  
getPushPublishSessions 315, 316  
getPushPublishType 1016, 1034  
getPushPublishVersionStr 1033, 1102  
getQOPValue 1120  
getQueryStr 367, 612, 1229, 1434  
getQueryString 556  
getQueueSize 1771  
getRange 409  
getReadAheadThreshold 851  
getReaderClass 850  
getReadRetries 828  
getReadTimeout 827  
getRealm 1119  
getReceiveBufferSize 828  
getReceiveVideoFPS 1426  
getReconnectDelay 1093  
getReconnectWaitTime 869, 1015, 1034  
getRecordedDuration 421, 483  
getRecordedEntries 420, 421, 477, 482  
getRecordedEntriesInRange 421, 478  
getRecordedEntriesMap 478  
getRecordedEntriesWithLimit 421, 482, 534  
getRecordedEntryByIndex 421, 478, 534  
getRecordedEntryByTime 534  
getRecordedEntryByTimeKey 421, 478  
getRecordedEntryStartingBeforeTime 421  
getRecorder 660  
getRecorderParams 669  
getRecordersList 660  
getRecordersMap 660  
getRecorderState 670  
getRecorderStateString 670  
getRecordingName 1522  
getRecordingStreamStore 518  
getRedirectSessionBody 622  
getRedirectSessionCode 621, 1237  
getRedirectSessionContentType 622  
getRedirectSessionMessage 1237  
getRedirectSessionURL 621, 1237  
getRefCount 1336, 1360  
getReference 133  
getReferrer 368, 612, 993, 1229  
getRemoteAddr 559  
getRemoteHost 559, 717  
getRendition 937  
getRenditions 937  
getRepeat 1551, 1583  
getRepeaterItemsDvr 524  
getRepeaterLastSeqence 1529  
getRepeaterMediaCasterDef 591  
getRepeaterOriginUrl 283, 374, 992  
getRepeaterQueryString 284  
getRequestFilters 547, 553

getRequestURI 556  
getRequestURL 557  
getRespAMFAudioObj 367, 1433  
getRespAMFDataObj 367, 1433  
getRespAMFVideoObj 367, 1433  
getRespFunctions 364  
getResponseAMFObj 367  
getResponseOnIncompleteChunk 594  
getRsoStorageDir 286  
getRsoStoragePath 286  
getRTMPAuthProvider 1096  
getRTPAVSyncMethod 289  
getRTPContext 1257, 1748  
getRTPDatagramConfigIncoming 1749  
getRTPDatagramConfigOutgoing 1749  
getRTPDestination 1283  
getRTPIIdleFrequency 285, 1745  
getRTPInfo 1276  
getRTPMaxRTCPWaitTime 290  
getRTPPlayAuthenticationMethod 289  
getRTPProperties 292  
getRTPPublishAuthenticationMethod 289  
getRTPPushPublishSession 1154  
getRTPSession 327, 335, 1207  
getRTPSessionCount 291  
getRTPSessionCountsByName 291  
getRTPSessions 290  
getRTPStream 376, 1437  
getRTPStreamContext 1285  
getRtpStreamWaitTimeout 1158  
getRTPWriteListener 1227  
getRTSPBindIpAddress 305, 1280  
getRTSPConnectionAddressType 306, 1280  
getRTSPConnectionIpAddress 305, 1280  
getRTSPMaximumPendingWriteBytes 303, 1279  
getRTSPMaxPathLen 1756  
getRTSPOriginAddressType 306, 1281  
getRTSPOriginIpAddress 306, 1281  
getRTSPPlayRangeStart 1233  
getRTSPPlayRangeStop 1233  
getRTSPRequestValidator 1756  
getRTSPSessionDescription 1277  
getRTSPSessionExtraLines 1283  
getRTSPSessionName 1277  
getRTSPSessionTimeout 303, 1279  
getRTSPStream 1222  
getRTSPTunnelingLock 1191  
getRTSPTunnelingSession 1191  
getRTSPTunnelingSessionId 1233  
getSample 582  
getScheme 558  
getSDPData 1206  
getSDPLang 1283  
getSecondByte 225  
getSecureTokenSharedSecret 1085  
getSegment 921, 922  
getSegmentationType 672  
getSegmentDuration 673  
getSegmentNumber 675  
getSegmentSchedule 677  
getSegmentScheduleString 673  
getSegmentSize 673  
getSendBufferSize 828  
getSeq 224  
getSerializer 202  
getSerializeSize 471  
getServerGUID 1321  
getServerHostPort 369  
getServerIp 610, 1228  
getServerName 558  
getServerPort 559, 611, 1228  
getSession 1123, 1240, 1259  
getSessionDescriptionData 909  
getSessionGUID 1321  
getSessionHolderByNetConnection 1082  
getSessionHolderBySession 1082  
getSessionId 605, 1221  
getSessionIds 1240  
getSessionListDebug 1081  
getSessionProtocol 588, 610  
getSessions 1188  
getSessionTimeout 607  
getSessionTracker 656  
getSessionType 609  
getSharedObjectReadAccess 288, 370  
getSharedObjects 273  
getSharedObjectWriteAccess 288, 370  
getShort 146, 182, 199, 200, 233, 235  
getSize 210, 221, 582, 765, 789, 821, 932  
getSlot 1335, 1359

getSlotNames 1335, 1359  
getSlots 1341, 1359  
getSlotVersion 1352  
getSourcePath 849  
getSoVersion 1352  
getSpeed 1576  
getSrc 211, 222, 1407  
getSrcStream 1016, 1035  
getSrcStreamBufferSize 1102  
getSrcStreamName 1014, 1032, 1095, 1155  
getSSLConfig 1707  
getSslFactoryClass 1704  
getStart 1554  
getstartIndex 409  
getStartLiveOnPreviousBufferTime 1586  
getStartOnPreviousBufferTime 1572  
getStartTC 1370  
getStartTCOffset 1370  
getStartTime 675  
getStartTimecode 404  
getStartupStreams 1753  
getStartupStreamsDelayTime 1757  
getStopTimecode 404  
getStoreDir 1340, 1348, 1365  
getStore 854  
getStoreItem 841  
getStream 606, 670, 865, 977, 1019, 1043, 1260, 1502, 1503, 1504, 1518, 1561  
getStreamAngle 1109  
getStreamArray 895  
getStreamAttributes 1271  
getStreamAudioSampleAccess 287, 371  
getStreamBandWidth 1109  
getStreamBaseName 517  
getStreamBitrate 991, 1688  
getStreamClientless 1503  
getStreamCloseWaitTime 1098  
getStreamCount 280  
getStreamDef 1769  
getStreamDefs 1768  
getStreamExt 614, 893, 1267  
getStreamFile 365, 366  
getStreamFileForRead 1432  
getStreamFileForWrite 1432  
getStreamFileMapper 293  
getStreamId 1260  
getStreamInfo 1261, 1271  
getStreamKeyDir 285  
getStreamKeyPath 286  
getStreamLastSeq 868  
getStreamLength 992, 1688  
getStreamList 896  
getStreamListLock 1502  
getStreamLock 1257  
getStreamManager 899  
getStreamMissingTime 868  
getStreamName 529, 613, 669, 1267, 1539  
getStreamNameAliasProvider 296  
getStreamNameLock 1502  
getStreamNameLogging 1268  
getStreamNameParts 614  
getStreamNames 906  
getStreamPacketizerItem 1188, 1235  
getStreamPort 1196  
getStreamPosition 614, 1387  
getStreamProperties 292  
getStreamProperty 1010  
getStreamQueryStr 1273  
getStreamReadAccess 287, 372  
getStreamRecorderProperties 317  
getStreamRunningIndex 870  
getStreams 271, 1413, 1502  
getStreamSessionId 1268  
getStreamSessionIp 1269  
getStreamSessionVersion 1268  
getStreamsToPublish 1124  
getStreamStorageDir 285  
getStreamStoragePath 286  
getStreamStore 517  
getStreamStores 517  
getStreamTimeout 867  
getStreamTimeoutLastReset 868  
getStreamTimeoutLastTime 868  
getStreamTimeoutReason 868  
getStreamTrack 1260  
getStreamType 272, 358, 880, 990, 1266, 1423, 1561  
getStreamTypeNames 1769  
getStreamTypes 1735  
getStreamTypeStr 592  
getStreamVersionHandler 519

getStreamVideoSampleAccess 286, 371  
getStreamWriterAccess 288, 373  
getString 146, 159, 181, 199, 200, 232, 234  
getStringReference 163  
getSyncLock 839  
getSystemTime 105  
getTarget 1695  
getTargetEncoding 162  
getTextEncoding 771, 792, 808, 814, 817  
getThreadPool 1298, 1320, 1739  
getTime 48  
getTimecode 211, 223, 810, 1172  
getTimecodesOutOfOrderThreshold 1100  
getTimedTextProperties 295  
getTimedTextProviderConfig 1753  
getTimeMap 486, 541  
getTimeMapEntries 541  
getTimeMapper 529  
getTimeMapping 418  
getTimeOffsetBetweenItems 1586  
getTimeRunning 244, 278, 365, 617, 1236, 1296, 1318, 1742  
getTimeRunningSeconds 244, 278, 365, 617, 1236, 1297, 1318, 1742  
getTimeSeconds 48  
getTimeStampFormat 808  
getTimeString 48  
getTitle 922, 934  
 getTotal 345  
 getTotalAccepted 345  
 getTotalIOPerformanceCounter 355  
 getTotalPacketsSent 1099  
 getTotalRejected 346  
 getTrack 1201, 1259  
 getTrackId 1259  
 getTrackNames 1259  
 getTrait 159, 202  
 getTraitReference 163  
 getTranscoderApplicationContext 310  
 getTranscoderLicenseInUse 1302, 1325  
 getTranscoderLicenseTotal 1302, 1326  
 getTranscoderName 1539  
 getTranscoderProperties 293  
 getTranscoderType 1541  
 getTranscoderVODContext 1755  
 getTranscoderVODDestination 627  
 getTranscoderVODIndex 624, 625  
 getTranscoderVODIndexDestinationsMap 624  
 getTranscoderVODIndexDestinationsOrder 624  
 getTranscoderVODSession 623  
 getTransportMode 1275  
 getTransportThreadPool 1299, 1320, 1739  
 getITL 1197, 1204  
 getTuningHandler 1302, 1325  
 getType 131, 181, 211, 222, 405, 409, 420, 932, 1019, 1042, 1173  
 getTypeAsString 929  
 getTypeStr 1707  
 getUDPManagedDeliveryCount 1284  
 getUDPManagedDeliveryDelay 1284  
 getUDPPortManager 1300, 1322  
 getUDPPortSharingManager 1301, 1322  
 getUDPTransport 1201, 1270  
 getUDPTransportManager 1189, 1752  
 getUnidentifiedSessionTimeout 1757  
 getUniqueId 891  
 getUniqueStreamIdStr 1440  
 getUnsignedShort 38  
 getUri 368, 611, 1229  
 getUrl 795, 816  
 getUserAgent 611, 1223  
 getUserAgents 1297, 1319  
 getUserHTTPHeaders 593, 620  
 getUserName 1031, 1049, 1115  
 getUserQueryStr 617  
 getUserTime 105  
 getUtcStartTime 404  
 getValidationFrequency 282, 1087, 1744  
 getValue 136, 148, 153, 171, 185, 202, 792, 811, 814  
 getVersion 116, 989, 1295, 1316, 1337, 1361  
 getVersioningOption 672  
 getVHost 243, 271, 327, 335, 361, 587, 591, 605, 865, 976, 1189, 1223, 1241, 1261, 1506  
 getVHostItems 1778  
 getVHostList 1300, 1321  
 getVHostMap 1777  
 getVHostNames 1778, 1781  
 getVideoCodec 61, 69, 397  
 getVideoCodecConfigPacket 1437  
 getVideoCodecId 931  
 getVideoFrameType 69  
 getVideoHost 1157, 1196

getVideoMissing 1421  
getVideoPacketizerItem 1188, 1235  
getVideoPort 1157, 1195  
getVideoSize 1406  
getVideoTC 1410  
getVideoTimecodeOffset 69, 70  
getVideoTrack 1260  
getVODLastTimeTC 1282  
getVODPlayLen 1282  
getVODSpeed 1587  
getVODStartTimeTC 1281  
getVODTimedTextProviderList 295  
getVODTimedTextProviderSet 295  
getVODTranscodeNGRP 623, 933  
getWaitOnMetadataTimeout 1101  
getWidth 932  
getWowzSession 380  
getWriteListener 373  
getXMLPropertyBool 115  
getXMLPropertyDouble 115  
getXMLPropertyExists 114  
getXMLPropertyInt 114  
getXMLPropertyIntSize 114  
getXMLPropertyLong 114  
getXMLPropertyLongSize 115  
getXMLPropertyStr 113  
gmtTimeZone 103  
grabFrame 1668  
GZIP 18

**H**

handleArchivedStream 391, 537  
handleCallback 1427  
handleConnectResponse 1048, 1112, 1119  
handleMessage 1178, 1204  
handleOnMetadata 1538  
handlePacket 1518, 1529, 1538  
hasAudio 485, 531  
hasCodecData 485  
hasData 485, 531  
hasEncryption 531  
hasOnMetadata 485, 531  
hasTimeMapData 486  
hasVideo 485, 531

head 823  
HEADERSIZE 779  
height 928  
hexadecimal 32  
hostname 1027  
HOSTNAME\_KEY 706  
HostPort 1703  
HostPortList 1709  
HOUR 1694  
howToPublish 1074  
HTTPORIGINMODE\_AUTO 590  
HTTPORIGINMODE\_OFF 590  
HTTPORIGINMODE\_ON 590  
HTTPRequestToByteArray 73, 74  
HTTPRequestToFile 73  
HTTPProvider2Base 546  
HTTPStreamerSessionCupertino 567  
HTTPStreamerSessionDirect 579  
HTTPStreamerSessionMPEGDash 636  
HTTPStreamerSessionSanJose 643  
HTTPStreamerSessionSmoothStreamer 650  
HTTPStreamerSessionWebM 656  
HTTPUtils 73

**I**

ID3FOOTER\_SIZE 763  
ID3Frames 764  
ID3HEADER\_SIZE 763  
ID3HEADER\_VERSION 763  
ID3HEADERFLAGS\_DEFAULT 763  
ID3HEADERFLAGS\_EXPERIMENTAL 764  
ID3HEADERFLAGS\_EXTENDED 764  
ID3HEADERFLAGS\_FOOTERPRESENT 764  
ID3HEADERFLAGS\_UNSYNC 764  
ID3V2FrameAttachedPicture 771  
ID3V2FrameBase 787  
ID3V2FrameComment 791  
ID3V2FrameLinkedInformation 794  
ID3V2FramePrivate 797  
ID3V2FrameRawBytes 800  
ID3V2FrameRecommendedBufferSize 802  
ID3V2FrameSynchronizedText 807  
ID3V2FrameSynchronizedTextDescriptor 810  
ID3V2FrameTextInformation 813

ID3V2FrameURLLink 816  
ID3V2Utils 818  
idle 1274, 1412  
idleFrequency 1220  
idleHandler 1219  
idStringToName 883  
importManifest 484  
incAbsTimecode 213  
incByteContainerLevel 213  
incClientCountTotal 272  
incPendingWriteCount 841  
incrementAccept 341  
incrementBytesIn 82  
incrementBytesLoss 84  
incrementBytesOut 84  
incrementFileIn 82  
incrementFileOut 83  
incrementMediaInBytes 1258, 1427  
incrementMediaLossBytes 1427  
incrementMediaOutBytes 1426  
incrementMessagesIn 82, 83  
incrementMessagesLoss 83, 84, 85  
incrementMessagesOut 83, 84  
incrementReject 341  
incSlotVersion 1352  
index 403  
indexOf 39  
indexOfDifferent 39  
indexTranscoderVOD 568, 624, 638, 645, 652  
info 724, 725  
init 8, 329, 330, 425, 427, 503, 508, 528, 539, 540, 546, 553, 587, 590, 659, 825, 829, 835, 839, 864, 891, 909, 1016, 1028, 1048, 1107, 1112, 1114, 1119, 1141, 1145, 1154, 1187, 1351, 1376, 1383, 1405, 1468, 1495, 1517, 1528, 1537, 1735, 1771, 1780  
initCOnce 1119  
initContextLogging 732  
initialAttributes 524  
initialize 477  
initializeLogging 732  
initialManifest 524  
initialManifestEnd 524  
initLiveStreamRepeating 990, 1441, 1472  
initMedia 854  
initProperties 1263  
initStream 987  
InputStream 25  
interleavePackets 64, 65  
internalConnect 1083, 1146  
interruptPlay 1471  
intToByteArray 36, 37  
intToByteArraySafeSync 819  
intToHexStr 99  
intToStr 99  
intValue 170  
invoke 1288  
invokePrevious 977, 978  
IOPerformanceCounter 81  
ioPerformanceCounter 1218  
ioSession 1221  
ioSessionToSessionHolder 1073  
ip 1217  
isAcceptConnection 272, 358  
isAcceptSession 613  
isAcceptWOWZConnections 1301, 1325  
isActive 606, 717, 1519, 1528  
isAdaptiveStreaming 1029  
isAddressMulticast 95  
isAdmin 1708  
isAMF0 158, 163  
isAMF3 158, 163  
isAMF3Start 132  
isAnnounce 1218, 1226  
isAnnounceOrDescribe 1227  
isAppend 1420  
isAppendFile 677  
isAppInstanceLoaded 242  
isApplicationLoaded 1737  
isArrayStart 132  
isAudio 224  
isAudioCodecConfig 68  
isAVSyncNonSR 1270  
isBlockInCache 840  
isBlockUDPOut 1205, 1285  
isByteArrayStart 133  
isByteContainerEmpty 213  
isByteContainerFull 213  
isCheckIpAddr 1269  
isCheckSSRC 1269  
isClient 1335, 1360  
isClustered 1431

isConnected 357, 1219, 1227  
isCurrentSession 1081  
isDebugAppTimeout 313  
isDebugEnabled 723  
isDebugLog 87, 1094, 1140  
isDebugOn 1697  
isDebugPackets 1099, 1139  
isDebugRTSPSession 1232  
isDescribe 1218, 1226  
isDRMAvailable 1303, 1326  
isDRMInUse 1303, 1326  
isDVRAvailable 1303, 1326  
isDVRInUse 1303, 1326  
isDynamic 205  
isDynamicLogContextLoaded 1300, 1321  
isEmpty 99, 410, 412, 420, 764  
isEnabledFor 723  
isEncoderInUse 1676  
isEncrypted 369  
isErrorEnabled 722  
isExpired 1697  
isExtraData 227  
isFileInfo 615  
isFirstABSTRequest 645  
isFlashMediaLiveEncoder 375  
isFlashVerionFMLE 1099  
isFlashVersion10 375  
isFlashVersion90115 375  
isFlashVersionFMLE 1099  
isFlashVersionH264Capable 375  
isForceAMF0 1173  
isForceMPEGTSOut 1284  
isForceRTSPInterleaved 1276  
isGlobalLogValueSet 733  
isHeadRequest 560  
isHeadResponse 563  
isHitEnd 1374  
isHTTPOrigin 623  
isHTTPOriginAvailable 1303, 1326  
isHTTPOriginOn 592  
isInfoEnabled 723  
isInitiateWOWZConnections 1302, 1325  
isInMulticast 1204  
isInRange 410, 412  
isIntData 157  
isLastSentAbsTimecode 214  
isLive 533, 1278  
isLiveRepeater 374  
isLiveStreamPacketize 1534  
isLiveStreamTranscode 1542  
isLoaded 530  
isLocal 843  
isLoggedConnect 1225  
isLongTimecode 214  
isMatch 1695  
isMediaCacheSourceAllowed 315  
isMediaCasterPlay 1442  
isMember 204  
isMergeOnMetadata 1442  
isMixedArrayStart 133  
isModePlay 1273  
isModePublish 1273  
isModeUnknown 1273  
isMoveFirstVideoFrameToZero 672  
isMoveToNextIfLiveStreamMissing 1586  
isMPEGTSOut 1278  
isMulticast 1194, 1204  
isNew 212  
ISO8601\_FORMAT 708  
ISO8601\_PATTERN 708  
isObjectEncodingAMF0 214, 376  
isObjectEncodingAMF3 214, 375  
isObjEnd 133  
isObjStart 132  
isOnMetadataPacket 71  
isOnMetadataToSetDataFrame 1097  
isOpen 9, 1377, 1384, 1424  
isOutMulticast 1204  
isOverlayAvailable 1671  
isPacketizeAudio 630, 1530  
isPacketizeData 630, 1531  
isPacketizeVideo 630, 1530  
isPassThru 851  
isPaused 1263  
isPending 1176  
isPersistent 1336, 1347, 1360  
isPlay 1411  
isPlaying 1406  
isPlaylistReady 491, 492  
isPlayLogged 607

isPublishDataEvents 1567  
isPublisherStream 1446  
isPublishStreamReady 1262, 1442  
isPurgingEnabled 503  
isPushPublishAvailable 1304, 1326  
isReadAhead 851  
isReadAheadTrigger 842  
isReceiveAudio 1425  
isReceiveVideo 1426  
isReconnectorRunning 1075  
isRecord 1411  
isRecordData 671  
isRecording 520, 533, 1521  
isRecordingPaused 520, 533, 1521  
isRedirectSession 621, 1236  
isRefuseIncomingUDPConnections 1286  
isRefuseOutgoingUDPConnections 1286  
isRemoveDefaultAppInstance 1097  
isRepeaterEdge 1530  
isRequestFullURL 827  
isResetOnTimecodeOutOfOrder 1099  
isResetPlayStream 1263  
isRESTAPIAvailable 1303, 1326  
isRestServerEnabled 1304, 1325  
isResyncAudioVideoOnSR 1265  
isRTPIgnoreProfileLevelId 1284  
isRTPIgnoreSPropParameterSets 1285  
isRTPIncomingDatagramPortValid 299  
isRTPWrapped 1153, 1157, 1197  
isRTSP 1272  
isRTSPAlwaysUseSDPPorts 1282  
isRTSPPull 1272  
isRunning 246, 1676  
isSecure 369, 559  
isSendCloseOnRejectConnection 380  
isSendFCAnnounce 1087  
isSendFCPublish 1080  
isSendOnMetadata 1097, 1548, 1569, 1585  
isSendOriginalTimecodes 1030, 1093  
isSendPlayStopLogEvent 1435  
isSendPublishStopLogEvent 1436  
isSendRecordStopLogEvent 1435  
isSendReleaseStream 1081  
isSendResult 975  
isSendSDESEvents 1277  
isSendStreamCloseCommands 1098  
isSession 868  
isSessionValid 1017, 1217, 1224  
isShutdownClient 1234  
isShutdownOnRelease 893  
isShuttingDown 1749  
isSingleStreamAV 1155  
isSplitOnTcDiscontinuity 671  
isSSL 369, 1079, 1103  
isStartLiveOnPreviousKeyFrame 1585  
isStartOnKeyFrame 671  
isStartOnPreviousKeyFrame 1572  
isStartStarupStreams 1754  
isStopIfStreamMissing 1572  
isStream 868, 1195  
isStreamDomainProtectionActive 592  
isStreaming 1708  
isStreamIsRunning 869  
isStreamStarted 1282  
isSuspended 1300, 1317, 1706, 1751  
isSwitchLog 1585  
isTemplateLoaded 1540  
isTimeout 605, 1279  
isTimeoutSession 607  
isTimesInMilliseconds 1585  
isTraceEnabled 723  
isTranscoderActive 1540  
isTranscodeResult 1444  
isTranscoderWatermark 1302, 1326  
isUnpublishOnEnd 1587  
isUseUstreamPingPong 1145  
isValid 891, 1391  
isValidated 607  
isValidFMLEConnections 307, 379  
isValidAudioCodec 630  
isValidStreamDomainStr 616  
isValidVideoCodec 630  
isVersionFile 677, 1485  
isVHostRunning 1316  
isVideo 224  
isVideoCodecConfig 68  
isVideoH264SEILListenerEmpty 1445  
isVideoKeyFrame 68, 70  
isVisited 1774  
isWaitForVideoKeyFrame 1485

isWaitOnMetadataAudio 1101  
isWaitOnMetadataAudioDatarate 1102  
isWaitOnMetadataAvailable 1100  
isWaitOnMetadataVideo 1101  
isWaitOnMetadataVideoDatarate 1102  
isWarnEnabled 722  
isWowzaAudioOnly 931  
isWowzaKeyFrameOnly 934  
isWowzaVideoOnly 934

## J

JNDI\_CONTEXT\_NAME 707

## K

KEYDATA\_MODE\_ENCRYPT 948  
KEYDATA\_MODE\_INIT 948  
KEYDATA\_MODE\_KEYFRAME\_ENCRYPT 948  
KEYDATA\_MODE\_KEYFRAME\_INIT 948  
KEYDATA\_MODE\_KEYFRAME\_PLAYLIST 948  
KEYDATA\_MODE\_PLAYLIST 948  
killClient 1736  
killRTSPSession 1737

## L

LANG\_ASCII 1614  
LANG\_ASCII\_DBL 1615  
LANG\_GB231280 1615  
LANG\_KSC56011987 1615  
LANG\_PRIV1 1615  
LANG\_PRIV2 1615  
LANG\_REG1 1615  
LANGUAGE\_ID\_ENGLISH 1641  
LANGUAGE\_LOCALE 1641  
lastAudioTC 1077  
lastAuthenticateMethod 1217  
lastDataTC 1077  
lastIdleReadTime 1077  
lastIdleWriteTime 1077  
lastModified 10, 843, 846  
lastVideoTC 1077  
len 216  
length 11, 100, 843, 846, 855, 1434, 1471

LICENSECOUNTER\_DRM\_BUYDRM\_LIVE 1726  
LICENSECOUNTER\_DRM\_BUYDRM\_VOD 1726  
LICENSECOUNTER\_DRM\_EZDRM\_LIVE 1725  
LICENSECOUNTER\_DRM\_EZDRM\_VOD 1725  
LICENSECOUNTER\_DRM\_VERIMATRIX\_LIVE 1725  
LICENSECOUNTER\_DRM\_VERIMATRIX\_VOD 1726  
LICENSECOUNTER\_NDVR 1725  
LICENSECOUNTER\_PUBLISHER 1725  
LICENSECOUNTER\_PUBLISHERTRANSCODER 1726  
LICENSECOUNTER\_TOTAL 1727  
LICENSECOUNTER\_TRANSCODE\_DECODE 1725  
LICENSECOUNTER\_TRANSCODE\_DECODECOUNTAU  
DIO 1726  
LICENSECOUNTER\_TRANSCODE\_DECODECOUNTAU  
DIOVIDEO 1727  
LICENSECOUNTER\_TRANSCODE\_DECODECOUNTVID  
EO 1726  
LICENSECOUNTER\_TRANSCODE\_DECODEPOLLING  
1726  
LICENSECOUNTER\_TRANSCODE\_ENCODE 1725  
LICENSECOUNTER\_TRANSCODE\_ENCODECOUNTAU  
DIO 1727  
LICENSECOUNTER\_TRANSCODE\_ENCODECOUNTAU  
DIOVIDEO 1727  
LICENSECOUNTER\_TRANSCODE\_ENCODECOUNTVID  
EO 1727  
LICENSECOUNTER\_TRANSCODE\_ENCODEPOLLING  
1726  
LICENSECOUNTER\_TRANSCODE\_STREAMNAMES  
1726  
LICENSECOUNTERS\_DRM 1727  
listeners 1073  
liveStreamPacketizers 1501  
load 1033, 1108, 1139, 1146, 1158, 1349, 1365  
loadArchivedRecordings 508  
loadArchivedStore 521  
loadCacheItem 848  
loadConfig 1777  
loadConfigFile 1684, 1690  
loadConfigProperties 113  
loadFile 773  
lock 839, 920, 927, 936, 1074, 1114, 1221, 1342, 1365  
lockRepeaterStreams 612, 1262  
log 723, 724  
LOG4J\_ID\_KEY 706

LOG4J\_PACKAGE\_NAME 705  
 logDebug 981  
 logError 981  
 loggedConnect 1218  
 LOGGERNAME\_SERVER 731  
 logInfo 981  
 logLiveChunk 568, 637, 644  
 logLiveFragment 651  
 logNotifier 1314  
 LogNotifyCalculateIncremental 712  
 logVODChunk 568, 637, 644  
 logVODFragment 651  
 logWarn 981  
 longToByteArray 37, 38  
 longToHexStr 99  
 longToStr 99  
 longValue 170

**M**

main 18, 1315  
 makeNewLoggerInstance 732  
 MANIFEST\_TAGNAME\_CAN\_PLAY 476  
 MANIFEST\_TAGNAME\_CAN\_RECORD 476  
 MANIFEST\_TAGNAME\_CHUNK\_GROUPING 477  
 MANIFEST\_TAGNAME\_CURRENT\_TIME 477  
 MANIFEST\_TAGNAME\_HAS\_ENCRYPTION 476  
 MANIFEST\_TAGNAME\_PURGE\_TIME 477  
 MANIFESTFILE\_KEY\_ARTIFACT 402  
 MANIFESTFILE\_KEY\_AUDIO\_CODEC 402  
 MANIFESTFILE\_KEY\_CHUNKINDEX 402  
 MANIFESTFILE\_KEY\_DVRTIME 402  
 MANIFESTFILE\_KEY\_ENCRYPTIONS 403  
 MANIFESTFILE\_KEY\_INDEX 401  
 MANIFESTFILE\_KEY\_METADATA 403  
 MANIFESTFILE\_KEY\_NAME 402  
 MANIFESTFILE\_KEY\_PACKETTIME 402  
 MANIFESTFILE\_KEY\_SIZE 402  
 MANIFESTFILE\_KEY\_START 401  
 MANIFESTFILE\_KEY\_STOP 401  
 MANIFESTFILE\_KEY\_TYPE 401  
 MANIFESTFILE\_KEY\_UTCTIME 402  
 MANIFESTFILE\_KEY\_VIDEO\_CODEC 402  
 mapFileIndex 1026  
 MAX\_CCCOUNT 1609  
 MAXSTREAMINDEX 1500  
 md5Digest 89, 1112  
 MD5DigestUtils 89  
 md5Lock 89  
 MEDIACACHE\_PREFIX 441  
 MediaCasterItem 880  
 MediaCasterList 885  
 MediaCasterSettings 887  
 MediaCasterStreamItem 890  
 MediaCasterStreamManager 895  
 MediaCasterStreamMap 898  
 MEDIICASTERTYPE\_LIVEREPEATER 864  
 MEDIICASTERTYPE\_RTPLIVE 864  
 MEDIICASTERTYPE\_SHOUTCAST 864  
 MEDIICASTERTYPE\_UNKNOWN 864  
 MediaList 921  
 mediaList 936  
 MediaListRendition 929  
 MediaListSegment 936  
 mediaListSegment 927  
 MediaReaderEncInfo 1489  
 MediaStreamBase 1495  
 mediaStreamListeners 1501  
 MediaStreamMap 1502  
 MediaUtils 94  
 members 196  
 METHOD\_NONE 1488  
 METHOD\_SAMPLE\_PLAYREADY 1489  
 methodValue 1118  
 MILLS\_PER\_HOUR 130  
 MIMETYPE\_VIDEO\_MP4 462  
 MIMETYPES\_JPEG 770  
 MIMETYPES\_PNG 770  
 MIMETYPES\_URL 770  
 mImplementation 1027  
 MINUTE 1694  
 MISSING 1372  
 MODE\_PLAY 1254  
 MODE\_PUBLISH 1254  
 MODE\_UNKNOWN 1254  
 ModuleBase 971  
 ModuleClientLogging 981  
 ModuleCore 986  
 ModuleFastPlay 995  
 ModuleMediaCaster 905

ModuleMediaCasterStreamMonitorAdvanced 1005  
ModuleProperties 1008  
monitors 1004  
MONTH 1694  
moveFirstVideoFrameToZero 697  
mRtpDestination 1153  
msb0baseTime 103  
msb1baseTime 103  
MSG\_STARTSTREAM 595  
MSG\_SWITCHSTREAM 595  
mType 1027

**N**

name 920, 927  
nameGroupDelayRemoveMap 1501  
nameGroupId 1500  
nameGroupRemoveDelayTime 1501  
nameGroups 1500  
ncCounter 1118  
netConnection 1123  
netconnectToSessionHolder 1073  
NetworkUtils 95  
newCacheItem 848  
newValue 106  
newXPathFactory 115  
next 1583  
nextStreamId 1501  
NO\_OPTIONS 17  
nonce 1118  
notifyActionOnCodecInfoAudio 1431  
notifyActionOnCodecInfoVideo 1431  
notifyActionOnMetaData 1430  
notifyActionPause 1430  
notifyActionPauseRaw 1430  
notifyActionPlay 1429  
notifyActionPublish 1430  
notifyActionSeek 1430  
notifyActionStop 1431  
notifyActionUnPublish 1430  
notifyAkamaiClientLogin 1089  
notifyAkamaiSetChallenge 1089  
notifyAkamaiSetOriginConnectionInfo 1089  
notifyConnect 1090  
notifyConnectFailure 1088  
notifyConnectStart 1088  
notifyConnectSuccess 1088  
notifyDvrStreamManagerCreate 312  
notifyDvrStreamManagerDestroy 312  
notifyDvrStreamManagerInit 312  
notifyDvrStreamStorageDeleted 523  
notifyDvrStreamStoreCreate 523  
notifyDvrStreamStoreDestroy 523  
notifyDvrStreamStoreInit 523  
notifyDvrStreamStoreLoaded 523  
notifyFCAnnounce 1090  
notifyFCPublish 1090  
notifyHandshakeResult 1089  
notifyHTTPSessionCreate 315, 568, 584, 638, 645, 651  
notifyHTTPSessionDestroy 315, 568, 584, 638, 645, 652  
notifyListener 699  
notifyLivePacket 1446  
notifyLiveStreamDvrRecorderCreate 311  
notifyLiveStreamDvrRecorderDestroy 311  
notifyLiveStreamDvrRecorderInit 311  
notifyLiveStreamPacketizerCreate 307  
notifyLiveStreamPacketizerDestroy 307  
notifyLiveStreamPacketizerInit 307  
notifyLiveStreamTranscoderCreate 308  
notifyLiveStreamTranscoderDestroy 308  
notifyLiveStreamTranscoderInit 308  
notifyMediaReaderClose 305  
notifyMediaReaderCreate 304  
notifyMediaReaderExtractMetaData 305  
notifyMediaReaderInit 304  
notifyMediaReaderOpen 304  
notifyMediaStreamCreate 1508  
notifyMediaStreamDestroy 1508  
notifyMediaWriterOnFLVAddMetadata 302  
notifyMediaWriterOnWriteComplete 302  
notifyPlayPublish 1507  
notifyPlayUnpublish 1507  
notifyPublishHandlerPlay 1091  
notifyPushPublisherSessionCreate 1092  
notifyPushPublisherSessionDestroy 1092  
notifyPushPublishSessionCreate 317  
notifyPushPublishSessionDestroy 317  
notifyRTPSessionCreate 314  
notifyRTPSessionDestroy 314  
notifySessionClosed 1092

notifySessionCreate 1242

notifySessionDestroy 1242

notifySessionIdle 1091

notifySessionOpened 1091

notifySlotDelete 1365

notifySlotSetValue 1365

notifyStreamCreate 1090

notifyStreamOnPlayStatus 1091

notifyStreamOnStatus 1091

notifyTimeReset 523

notifyValidateSession 1092

notifyValidateSessionResult 1092

notifyVHostClientConnect 1781

notifyVHostCreate 1782

notifyVHostInit 1782

notifyVHostItemCreate 1777

notifyVHostItemDestroy 1778

notifyVHostItemUpdate 1778

notifyVHostShutdownComplete 1782

notifyVHostShutdownStart 1782

notifyVideoH264Packet 1445

## O

ODDPARITY 1609

offlineTime 1079

offset 216

ON\_CAPTION 1659

ON\_CAPTION\_INFO 1659

ON\_METADATA\_TYPE 476

ON\_TEXT\_DATA 1659

onAACEncodeInfo 916

onAcceptConnection 383

onAcceptorCreate 1711

onAcceptorDestroy 1711

onAfterDecodeFrame 1673

onAfterEncodeFrame 1675

onAfterScaleFrame 1673

onAkamaiClientLogin 1052, 1127

onAkamaiSetChallenge 1052, 1127

onAkamaiSetOriginConnectionInfo 1052, 1127

onAnnounce 1163, 1182, 1230

onAppCreate 943

onAppDestroy 943

onApplicationCreate 319

onApplicationDestroy 319

onApplicationInstanceCreate 318

onApplicationInstanceDestroy 318

onAppStart 905, 942, 1005

onAppStop 905, 942, 1005

onBeforeDecodeFrame 1673

onBeforeEncodeFrame 1675

onBeforeScaleFrame 1673

onBind 548, 549

onCalculateSourceAudioBitrate 1663

onCalculateSourceVideoBitrate 1663

onCall 944

onCallback 1452

onClientAccept 381

onClientConnect 381

onClientDisconnect 381

onClientReject 382

onCodecConfigAAC 913

onCodecInfoAudio 1451

onCodecInfoVideo 1451

onConnect 945, 1052, 1128

onConnectAccept 945

onConnectFailure 876, 1051, 1127

onConnectReject 946

onConnectStart 875, 1051, 1127

onConnectSuccess 875, 1051, 1127

onCreate 570

onCreateRecord 664

onCreateRecorder 678

onCronEvent 1699

onData 871

onDescribe 1163, 1182, 1230

onDestroy 572

onDisconnect 384, 911, 945

onDvrStreamManagerCreate 1513, 1514

onDvrStreamManagerDestroy 1513, 1514

onDvrStreamManagerInit 1513, 1514

onDvrStreamManagerInitStreamName 1515

onFCAnnounce 1052, 1128

onFCPublish 1052, 1128

onFile 78, 857

onFillChunkDataPacket 573, 575

onFillChunkEnd 572, 575

onFillChunkStart 572, 574

onFLVAddMetadata 1487

onFrameAAC 913  
onFrameMP3 913  
onGetParameter 1163, 1182, 1230  
onGrabFrame 1669  
onHandshakeResult 1051, 1127  
onHeaderData 914, 916  
onHTTPCupertinoEncryptionKeyCreateLive 949  
onHTTPCupertinoEncryptionKeyCreateVOD 949  
onHTTPCupertinoEncryptionKeyData 949  
onHTTPCupertinoEncryptionKeyLiveChunk 950  
onHTTPCupertinoEncryptionKeyRequest 948  
onHTTPCupertinoEncryptionKeyVODChunk 949  
onHTTPCupertinoStreamingSessionCreate 951  
onHTTPCupertinoStreamingSessionDestroy 951  
onHTTPPMPEGDashEncryptionKeyLiveChunk 952  
onHTTPPMPEGDashEncryptionKeyVODChunk 952  
onHTTPPMPEGDashStreamingSessionCreate 953  
onHTTPPMPEGDashStreamingSessionDestroy 953  
onHTTPRequest 549  
onHTTPSanJoseStreamingSessionCreate 954  
onHTTPSanJoseStreamingSessionDestroy 954  
onHTTPSessionCreate 955  
onHTTPSessionDestroy 955  
onHTTPSmoothStreamingPlayReadyCreateLive 956  
onHTTPSmoothStreamingPlayReadyCreateVOD 956  
onHTTPSmoothStreamingSessionCreate 957  
onHTTPSmoothStreamingSessionDestroy 957  
onHTTPStreamerSessionCreate 626  
onHTTPStreamerSessionDestroy 626  
onIndex 571  
onInit 571  
onInitAfterLoadTemplate 1663  
onInitBeforeLoadTemplate 1663  
onInitStart 1662  
onInitStop 1663  
onItem 858  
onlineTime 1079  
onLivePacket 1456  
onLiveStreamDvrRecorderCreate 1523  
onLiveStreamDvrRecorderDestroy 1523  
onLiveStreamDvrRecorderInit 1523  
onLiveStreamPacketizerCreate 1532  
onLiveStreamPacketizerDestroy 1532  
onLiveStreamPacketizerInit 1532  
onLiveStreamTranscoderCreate 1543  
onLiveStreamTranscoderDestroy 1543  
onLiveStreamTranscoderInit 1543  
onLog 710, 713  
onMediaCacheSourceCreate 852  
onMediaCacheSourceDestory 852  
onMediaCacheStoreCreate 856  
onMediaCacheStoreDestory 856  
onMediaCasterCreate 873  
onMediaCasterDestroy 873  
onMediaReaderClose 1389  
onMediaReaderCreate 1388  
onMediaReaderExtractMetaData 1389  
onMediaReaderInit 1388  
onMediaReaderOpen 1388  
onMediaStreamCreate 1463  
onMediaStreamDestroy 1463  
onMetaData 914, 916, 1450  
ONMETADATA\_FILTER\_DURATION 1574  
ONMETADATA\_FILTER\_NONE 1574  
onMetadataToSetDataFrame 1074  
onModuleLoad 941  
onModuleUnload 941  
onMP3EncodeInfo 916  
onNewVHost 1318  
onOpen 571  
onOptions 1164, 1183, 1231  
onPause 1164, 1183, 1231, 1448  
onPauseRaw 1450  
onPerSourceInfoAdd 833  
onPerSourceInfoRemove 833  
onPlay 1164, 1183, 1231, 1447  
onPlaylistItemStart 1549  
onPlaylistItemStop 1549  
onPublish 1448  
onPublishHandlerPlay 1053, 1129  
onPushPublisherSessionCreate 1054, 1129  
onPushPublisherSessionDestroy 1054, 1129  
onPushPublishSessionCreate 958, 1021, 1045  
onPushPublishSessionDestroy 958, 1021, 1046  
onRecord 1164, 1183, 1231  
onRedirect 1164, 1183, 1231  
onRegisterPlayer 873  
onRegisterStreamNameGroup 1666  
onRejectConnection 383  
onReleaseStream 1052, 1128

onResetMediaCaster 878, 1006  
onResetStream 1667  
onResponseWriteStart 1290  
onResponseWriteStop 1290  
onResult 940, 961  
onRTPSessionCreate 959, 1180  
onRTPSessionDestroy 959, 1180  
onSeek 1448  
onSegmentEnd 680  
onSegmentStart 679  
onServerAfterVHostStartup 1308  
onServerBeforeVHostStartup 1308  
onServerConfigLoaded 1307  
onServerCreate 1305  
onServerInit 1305  
onServerShutdownComplete 1306  
onServerShutdownStart 1305  
onSessionAudioDecodeCodecInfo 1666  
onSessionAudioEncodeCodecInfo 1665  
onSessionAudioEncodeCreate 1664  
onSessionAudioEncodeInit 1664  
onSessionAudioEncodeSetup 1665  
onSessionClosed 1053, 1129  
onSessionDataEncodeCreate 1664  
onSessionDataEncodeInit 1665  
onSessionDestinationCreate 1663  
onSessionIdle 1053, 1129  
onSessionOpened 1053, 1129  
onSessionVideoDecodeCodecInfo 1666  
onSessionVideoEncodeCodecInfo 1665  
onSessionVideoEncodeCreate 1664  
onSessionVideoEncodeInit 1664  
onSessionVideoEncodeSetup 1665  
onSetParameter 1164, 1182, 1231  
onSetSourceStream 874  
onSetup 1164, 1184, 1232  
onSharedObjectConnect 1343  
onSharedObjectCreate 1343  
onSharedObjectDestroy 1343  
onSharedObjectDisconnect 1344  
onShutdownStart 1666  
onShutdownStop 1667  
onSlotDelete 1354  
onSlotSetValue 1354  
onSplitRecord 664  
onSplitRecorder 679  
onStartRecord 664  
onStartRecorder 678  
onStop 1448  
onStopRecord 665  
onStopRecorder 679  
onStreamCreate 960, 1005, 1053, 1128  
onStreamDestroy 960, 1005  
onStreamOnPlayStatus 1053, 1128  
onStreamOnStatus 1053, 1128  
onStreamStart 876, 1458  
onStreamStop 876  
onSwitchRecord 665  
onSwitchRecorder 679  
onTeardown 1164, 1184, 1232  
onTrim 914  
onUnbind 548, 550  
onUnPublish 1448  
onUnRegisterPlayer 874  
onUnregisterStreamNameGroup 1666  
onValidateMediaCaster 877, 1005  
onValidateMediaCastersStart 877, 1005  
onValidateMediaCastersStop 877, 1006  
onValidateSession 1054, 1129  
onValidateSessionResult 1054, 1129  
onVHostClientConnect 1761  
onVHostCreate 1760  
onVHostInit 1760  
onVHostItemCreate 1759  
onVHostItemDestroy 1759  
onVHostItemUpdate 1759  
onVHostShutdownComplete 1761  
onVHostShutdownStart 1760  
onVideoH264Packet 1455  
onWriteComplete 1487  
opaque 1111  
open 8, 846, 1377, 1383, 1551  
order 196  
ORDERED 18  
originalTimecodeThreshold 1028  
outprint 1079  
output 1176  
OUTPUT\_TYPE\_AMF 1640  
OUTPUT\_TYPE\_CEA608 1640  
OUTPUT\_TYPE\_TTML 1640

outputFile 699  
outputPath 699  
OutputStream 28  
OVERWRITE\_FILE 686

P

p 1495  
PACKET\_TIME 469  
packetComplete 1413  
packetizerLicenses 1500  
packetOutput 1689  
packetsSentAudio 1077  
packetsSentData 1078  
packetsSentVideo 1077  
packetTime 403  
PARAM1 966, 1114  
PARAM10 968  
PARAM2 967  
PARAM3 967  
PARAM4 967  
PARAM5 967  
PARAM6 967  
PARAM7 967  
PARAM8 967  
PARAM9 967  
PARAMMETHODNAME 966  
parseAllowDomains 284  
parseBodyForParams 557  
parseCommaSeparatedLanguages 933  
parseIdString 883  
parseQueryStr 110  
password 1027, 1114  
PASSWORDFILEFORMAT\_CLEAR 329  
PASSWORDFILEFORMAT\_UNKNOWN 329  
pathToFileURL 108  
pause 989, 1265, 1469  
PAUSE\_PAUSE 1466  
PAUSE\_PLAY 1466  
PAUSE\_TOGGLE 1466  
pauseRaw 993, 1469  
pauseRecording 521, 532, 1521  
peekByte 131  
PICTUREMAXFILESIZE 770  
PICTURETYPE\_ARTISTLOGO 770

PICTURETYPE\_COVERBACK 770  
PICTURETYPE\_COVERFRONT 769  
PICTURETYPE\_FILEICON 769  
PICTURETYPE\_ILLUSTRATION 770  
PICTURETYPE\_MOVIESCREENCAPTURE 770  
PICTURETYPE\_OTHER 769  
PICTURETYPE\_OTHERFILEICON 769  
PICTURETYPE\_PUBLISHERLOGO 770  
ping 362  
play 988, 1264, 1468, 1469, 1546, 1571, 1575, 1581, 1582, 1583  
play2 988  
playDuration 929  
PLAYEVENT\_AFTERBUFFERFILL 1382  
PLAYEVENT\_AFTERMETADATA 1382  
PLAYEVENT\_BEFOREBUFFERFILL 1382  
PLAYEVENT\_BEFOREMETADATA 1382  
PLAYEVENT\_STARTPLAYBACK 1382  
Playlist 1550  
PlaylistItem 1553  
playSeek 929  
PLAYSIZES\_AUDIO\_BYTES 1467  
PLAYSIZES\_AUDIO\_COUNT 1467  
PLAYSIZES\_DATA\_BYTES 1467  
PLAYSIZES\_DATA\_COUNT 1467  
PLAYSIZES\_LOSS\_BYTES 1467  
PLAYSIZES\_LOSS\_COUNT 1468  
PLAYSIZES\_SIZE 1468  
PLAYSIZES\_VIDEO\_BYTES 1467  
PLAYSIZES\_VIDEO\_COUNT 1467  
playStart 928  
PLAYSTATUSTYPE\_COMPLETE 1467  
PLAYSTATUSTYPE\_STOP 1467  
PLAYSTATUSTYPE\_SWITCH 1466  
PLAYTRANSITION\_APPEND 970  
PLAYTRANSITION\_APPEND\_IMMEDIATE 970  
PLAYTRANSITION\_RESET 970  
PLAYTRANSITION\_RESET\_IMMEDIATE 970  
PLAYTRANSITION\_STOP 970  
PLAYTRANSITION\_SWAP 970  
PLAYTRANSITION\_SWITCH 970  
PLAYTRANSITION\_UNKNOWN 970  
PLAYTRANSITIONSTR\_APPEND 969  
PLAYTRANSITIONSTR\_RESET 969  
PLAYTRANSITIONSTR\_STOP 969  
PLAYTRANSITIONSTR\_SWAP 969

PLAYTRANSITIONSTR\_SWITCH 969  
PLAYTRANSITIONSTR\_UNKNOWN 969  
port 1027  
postChunkAdded 510  
postChunksPurged 511  
posToBlock 842  
preChunkAdded 510  
preChunksPurged 510  
preload 842  
previous 1584  
profile 1042  
PROP\_3GPP\_CAPTION\_CHARACTER\_SET 1649  
PROP\_AMF\_CONVERTER\_INSERT\_ERASES\_IN\_GAPS  
1651  
PROP\_AMF\_CONVERTER\_TEXT\_TYPE 1650  
PROP\_AMF\_CONVERTER\_TRACK\_INDEX 1650  
PROP\_CAPTION\_CHARACTER\_SET 1648  
PROP\_CAPTION\_FILE\_NAMING\_RULE 1646  
PROP\_CAPTION\_FILENAME\_QUERY\_PARAMETER  
1655  
PROP\_CAPTION\_LANGUAGE\_QUERY\_PARAMETER  
1655  
PROP\_CAPTION\_LANGUAGES 1642  
PROP\_CAPTION\_PATH\_NAMING\_RULE 1647  
PROP\_CEA608\_CONVERTER\_CHARACTER\_SET 1649  
PROP\_CEA608\_CONVERTER\_CHARS\_PER\_LINE 1649  
PROP\_CEA608\_CONVERTER\_COLOR 1649  
PROP\_CUPERTINO\_CAPTION\_LANGUAGES 1642  
PROP\_CUPERTINO\_LIVE\_USE\_WEBVTT 1650  
PROP\_CUPERTINO\_VOD\_CAPTIONS\_ENABLED 1641  
PROP\_CUPERTINO\_VOD\_REMOVE\_EXISTING\_CEA608  
1650  
PROP\_CUPERTINO\_VOD\_USE\_WEBVTT 1650  
PROP\_DEBUG\_3GPP\_CAPTION\_PARSER 1654  
PROP\_DEBUG\_3GPP\_CAPTION\_PARSER\_TIME 1654  
PROP\_DEBUG\_CUPERTINO\_VOD\_CAPTION\_LANGUAGE\_SELECTION 1643  
PROP\_DEBUG\_CUPERTINO\_VOD\_CAPTION\_PROVIDE\_R\_DETERMINATION 1644  
PROP\_DEBUG\_LIVE\_CAPTION\_INGEST 1656  
PROP\_DEBUG\_LIVE\_CAPTION\_INGEST\_ON\_CAPTION\_CEA608\_LOG\_AS\_RAW 1659  
PROP\_DEBUG\_LIVE\_CAPTION\_INGEST\_ON\_CAPTION\_CEA608\_LOG\_AS\_SCC 1658  
PROP\_DEBUG\_LIVE\_CAPTION\_INGEST\_ON\_CAPTION\_CEA608\_VERBOSE 1658  
PROP\_DEBUG\_LIVE\_CAPTION\_ONCAPTION\_INGEST 1656  
PROP\_DEBUG\_LIVE\_CAPTION\_ONTEXTDATA\_INGEST 1656  
PROP\_DEBUG\_RTMP\_VOD\_CAPTION\_LANGUAGE\_SELECTION 1644  
PROP\_DEBUG\_RTMP\_VOD\_CAPTION\_PROVIDER\_DETERMINATION 1645  
PROP\_DEBUG\_SANJOSE\_VOD\_CAPTION\_LANGUAGE\_SELECTION 1644  
PROP\_DEBUG\_SANJOSE\_VOD\_CAPTION\_PROVIDER\_DETERMINATION 1645  
PROP\_DEBUG\_SCC\_CAPTION\_PARSER 1654  
PROP\_DEBUG\_SCC\_CAPTION\_PARSER\_TIME 1654  
PROP\_DEBUG\_SCC\_VOD\_CAPTION\_COMPANION\_FILES 1654  
PROP\_DEBUG\_SRT\_CAPTION\_PARSER 1652  
PROP\_DEBUG\_SRT\_CAPTION\_PARSER\_TIME 1653  
PROP\_DEBUG\_SRT\_VOD\_CAPTION\_COMPANION\_FILES 1652  
PROP\_DEBUG\_TTML\_CAPTION\_PARSER 1651  
PROP\_DEBUG\_TTML\_CAPTION\_PARSER\_TIME 1652  
PROP\_DEBUG\_TTML\_VOD\_CAPTION\_COMPANION\_FILES 1651  
PROP\_DEBUG\_VOD\_CAPTION\_COMPANION\_FILES 1645  
PROP\_DEBUG\_VOD\_CAPTION\_LANGUAGE\_SELECTION 1643  
PROP\_DEBUG\_VOD\_CAPTION\_PROVIDER\_DETERMINATION 1644  
PROP\_DEBUG\_WEBVTT\_CAPTION\_PARSER 1653  
PROP\_DEBUG\_WEBVTT\_CAPTION\_PARSER\_TIME 1653  
PROP\_DEBUG\_WEBVTT\_VOD\_CAPTION\_COMPANION\_FILES 1653  
PROP\_LIVE\_CAPTION\_DEFAULT\_LANGUAGES 1656  
PROP\_LIVE\_CAPTION\_INGEST\_AMF\_ID\_STRING 1655  
PROP\_LIVE\_CAPTION\_INGEST\_ON\_CAPTION\_CEA608\_ENABLE\_FIELD1 1657  
PROP\_LIVE\_CAPTION\_INGEST\_ON\_CAPTION\_CEA608\_ENABLE\_FIELD2 1657  
PROP\_LIVE\_CAPTION\_INGEST\_ON\_CAPTION\_CEA608\_IDLE\_TIME\_TO\_FLUSH\_DECODER 1657  
PROP\_LIVE\_CAPTION\_INGEST\_ON\_CAPTION\_CEA608\_LANGUAGE\_ID 1658

PROP\_LIVE\_CAPTION\_INGEST\_ON\_CAPTION\_CEA608  
\_MAX\_WARN\_MESSAGES 1658

PROP\_LIVE\_CAPTION\_INGEST\_ON\_CAPTION\_CEA608  
\_TRACK\_NUMBER 1658

PROP\_LIVE\_CAPTION\_INGEST\_ON\_CAPTION\_INFO\_STRING 1657

PROP\_LIVE\_CAPTION\_INGEST\_ON\_CAPTION\_STRING 1657

PROP\_LIVE\_CAPTION\_INGEST\_TYPE 1656

PROP\_LIVE\_CAPTION\_USE\_STREAM\_NAME\_GROUPS 1656

PROP\_MAXIMUM\_CAPTION\_DURATION 1647

PROP\_RTMP\_CAPTION\_LANGUAGES 1643

PROP\_RTMP\_VOD\_CAPTIONS\_ENABLED 1641

PROP\_SANJOSE\_CAPTION\_LANGUAGES 1643

PROP\_SANJOSE\_VOD\_CAPTIONS\_ENABLED 1642

PROP\_SRT\_CAPTION\_CHARACTER\_SET 1648

PROP\_TTML\_CAPTION\_CHARACTER\_SET 1648

PROP\_TTML\_INLINE\_TAGS\_IN\_TEXT 1652

PROP\_UNDEFINED\_LANG\_CODE 1646

PROP\_VOD\_CAPTIONS\_ENABLED 1641

PROP\_WEBVTT\_CONVERTER\_INSERT\_ERASES\_IN\_GAPS 1651

properties 546, 920, 927, 936, 1026, 1218

PROPERTY\_ALLOWABLE\_AV\_PACKET\_DELTA 446

PROPERTY\_APPEND\_DISCONTINUITY\_DELTA 441

PROPERTY\_ARCHIVE\_STRATEGY 442

PROPERTY\_AUDIO\_GROUP\_COUNT 500

PROPERTY\_AUDIO\_ONLY\_CHUNK\_TARGET\_DURATION 448

PROPERTY\_BREAK\_ON PTS 449

PROPERTY\_CHUNK\_CACHE\_CLASS 445

PROPERTY\_CHUNK\_DURATION\_MINIMUM 450

PROPERTY\_CHUNK\_GROUPING\_SECONDS 441

PROPERTY\_CHUNK\_MEMORY\_CACHESIZE 450

PROPERTY\_CHUNK\_READER\_CLASS 443

PROPERTY\_CHUNK\_WRITER\_CLASS 443

PROPERTY\_CUPERTINO\_ON\_CHUNK\_START\_RESET\_COUNTER 464

PROPERTY\_CUPERTINO\_PLAYLIST\_ALLOW\_CACHING 464

PROPERTY\_CUPERTINO\_PLAYLIST\_FORCE\_LIVE 463

PROPERTY\_CUPERTINO\_PLAYLIST\_FORCE\_NONLIVE 463

PROPERTY\_CUPERTINO\_PLAYLIST\_GZIP\_THRESHOLD D 464

PROPERTY\_CUPERTINO\_PLAYLIST\_USE\_GZIP 463

PROPERTY\_CUPERTINO\_PUT\_END\_TAG\_BEFORE\_CHUNKS 464

PROPERTY\_DEBUG\_CHUNK\_RETRIEVALS 456

PROPERTY\_DEBUG\_CUPERTINO\_PLAYER\_ADAPTER 458

PROPERTY\_DEBUG\_ENABLE 684

PROPERTY\_DEBUG\_FAILED\_CHUNK\_RETRIEVALS 457

PROPERTY\_DEBUG\_LOG\_INVALID\_CHUNK\_DETAILS 453

PROPERTY\_DEBUG\_LOG\_INVALID\_CHUNK\_MATCHER 454

PROPERTY\_DEBUG\_LOG\_VALID\_CHUNK\_DETAILS 454

PROPERTY\_DEBUG\_LOG\_VALID\_CHUNK\_MATCHER 455

PROPERTY\_DEBUG\_MAX\_INVALID\_CHUNKS\_LOGGED D 454

PROPERTY\_DEBUG\_MAX\_VALID\_CHUNKS\_LOGGED 455

PROPERTY\_DEBUG\_MAXIMUM\_RAW\_PACKETS 456

PROPERTY\_DEBUG\_MBR\_ALIGNMENT 459

PROPERTY\_DEBUG\_MBR\_ALIGNMENT\_RESOLUTION 459

PROPERTY\_DEBUG\_MBR\_DETAILS\_WHEN\_RECORD\_DNE 459

PROPERTY\_DEBUG\_MBR\_PLAYER\_ADAPTER 459

PROPERTY\_DEBUG\_MBR\_PLAYER\_ADAPTER\_IF\_SHIFTED\_EMPTY 459

PROPERTY\_DEBUG\_MBR\_RESOLVER 501

PROPERTY\_DEBUG\_MBR\_RESOLVER\_GAP\_DETAILS 501

PROPERTY\_DEBUG\_MBR\_RESOLVER\_INTERSECTION\_DETAILS 501

PROPERTY\_DEBUG\_METHODS 458

PROPERTY\_DEBUG\_PLAYER\_ADAPTER 458

PROPERTY\_DEBUG\_PLAYLIST\_REQUEST 460

PROPERTY\_DEBUG\_RAW\_PACKETS 455

PROPERTY\_DEBUG\_RAW\_PACKETS\_MATCHER 456

PROPERTY\_DEBUG\_REPEATERS 458

PROPERTY\_DEBUG\_SANJOSE\_PLAYER\_ADAPTER 458

PROPERTY\_DEBUG\_SMOOTH\_PLAYER\_ADAPTER 458

PROPERTY\_DEBUG\_STATE\_CHANGE 456

PROPERTY\_DEBUG\_TOSSED\_HOLDERS 455  
PROPERTY\_ENCRYPTION\_INFO\_DELEGATE 460  
PROPERTY\_FILE\_FORMAT 685  
PROPERTY\_FILE\_SYSTEM\_CLASS 444  
PROPERTY\_FILE\_VERSION\_DELEGATE 689  
PROPERTY\_FILE\_VERSION\_TEMPLATE 689  
PROPERTY\_INTERCEPTOR\_CHUNK\_DELEGATE 495  
PROPERTY\_MANIFEST\_PERSISTER\_CLASS 444  
PROPERTY\_MANIFEST\_PURGE\_CACHE\_SIZE 497  
PROPERTY\_MAX\_ALLOWABLE\_CHUNK\_DURATION 498  
PROPERTY\_MAX\_CHUNK\_LOG 457  
PROPERTY\_MAX\_RECALC\_DURATION\_LOG 457  
PROPERTY\_MAX\_RECORDING\_LENGTH 453  
PROPERTY\_MAX\_SAVE HOLDER\_SIZE 499  
PROPERTY\_MBR\_ALTERNATIVE\_MATCH\_DELTA 451  
PROPERTY\_MBR\_MINIMUM\_PACKETTIME\_GAP\_SIZE 451  
PROPERTY\_MBR\_MINIMUM\_UTCTIME\_GAP\_SIZE 452  
PROPERTY\_MBR\_USE\_SIMPLEALIGNMENT 498  
PROPERTY\_MBR\_USE\_UTC\_FOR\_ALIGNMENT 451  
PROPERTY\_MEDIACACHE\_ENABLED 453  
PROPERTY\_MEDIACACHE\_READER\_CLASS 453  
PROPERTY\_MINIMUM\_AVAILABLE\_CHUNKS 496  
PROPERTY\_MOVE\_FIRST\_VIDEO\_FRAME\_TO\_ZERO 687  
PROPERTY\_NOTIFY\_LISTENER 688  
PROPERTY\_OUTPUT\_FILE 688  
PROPERTY\_OUTPUT\_PATH 688  
PROPERTY\_PACKET\_DELTA\_TO\_NOTIFY 447  
PROPERTY\_PACKET\_DELTA\_TO\_RESET\_TIME 447  
PROPERTY\_PACKET\_DURATION\_MAXIMUM 452  
PROPERTY\_PACKET\_SORT\_TIME 449  
PROPERTY\_PLAYLIST\_REQUEST\_DELEGATE 460  
PROPERTY\_PURGE\_CONTROL\_CLASS 497  
PROPERTY\_RECALC\_DURATION\_MAX\_AV\_DIFFERENCE\_TRIGGER 498  
PROPERTY\_RECALC\_DURATION\_TRIGGER\_SIZE 499  
PROPERTY\_RECORD\_AUDIO 448  
PROPERTY\_RECORD\_DATA 448, 687  
PROPERTY\_RECORD\_VIDEO 448  
PROPERTY\_RECORDINGS\_LOADER\_CLASS 445  
PROPERTY\_REPEATERS\_HEARTBEAT\_DURATION 450  
PROPERTY\_REPEATERS\_SHARED\_SECRET 450  
PROPERTY\_SANJOSE\_ABST\_DURATION\_TOLERANCE 463  
PROPERTY\_SANJOSE\_ABST\_TIMESCALE 462  
PROPERTY\_SANJOSE\_PLAYLIST\_DELIVERYTYPE 460  
PROPERTY\_SANJOSE\_PLAYLIST\_LIVE\_STREAMTYPE 461  
PROPERTY\_SANJOSE\_PLAYLIST\_MIMETYPE 460  
PROPERTY\_SANJOSE\_PLAYLIST\_RECORDED\_STREAM\_TYPE 461  
PROPERTY\_SANJOSE\_PLAYLIST\_VERSION 461  
PROPERTY\_SEGMENT\_DURATION 687  
PROPERTY\_SEGMENT\_SCHEDULE 688  
PROPERTY\_SEGMENT\_SIZE 687  
PROPERTY\_SEGMENTATION\_TYPE 684  
PROPERTY\_SMOOTH\_MANIFEST\_H264\_CODEC 467  
PROPERTY\_SMOOTH\_MANIFEST\_LIVE\_CAN\_PAUSE 465  
PROPERTY\_SMOOTH\_MANIFEST\_LIVE\_CAN\_SEEK 464  
PROPERTY\_SMOOTH\_MANIFEST\_MAJOR\_VERSION 466  
PROPERTY\_SMOOTH\_MANIFEST\_MINOR\_VERSION 466  
PROPERTY\_SMOOTH\_MANIFEST\_RECORDED\_CAN\_PAUSE 465  
PROPERTY\_SMOOTH\_MANIFEST\_RECORDED\_CAN\_SEEK 465  
PROPERTY\_SMOOTH\_MANIFEST\_RECORDED\_SPECIFY\_DURATION 466  
PROPERTY\_SMOOTH\_MANIFEST\_VERBOSE\_DURATION 467  
PROPERTY\_SPLIT\_ON\_TC\_DISCONTINUITY 687  
PROPERTY\_START\_ON\_KEYFRAME 686  
PROPERTY\_START\_RECORDING\_ON\_STARTUP 448  
PROPERTY\_STORAGE\_DIRECTORY 442  
PROPERTY\_STREAM\_DATA 467  
PROPERTY\_TEXT\_READER\_CLASS 446  
PROPERTY\_TEXT\_WRITER\_CLASS 446  
PROPERTY\_USE\_SIMPLE\_FILE\_VERSION\_NAMING 684  
PROPERTY\_UTC\_ALIGN\_TO\_AUDIO 496  
PROPERTY\_UTC\_DELEGATE\_CLASS 496  
PROPERTY\_UTC\_USE\_WOWZA\_PACKET\_ARRIVAL 496  
PROPERTY\_VERIFY\_CHUNK\_EXISTENCE\_ON\_MANIFEST\_LOAD 497  
PROPERTY\_VERSIONING\_OPTION 686  
PROPERTY\_WAIT\_FOR\_CODEC\_TIME 500

PROPERTY\_WINDOW\_DURATION 442  
PROPS\_ITEM\_LASTMODIFIED 838  
PROPS\_ITEM\_LENGTH 838  
PROPS\_ITEM\_MEDIANAME 838  
PROPS\_SOURCE\_BLOCKSIZE 839  
PROPS\_SOURCE\_NAME 839  
PROTCOLUSAGE\_CUPERTINO 265  
PROTCOLUSAGE\_MC\_LIVEREPEATER 269  
PROTCOLUSAGE\_MC\_RTP 269  
PROTCOLUSAGE\_MC\_SHOUTCAST 269  
PROTCOLUSAGE\_MEDIACACHE 266  
PROTCOLUSAGE\_MPEGDASH 265  
PROTCOLUSAGE\_ORIGINCUPERTINO 266  
PROTCOLUSAGE\_ORIGINMPEGDASH 266  
PROTCOLUSAGE\_ORIGINSANJOSE 266  
PROTCOLUSAGE\_ORIGINSMOOTH 266  
PROTCOLUSAGE\_PUSHUBLISHING 266  
PROTCOLUSAGE\_RTMP 264  
PROTCOLUSAGE\_RTMP\_IN 267  
PROTCOLUSAGE\_RTMP\_OUT 267  
PROTCOLUSAGE\_RTMPE 265  
PROTCOLUSAGE\_RTMPE\_IN 268  
PROTCOLUSAGE\_RTMPE\_OUT 268  
PROTCOLUSAGE\_RTMPS 264  
PROTCOLUSAGE\_RTMPS\_IN 267  
PROTCOLUSAGE\_RTMPS\_OUT 267  
PROTCOLUSAGE\_RTMPT 264  
PROTCOLUSAGE\_RTMPT\_IN 267  
PROTCOLUSAGE\_RTMPT\_OUT 267  
PROTCOLUSAGE\_RTMPTE 265  
PROTCOLUSAGE\_RTMPTE\_IN 268  
PROTCOLUSAGE\_RTMPTE\_OUT 268  
PROTCOLUSAGE\_RTMPTS 264  
PROTCOLUSAGE\_RTMPTS\_IN 267  
PROTCOLUSAGE\_RTMPTS\_OUT 267  
PROTCOLUSAGE\_RTP 265  
PROTCOLUSAGE\_RTP\_IN 268  
PROTCOLUSAGE\_RTP\_OUT 268  
PROTCOLUSAGE\_RTSP\_IN 268  
PROTCOLUSAGE\_RTSP\_OUT 268  
PROTCOLUSAGE\_SANJOSE 265  
PROTCOLUSAGE\_SMOOTH 265  
PROTCOLUSAGE\_TOTAL 270  
PROTCOLUSAGE\_WEBM 265  
PROTCOLUSAGE\_WOWZ 265

PROTCOLUSAGE\_WOWZ\_IN 268  
PROTCOLUSAGE\_WOWZ\_OUT 269  
PROTCOLUSAGE\_WOWZE 266  
PROTCOLUSAGE\_WOWZE\_IN 269  
PROTCOLUSAGE\_WOWZE\_OUT 269  
PROTCOLUSAGE\_WOWZS 266  
PROTCOLUSAGE\_WOWZS\_IN 269  
PROTCOLUSAGE\_WOWZS\_OUT 269  
PROTO\_HTTPCUPERTINO 758  
PROTO\_HTTPDVRCHUNK 759  
PROTO\_HTTPMPEGDASH 759  
PROTO\_HTTPSANJOSE 758  
PROTO\_HTTPSCUPERTINO 759  
PROTO\_HTTPSDVRCHUNK 759  
PROTO\_HTTPSMOOTH 758  
PROTO\_HTTPSSANJOSE 759  
PROTO\_HTTPSSMOOTH 759  
PROTO\_HTTPSSTREAMER 759  
PROTO\_HTTPSTREAMER 758  
PROTO\_RTMP 757  
PROTO\_RTMPE 757  
PROTO\_RTMPS 757  
PROTO\_RTMPT 757  
PROTO\_RTMPTE 757  
PROTO\_RTMPTS 757  
PROTO\_RTSP 758  
PROTO\_WOWZ 758  
PROTO\_WOWZE 758  
PROTO\_WOWZS 758  
ptToDvr 542  
publish 988, 1427, 1495, 1561, 1562  
publisher 1114  
PublishingProviderBase 1568  
PublishingProviderLive 1571  
PublishingProviderMediaReader 1574  
publishStream 1123  
purge 1341, 1359  
purgeEntries 479, 533  
purgeManifestEntries 522  
PUSHPUBLISH\_TYPE\_RTMP 1013  
PUSHPUBLISH\_TYPE\_RTP 1013  
PUSHPUBLISH\_TYPE\_UNKNOWN 1013  
PushPublishBase 1028  
PushPublishContext 1037  
PushPublishHTTPBase 1040

pushPublishImpl 1042  
 PushPublishRTMP 1079  
 PushPublishRTMPAkamaiHandler 1107  
 PushPublishRTMPAuthProviderAdobe 1112  
 PushPublishRTMPAuthProviderBase 1114  
 PushPublishRTMPAuthProviderLimelight 1119  
 PushPublishRTMPNetConnectionSession 1123  
 PushPublishRTMPNotifyBase 1127  
 PushPublishRTMPProfileHandler 1135  
 PushPublishRTMPUstream 1145  
 PushPublishRTP 1154  
 PushPublishRTPRunner 1161  
 PushPublishRTSPActionNotify 1163  
 PUSH\_PUBLISH\_SESSION\_TYPE\_RTMP 1019  
 PUSH\_PUBLISH\_SESSION\_TYPE\_RTP 1019  
 PUSH\_PUBLISH\_SESSION\_TYPE\_UNKNOWN 1018  
 PushPublishSessionBase 1042  
 PushPublishSessionHTTPBase 1044  
 PushPublishSessionNotifyBase 1045  
 PushPublishSessionRTMP 1147  
 PushPublishSessionRTP 1166  
 pushPublishStreamDebug 1026  
 pushPublishType 1026  
 put 197, 198, 230, 231, 1346  
 putAll 323  
 putAllExtraData 226  
 putAudioPacketizerItem 1187, 1235  
 putCrossSession 1037  
 putDvrRecorder 1443  
 putExtraData 226  
 putFileInfo 615  
 putFrame 765  
 putGlobalLogValue 733  
 putHTTPStreamerAdapter 1753  
 putLiveStreamTranscoder 1444  
 putMetaData 1486  
 putProfileData 1037  
 putRTSPSessionExtraLine 1283  
 putRTSPTunnelingSession 1192  
 putSlot 1335, 1360  
 putStreamNameParts 614  
 putStreamPacketizerItem 1188, 1236  
 putVideoPacketizerItem 1188, 1236

qopValue 1118  
 QUERY\_PARAM\_DVR 491  
 queryStr 1219  
 QUERYSTR\_CAPTION\_FILENAME 1655  
 QUERYSTR\_CAPTIONLANGUAGES 1655

**R**

ranges 411  
 read 9, 25, 823, 844, 845  
 READ\_ACCESS\_ALL 354  
 READ\_ACCESS\_NONE 354  
 READACCESS 1358, 1404  
 readAhead 840  
 readAppConfig 245  
 readAppInstConfig 300  
 readChunk 62  
 readChunkHeader 62  
 readConfig 1301, 1323  
 readHeader 63  
 readPrevChunkHeader 62  
 readXMLConfig 1323  
 realm 1118  
 receiveAudio 991  
 RECEIVER\_NAME\_KEY 706  
 receiveVideo 991  
 reconnectWaitTime 1026  
 record 1264  
 recordData 697  
 RECORDER\_STATE\_ERROR 689  
 RECORDER\_STATE\_ERROR\_STRING 690  
 RECORDER\_STATE\_PENDING 689  
 RECORDER\_STATE\_PENDING\_STRING 690  
 RECORDER\_STATE\_RECORDING 690  
 RECORDER\_STATE\_RECORDING\_STRING 691  
 RECORDER\_STATE\_WAITING 690  
 RECORDER\_STATE\_WAITING\_STRING 690  
 recordFactory 663  
 recordingPaused 506  
 recordingReset 507  
 recordingResumed 506  
 recordingStarted 506  
 recordingStopped 507  
 redirectConnection 359, 360

**Q**

redirectSession 621, 622, 1220, 1237  
redirectSessionCode 1220  
redirectSessionMessage 1221  
redirectSessionURL 1221  
referrer 1219  
refreshManifest 477  
registerCallback 1428  
registerLiveStreamTranscoder 1324  
registerOnPlayStatus 1428  
registerOnStatus 1428  
registerPlayer 866, 892, 903  
registerPlayRTPSession 291  
rejectConnection 341, 359  
rejectSession 612, 1221  
release 893, 902, 1336, 1360  
releaseMediaCaster 907  
releaseSessionId 1240  
releaseSocketAddress 1189  
releaseStream 987  
releaseUDPPortPair 1190  
reloadConfig 1777  
reloadVHostConfig 1295, 1316  
remove 141, 177, 199, 232, 903, 1347, 1781  
removeAcceptorListener 1741  
removeActionListener 1230  
removeAppInstance 245  
removeApplicationInstanceListener 243  
removeApplicationListener 1740, 1781  
removeCallbackListener 1445  
removeClient 1336, 1348, 1360, 1736  
removeClientListener 274, 1429  
removeConnectionListener 340  
removeCrossSession 1037  
removeDefaultAppInstance 1076  
removeDvrChunkListener 534  
removeDvrRecorder 1443, 1510  
removeDvrRecorderListener 311  
removeDvrRecordingListener 523  
removeDvrStoreListener 522  
removeDvrStreamManagerListener 312  
removeFrame 765  
removeFromPlaylist 1582  
removeGlobalLogValue 733  
removeGlobalLogValues 732  
removeHeader 562  
removeHTTPLListener 315  
removeHTTPStreamerSession 299  
removeIdleWorkerListener 1747  
removeItem 1551  
removeListener 662, 676, 1088, 1584  
removeLivePacketListener 1446  
removeLiveStreamPacketizer 1508  
removeLiveStreamPacketizerListener 307  
removeLiveStreamTranscoder 1444  
removeLiveStreamTranscoderListener 308  
removeMediaCasterListener 277  
removeMediaReaderListener 304  
removeMediaStreamListener 275, 1508  
removeMediaWriterListener 302  
removeMetaDataListener 915  
removeModuleListener 281  
removeNameGroup 1509  
removePendingWrites 841  
removePlayStreamByName 279  
removePublisher 296  
removePushPublishSession 316  
removePushPublishSessionListener 316  
removePushPublishSessions 316  
removeRendition 937  
removeRTPLListener 314  
removeRTPSession 291  
removeRTSPStream 1222  
removeRTSPTunnelingSession 1191  
removeSDPFiles 1153  
removeSegment 921, 922  
removeServerListener 1297, 1319  
removeSession 1082, 1241  
removeSessionListener 1242  
removeSharedObjectListener 276, 1348  
removeSlotListener 1340, 1365  
removeStream 1506  
removeStreamDomainStr 616  
removeStreamStore 524  
removeTrack 1258  
removeTrackInternal 1258  
removeVHostItemListener 1777  
removeVideoH264SEILListener 1445  
renditions 936  
reparentClient 377, 1750  
ReplaceItem 106

requestFilters 546  
requiresLayout 717  
reset 528, 892, 921, 1468, 1763, 1773  
resetConnection 1014, 1083, 1154  
resetMDC 732  
resetMediaCasterStream 300  
resetNoLookup 1468  
resetOnTimecodeOutOfOrder 1076  
resetSentMetadataFlag 1266  
resetStream 490, 520, 630, 906, 1518, 1529, 1538  
resetTimecodes 1473  
resolveMediaList 1375  
resolvePlayAlias 1459, 1460, 1461  
resolveStreamAlias 1459, 1461  
ResponseFunction 1169, 1170  
ResponseFunctions 1175  
resumeEncoding 29  
resumeRecording 521, 532, 1521  
retrieveRawChunk 425, 505  
REVERSE 8  
rewind 1384  
rtmpAuthProvider 1079  
rtmpDebugLog 1072  
RTMPENABLED 1289  
RTPContext 1187  
RTPDestination 1194  
RTPPort 1201  
RTPPushPublishSession 1206  
rtpPushPublishSession 1153  
RTPSession 1221  
rtpSession 327  
RTPSessions 1240  
RTPStream 1257  
RTPUtils 1681  
rtpWriteListener 1218  
rtspPlayRangeStart 1220  
rtspPlayRangeStop 1220  
rtspTunnelingSessionId 1216  
run 1161, 1581

S

salt 1111  
SANJOSE\_F4M\_STREAMINGTYPE\_STREAMING 462  
SANJOSE\_F4M\_STREAMTYPE\_DVR 461  
SANJOSE\_F4M\_STREAMTYPE\_LIVE 462  
SANJOSE\_F4M\_STREAMTYPE\_LIVEORRECORDED 462  
SANJOSE\_F4M\_STREAMTYPE\_RECORDED 462  
SANJOSE\_F4M\_VERSION\_1\_0 462  
SANJOSE\_F4M\_VERSION\_2\_0 462  
saveBlock 840  
SDPFILEEXTENSION 1152  
SDPLOCATION\_AUDIO 1254  
SDPLOCATION\_STREAM 1254  
SDPLOCATION\_VIDEO 1254  
secureTokenSharedSecret 1075  
seek 9, 845, 989, 1384, 1469, 1547, 1571, 1575  
SEEK\_EXACT 1383  
SEEK\_KEYCLOSE 1383  
SEEK\_KEYDOWN 1383  
SEEK\_KEYUP 1382  
SEEKTARGET\_AUDIO 1382  
SEEKTARGET\_ENHANCED 1382  
SEEKTARGET\_VIDEOKEYFRAME 1382  
SEGMENT\_BY\_DURATION 685  
SEGMENT\_BY\_SCHEDULE 685  
SEGMENT\_BY\_SIZE 685  
SEGMENT\_NONE 685  
segmentationType 696  
segmentDuration 698  
segments 920  
segmentSchedule 698  
segmentSize 698  
SEI\_PAYLOADTYPE 1602  
SEI\_STARTCODE 1609  
SEI\_USERDATATYPECODE 1603  
send 1340, 1364, 1424  
sendAMF3 1424  
sendClientOnStatusError 978  
sendControlBytes 1433  
sendDirect 1423, 1424  
sendDirectAMF3 1423  
sendFCAnnounce 1074  
sendFCPublish 1074  
sendingPerfomance 1026  
sendInternal 1364  
sendLivePlaySeek 1414  
sendLivePlayStart 1413  
sendLivePlaySwitch 1414  
sendMessage 1203

sendOnMetadata 489, 1074, 1568  
sendOriginalTimecodes 1028, 1074  
sendPauseNotify 1419  
sendPlayReset 1415  
sendPlaySeek 1416  
sendPlayStart 1416, 1417, 1418  
sendPlayStatus 1418, 1419  
sendPlayStop 1415  
sendPlaySwitch 1417  
sendReleaseStream 1074  
sendResponse 1203  
sendResult 976, 977  
sendStreamCloseCommands 1074  
sendStreamCloseCommandsSendTime 1076  
sendStreamNotFound 1413  
sendStreamOnStatusError 978  
SendToBackupServer 1107  
sendUnpauseNotify 1419, 1420  
sendVODPlaySwitch 1414  
sendZeroLengthPacket 1386  
serialize 135, 148, 154, 171, 172, 184, 189, 201, 202, 237, 394, 397, 405, 410, 412, 416, 418, 471, 486, 766, 788, 821  
SERIALIZE\_CURRENT\_VERSION 403  
serializeBody 771, 791, 794, 797, 800, 802, 807, 813, 816, 821  
serializeDate 119  
serializeFooter 765  
serializeHeader 765  
serializeInt 119  
serializePacket 226  
serializer 196  
serializeString 121, 788  
serializeStringLen 788  
serializeStringNoLength 121  
serializeTags 765  
serializeZeroLengthString 120  
Server 1315  
serverIp 1219  
serverPort 1219  
service 586  
session 1123  
sessionClosed 867, 1082, 1204  
sessionId 1216  
sessionOpened 867, 1082, 1204  
sessionOrder 1073  
SESSIONPROTOCOL\_COUNT 604  
SESSIONPROTOCOL\_CUPERTINOSTREAMING 604  
SESSIONPROTOCOL\_DIRECTSTREAMING 604  
SESSIONPROTOCOL\_DVRCHUNKSTREAMING 604  
SESSIONPROTOCOL\_MPEGDASHSTREAMING 604  
SESSIONPROTOCOL\_SANJOSESTREAMING 604  
SESSIONPROTOCOL\_SMOOTHSTREAMING 604  
SESSIONPROTOCOL\_UNKNOWN 604  
SESSIONPROTOCOL\_WEBMSTREAMING 604  
sessionStart 909  
sessionStop 909  
SESSIONTYPE\_LIVE 603  
SESSIONTYPE\_LIVEDVR 603  
SESSIONTYPE\_UNKNOWN 603  
SESSIONTYPE\_VOD 603  
set 144, 145, 180, 181  
setAbsTimecode 224  
setAbsTimecodeLong 212  
setAbsTimecodeShort 212  
setAcceptConnection 272, 377  
setAcceptConnectionDescription 376  
setAcceptConnectionExObj 376  
setAcceptConnectionObj 376  
setAcceptSession 613  
setAcceptWOWZConnections 1301, 1325  
setActive 606  
setAdaptiveStreaming 1030  
setAddressCount 1205  
setAdminInterfaceHostPort 1739  
setAkamaiOriginIp 1087, 1108  
setAkamaiOriginPort 1087, 1108  
setAkamaiPassword 1086, 1109  
setAkamaiStreamId 1108  
setAkamaiUserName 1086, 1109  
setAllowDomains 284  
setAnnounce 1226  
setAppend 1420  
setAppInstance 609, 865, 1031, 1094, 1155, 1225, 1539  
setAppInstanceName 1267  
setAppInstanceProperty 1009  
setApplication 717, 1140  
setApplicationInstanceTouchTimeout 303  
setApplicationProperty 1008  
setApplicationTimeout 281, 1743  
setAppName 1267  
setAudioCodecId 931

setAudioHost 1156, 1197  
setAudioPort 1156, 1157, 1196  
setAudioSize 1406  
setAudioTC 1409, 1410  
setAuthenticatePlayHandler 1224  
setAuthenticatePublishHandler 1224  
setAuthenticationMethod 548, 553  
setAuthMode 1120  
setAVSyncMethod 1270  
setBandwidthLimit 988  
setBaseClass 849, 882, 1765  
setBasePath 830  
setBitrate 627  
setBitrateAudio 930  
setBitrateTotal 930  
setBitrateVideo 931  
setBlockMask 854  
setBlockUDPOut 1205, 1285  
setBody 1172  
setBuffer 581  
setBufferSize 803  
setBufferTime 366, 989, 1405, 1470  
setByteContainerLevel 213  
setCacheControlStatus4xx 593  
setCanHandleFilter 831  
setCanPlay 531  
setCanRecord 530  
setCheckIpAddr 1269  
setCheckSSRC 1269  
setChecksum 1373  
setChunkCounter 215  
setChunkGroupDuration 424  
setClassName 202, 205, 237  
setClient 328, 335, 1413  
setClientIdleFrequency 285, 1745  
setClientProperty 1009  
setClientTimeout 1739  
setClustered 1431  
setCodecInfoAudio 490, 630  
setCodecInfoVideo 490, 630  
setCommandInterface 1319  
setCommandInterfaceHostPort 1297, 1319  
setConce 1120  
setConfigDir 1773  
setConnected 1227  
setConnectionFlashVerion 1085  
setConnectionFlashVersion 1085, 1136  
setConnectionInfoPageURL 1137  
setConnectionInfoQueryString 1136  
setConnectionInfoSwfURL 1136  
setConnectionLimit 1774  
setConnectionPageURL 1085  
setConnectionQueryStr 1084  
setConnectionSwfURL 1085  
setConnectionTimeout 824, 828, 882, 1080, 1135  
setConnectionValidator 340  
setConnectLastAttempt 1092, 1137  
setConnectLastSuccess 1093, 1137  
setContent-Type 809  
setCookieStr 613, 1234  
setCoreHandlerPoolSize 1298, 1320, 1738  
setCoreTransportPoolSize 1298, 1320, 1738  
setCurrentDvrTime 503  
setData 795, 798, 1351  
setDataBuffer 222  
setDataSize 1407  
setDataTC 1410, 1411  
setDataType 1411  
setDebug 1697  
setDebugAppTimeout 314  
setDebugLog 87, 1015, 1033, 1094  
setDebugPackets 1099, 1139, 1140  
setDebugRTSPSession 1233  
setDefaultBlockSize 850  
setDefaultStreamingStore 518  
setDefaultStreamPrefix 1304, 1327  
setDescribe 1226  
setDescription 772, 792, 795, 849, 882, 1767  
setDestinationAddress 824, 829  
setDirection 1370  
setDirecton 10  
setDomainName 1703  
setDstAppInstanceName 1096, 1141  
setDstApplicationName 1096, 1140  
setDstStreamName 1015, 1032, 1095, 1141  
setDuration 1547, 1572, 1575  
setDvrRecorder 1443  
setDvrRecorderId 1519  
setDvrRecorderList 311, 1439  
setDvrSessionInfo 617

setDynamic 205  
setDynamicLogProperties 1300, 1321, 1750  
setEmbeddedFlag 803  
setEncMethod 1489  
setEncoding 717  
setEncryptions 394  
setError 1696  
setErrorField 1696  
setEventHandler 1697  
setEventStr 1696  
setExt 1435  
setFastPlay 996  
setFastPlaySettings 363, 1425  
setField 1695  
setFileExtension 1562  
setFileIOPoolSize 1738  
setFlags 789, 821  
setFlashVer 355  
setForceAMF0 1173  
setForceMPEGTSOut 1285  
setForceRTSPInterleaved 1276  
setFps 1369  
 setFrameCount 1676  
setFrameIdentifier 795  
setHandler 1203  
setHasEncryption 531  
setHeader 562  
setHeaderSize 1437  
setHeadResponse 563  
setHeight 932  
setHost 1015, 1032, 1156, 1195, 1261  
setHostname 1015, 1033, 1095  
setHostType 1158, 1197  
setHowToPublish 1080  
setHTTPTPIdent2Response 1707  
setHTTPTPIdentResponse 1707  
setHTTPTPOrigin 623  
setHTTPTPOriginMode 592  
setHTTPTPStreamerAdapter 608  
setHTTPTPStreamerItem 587  
setHTTPTPStreamerList 294  
setHTTPTPStreamerMaxPathLen 1756  
setHTTPTPStreamerRequestValidator 1755  
setHTTPTPStreamerSession 1378, 1440  
setID 588  
setId 210  
setIdleCheckFrequency 1746  
setIdleFrequency 370, 1232  
setIdleHandler 1228  
setIdleMinimumWaitTime 1749  
setIdleWorkerCount 1746  
setIdStr 788, 820  
setImplementation 1028  
setIndex 1554  
setInitiateWOWZConnections 1302, 1325  
setInnerObj 206  
setIntData 157  
setInternalRecorder 677  
setIntHeader 562  
setIOPerformanceCounter 1234  
setIoPerformanceCounter 1318  
setIoSession 1238  
setIp 1226  
setIpAddress 610, 1704  
setIsPlaying 1406  
setKeepAliveTime 883  
setKeepAliveTimeout 1747  
setLanguage 808, 933  
setLastAccessed 1373  
setLastAuthenticateMethod 1225  
setLastClientId 1353  
setLastModified 843  
setLastOperation 1353  
setLastRun 1696, 1697  
setLastSentAbsTimecode 214  
setLastValidateTime 374  
setLength 843, 855  
setLive 1278  
setLivePlaylist 651  
setLiveRepeaterCapabilities 378  
setLiveStreamDvrRecorderControl 310  
setLiveStreamingPacketizer 609  
setLiveStreamPacketizer 894, 990, 1440  
setLiveStreamPacketizerControl 300  
setLiveStreamPacketizerId 1529  
setLiveStreamPacketizerList 294, 378, 1438  
setLiveStreamRepeater 894, 1441  
setLiveStreamTranscoderControl 309  
setLiveStreamTranscoderItem 1540  
setLiveStreamTranscoderList 309, 378, 1439

setLocal 843  
setLocalPath 843  
setLock 930, 938  
setLoggedConnect 1225  
setLongTimecode 214  
setMapFileIndex 1014, 1034  
setMaximumPendingReadBytes 283, 1744  
setMaximumPendingWriteBytes 282, 1744  
setMaximumSetBufferTime 283, 1745  
setMaxRetries 824  
setMaxRTCPWaitTime 1270  
setMaxStorageDirDepth 293  
setMaxTimeToLive 842, 851  
setMediaCache 850  
setMediaCacheItem 829, 846  
setMediaCacheSource 839  
setMediaCasterDef 866  
setMediaCasterHostPortConfig 888  
setMediaCasterId 865  
setMediaCasterItem 1457  
setMediaCasterPlay 1442  
setMediaCasterProcessorCount 888  
setMediacasterRTPRTSPRTPTTransportMode 313  
setMediaCasterType 870  
setMediaCasterValidator 302  
setMediaList 938  
setMediaListProvider 313  
setMediaListSegment 930  
setMediaNameExpression 831  
setMediaReaderItem 1377, 1383  
setMediaWriterItem 1485  
setMergeOnMetadata 1443  
setMessageBytes 1171  
setMetaDataProvider 1436  
setMethodValue 1120  
setMimeType 772  
setMinTimeToLive 842, 851  
setMode 1275  
setMoveToNextIfLiveStreamMissing 1587  
setMPEGTSAudioBitrate 1285  
setMPEGTSAudioLanguage 1271  
setMPEGTSAudioPID 1272  
setMPEGTSSOut 1278  
setMPEGTSProgramID 1271  
setMPEGTSVideoBitrate 1285  
setMPEGTSVideoPID 1272  
setMultiplier 1369  
setName 242, 270, 832, 849, 881, 922, 931, 1198, 1339, 1352, 1364, 1408, 1470, 1708, 1765, 1773  
setNameGroupRemoveDelayTime 1511  
setNCCCounter 1121  
setNetConnection 1124, 1412  
setNetConnectionIdleFrequency 1746  
setNetConnectionProcessorCount 1740  
setNew 212  
setNonce 1120  
setObjectEncoding 158, 162, 214, 376  
setOffset 582  
setOffsetToNextTag 803  
setOnMetadataFilter 1576  
setOnMetadataToSetDataFrame 1097, 1138  
setOpaqueObject 846  
setOpen 1425  
setOriginalTimecodeThreshold 1030, 1031, 1094, 1138  
setOutHost 1274  
setOwnerIdentifier 798  
setPacketizeAudio 1530  
setPacketizeData 1531  
setPacketizeVideo 1530  
setPacketType 582  
setParent 1366, 1485  
setPassThru 851  
setPassword 1031, 1049, 1108, 1115  
setPathDelimiter 831  
setPersistent 1337, 1347, 1361  
setPictureData 772  
setPictureDataAsURL 772  
setPictureType 772  
setPingTimeout 282, 1743  
setPlay 1412  
setPlayClass 1766  
setPlayDuration 619, 933  
setPlayer 1421  
setPlayLogged 608  
setPlaySeek 619, 933  
setPlayStart 619, 933  
setPollingInterval 1584  
setPort 718, 828, 1015, 1033, 1096, 1156, 1704  
setProcessorCount 1705  
setProfile 1019, 1043

setProperties 547, 550, 586, 850, 854, 1377, 1387, 1774  
setProperty 321, 881, 1338, 1339, 1361, 1362, 1766  
setProxyAddress 824, 827  
setPublishAudioCodecId 1441  
setPublishDataEvents 1567  
setPublisherStream 1446  
setPublishStream 1124  
setPublishVideoCodecId 1441  
setPushPublishImpl 1019, 1043  
setQOPValue 1120  
setQueryStr 612, 1229, 1434  
setRange 409  
setReadAhead 851  
setReadAheadThreshold 851  
setReaderClass 850  
setReadRetries 828  
setReadTimeout 824, 827  
setRealm 1119  
setRealTimeStartTime 1547, 1572, 1575  
setReceiveAudio 1425  
setReceiveBufferSize 824, 828  
setReceiveVideo 1426  
setReceiveVideoFPS 1426  
setReconnectWaitTime 869, 1016, 1034, 1137  
setRecord 1411  
setRecordAudio 1520  
setRecordData 1520  
setRecordingName 1521  
setRecordingStreamStore 518  
setRecordVideo 1520  
setRedirectSession 621, 1236  
setRedirectSessionBody 622  
setRedirectSessionCode 621, 1237  
setRedirectSessionContentType 622  
setRedirectSessionMessage 1237  
setRedirectSessionURL 621, 1237  
setReferrer 612, 1229  
setRefuseIncomingUDPConnections 1286  
setRefuseOutgoingUDPConnections 1286  
setRemoteHost 717  
setRemoveDefaultAppInstance 1097, 1138  
setRepeat 1551, 1583  
setRepeaterEdge 1530  
setRepeaterMediaCasterDef 591  
setRepeaterOriginUrl 283, 374, 992  
setRepeaterQueryString 284  
setRequestFilters 547, 553  
setRequestFullURL 824, 827  
setResetOnTimecodeOutOfOrder 1100, 1138  
setResetPlayStream 1263  
setResponseCode 563  
setResponseOnIncompleteChunk 594  
setRestServerEnabled 1304, 1325  
setResyncAudioVideoOnSR 1265  
setRetAMFNumber 1172  
setRsoStorageDir 286  
setRTMPAuthProvider 1097  
setRTPAVSyncMethod 290  
setRTPDestination 1283  
setRTPIdleFrequency 285, 1746  
setRTPIgnoreProfileLevelId 1284  
setRTPIgnoreSPropParameterSets 1285  
setRTPMaxRTCPWaitTime 290  
setRTPPlayAuthenticationMethod 289  
setRTPPublishAuthenticationMethod 289  
setRTPPushPublishSession 1155  
setRTPSession 328, 335, 1207  
setRTPStream 1438  
setRTPStreamContext 1286  
setRtpStreamWaitTimeout 1158  
setRTPWrapped 1158, 1197  
setRTSP 1272  
setRTSPAlwaysUseSDPPorts 1283  
setRTSPBindIpAddress 305, 1280  
setRTSPConnectionAddressType 306, 1281  
setRTSPConnectionIpAddress 305, 1280  
setRTSPMaximumPendingWriteBytes 304, 1279  
setRTSPMaxPathLen 1756  
setRTSPOriginAddressType 306, 1281  
setRTSPOriginIpAddress 306, 1281  
setRTSPPlayRangeStart 1233  
setRTSPPlayRangeStop 1233  
setRTSPPull 1273  
setRTSPRequestValidator 1756  
setRTSPSessionDescription 1277  
setRTSPSessionName 1277  
setRTSPSessionTimeout 303, 1279  
setRTSPTunnelingSessionId 1233  
setSample 582  
setSDPData 1207

setSDPLang 1284  
setSecureTokenSharedSecret 1086, 1137  
setSendBufferSize 824, 828  
setSendCloseOnRejectConnection 380  
setSendFCAnnounce 1087, 1137  
setSendFCPublish 1081, 1136  
setSendOnMetadata 1097, 1138, 1547, 1569, 1585  
setSendOriginalTimecodes 1030, 1093, 1137  
setSendPlayStopLogEvent 1435  
setSendPublishStopLogEvent 1436  
setSendRecordStopLogEvent 1435  
setSendReleaseStream 1081, 1136  
setSendSDEEvents 1278  
setSendStreamCloseCommands 1098, 1138  
setSeq 224  
setSerializer 202  
setServer 1781  
setServerIp 610, 1228  
setServerPort 611, 1228  
setSession 1123, 1260  
setSessionId 605, 1222  
setSessionProtocol 588, 610  
setSessionTimeout 607  
setSessionTracker 656  
setSessionType 609  
setSessionValid 1224  
setSharedObjectReadAccess 288, 370  
setSharedObjectWriteAccess 289, 370  
setShutdownClient 377, 1234  
setShutdownOnRelease 893  
setShuttingDown 1750  
setSingleStreamAV 1155  
setSize 210, 221, 582, 932  
setSlotValue 1362, 1363  
setSlotVersion 1352  
setSourcePath 849  
setSoVersion 1352  
setSpeed 1576  
setSrc 211, 223, 1170, 1407  
setSrcStream 1016, 1035  
setSrcStreamBufferSize 1103  
setSrcStreamName 1014, 1032, 1095, 1155  
setSSL 1103, 1139  
setSSLConfig 1707  
setSslFactoryClass 1705  
setStartLiveOnPreviousBufferTime 1586  
setStartLiveOnPreviousKeyFrame 1586  
setStartOnPreviousBufferTime 1572  
setStartOnPreviousKeyFrame 1572  
setStartRecordingOnStartup 1522  
setStartStarupStreams 1754  
setStartTC 1370  
setStartTCOffset 1370  
setStartupStreamsDelayTime 1757  
setStopIfStreamMissing 1572  
setStorageDir 1340, 1348, 1365  
setStream 606, 670, 865, 1019, 1043  
setStreamAngle 1109  
setStreamAudioSampleAccess 287, 372  
setStreamBandWidth 1109  
setStreamCloseWaitTime 1098, 1138  
setStreamDomainProtectionActive 592  
setStreamExt 614, 893, 1267  
setStreamFileMapper 293  
setStreamId 1108  
setStreamKeyDir 285  
setStreamName 613, 1015, 1032, 1267, 1504, 1539  
setStreamNameAliasProvider 296  
setStreamNameLogging 1268  
setStreamPort 1196  
setStreamPosition 615, 1387  
setStreamProperty 1009  
setStreamQueryStr 1273  
setStreamReadAccess 287, 372  
setStreamRunning 870  
setStreamSessionId 1268  
setStreamSessionIp 1269  
setStreamSessionVersion 1268  
setStreamsToPublish 1124  
setStreamStorageDir 285  
setStreamTimeout 867  
setStreamType 272, 358, 880, 990, 1266, 1423, 1561  
setStreamTypeStr 592  
setStreamVersionHandler 519  
setStreamVideoSampleAccess 287, 371  
setStreamWriteAccess 288, 373  
setSuspended 1706  
setSwitchLog 1585  
setTarget 1695  
setTargetEncoding 162

setTextEncoding 771, 792, 808, 814, 817  
setThreadContext 379  
setTimecode 211, 223, 811, 1173  
setTimelines 223  
setTimecodesOutOfOrderThreshold 1100, 1138  
setTimeOffsetBetweenItems 1586  
setTimeoutSession 607  
setTimesInMilliseconds 1585  
setTimeStampFormat 808  
setTimeToLive 1158  
setTitle 923, 934  
setTrack 1201  
setTranscodeResult 1444  
setTranscoderName 1539  
setTranscoderType 1541  
setTranscoderVODDestination 627  
setTranscoderVODIndex 624  
setTranscoderVODSession 623  
setTransportMode 1275  
setTTL 1197, 1204  
setType 131, 211, 222, 932, 1173  
setTypeStr 1707  
setUDPManagedDeliveryCount 1284  
setUDPManagedDeliveryDelay 1284  
setUnidentifiedSessionTimeout 1757  
setUnpublishOnEnd 1587  
setUri 611, 1229  
setURL 795, 817  
setUserAgent 611, 1223  
setUserAgents 1297, 1319  
setUserHTTPHeader 593, 620  
setUserName 1031, 1049, 1108, 1115  
setUserQueryStr 617  
setUseUstreamPingPong 1145  
setValid 891  
setValidateFMLEConnections 308, 379  
setValidationFrequency 282, 1087, 1137, 1744  
setValue 792, 811, 814  
setVersion 1337, 1361  
setVersionFile 1485  
setVHost 327, 335, 587, 605, 1223  
setVideoCodecId 931  
setVideoHost 1157, 1196  
setVideoPort 1157, 1195  
setVideoSize 1406  
setVideoTC 1410  
setVisited 1774  
setVODLastTimeTC 1282  
setVODPlayLen 1282  
setVODSpeed 1587  
setVODStartTimeTC 1281  
setVODTimedTextProviderList 295  
setVODTranscodeNGRP 624, 933  
setWaitForVideoKeyFrame 1485  
setWaitOnMetadataAudio 1101, 1139  
setWaitOnMetadataAudioDatarate 1102, 1139  
setWaitOnMetadataAvailable 1101, 1139  
setWaitOnMetadataTimeout 1102  
setWaitOnMetadataVideo 1101, 1139  
setWaitOnMetadataVideoDatarate 1102, 1139  
setWidth 932  
setWowzaAudioOnly 931  
setWowzaKeyFrameOnly 934  
setWowzaVideoOnly 934  
setWowzSession 380  
SharedObject 1358, 1359  
SHAREDOBJECT\_CMD\_CONNECT 1333  
SHAREDOBJECT\_CMD\_CONNECTSUCCESS 1334  
SHAREDOBJECT\_CMD\_DELETE 1334  
SHAREDOBJECT\_CMD\_DISCONNECT 1333  
SHAREDOBJECT\_CMD\_ERROR 1334  
SHAREDOBJECT\_CMD\_SEND 1334  
SHAREDOBJECT\_CMD\_SETVALUE 1333  
SHAREDOBJECT\_STATUS\_CHANGE 1334  
SHAREDOBJECT\_STATUS\_CLEAR 1334  
SHAREDOBJECT\_STATUS\_DELETE 1334  
SHAREDOBJECT\_STATUS\_SUCCESS 1334  
shortValue 170  
shouldDeleteArchivedStream 391, 538  
shouldDvrRecord 1525  
shouldLoadArchivedStream 391, 538  
shouldLoadStream 508  
shouldLoadStreamVersion 509  
shouldStartRecordingOnStartup 1522  
shutdown 241, 270, 528, 567, 606, 636, 643, 650, 656, 866, 892, 899, 1187, 1202, 1223, 1263, 1409, 1468, 1518, 1528, 1538, 1735, 1781  
shutdownAppInstance 245  
shutdownApplication 1750  
shutdownClient 270, 357, 1220

shutdownRTPSession 1192  
shutdownSession 587  
shutdownStream 906  
sinfo 1494  
singleStreamAV 1153  
size 142, 152, 177, 197, 1336, 1346, 1359, 1434, 1471, 1706, 1710  
skipByte 132  
sloppyGetTrack 1258  
SMIL\_PARAM\_IS\_WOWZA\_CAPTION\_STREAM 1659  
SMIL\_PARAM\_WOWZA\_CAPTION\_INGEST\_TYPE 1659  
socketAddressToUniqueStr 96  
SOURCE\_TYPE\_3GPP 1640  
SOURCE\_TYPE\_AMF 1639  
SOURCE\_TYPE\_SCC 1640  
SOURCE\_TYPE\_SRT 1640  
SOURCE\_TYPE\_TTML 1639  
SOURCE\_TYPE\_WEBVTT 1640  
splitCookie 75  
splitOnTcDiscontinuity 697  
splitPragmas 74  
splitQueryStr 74, 1115  
splitRecorder 669  
splitRecording 661  
srcStream 1027  
srcStreamBufferSize 1078  
srcStreamName 1027  
stackTraceToSingleLineString 45  
stackTraceToString 45  
stampToString 100  
stampToStringNoUnits 100  
start 106, 1315, 1695  
startApplicationInstance 1752  
startAudioData 1564  
startAudioPacket 1438  
startChunk 489, 629  
startCommandInterface 1294, 1316  
startDataData 1566  
startDataPacket 1438  
startMediaCasterStream 301  
startOnKeyFrame 697  
startPlay 1471  
startPlayback 1386  
startPublishing 1432  
startRecorder 668  
startRecording 520, 532, 660, 661, 1520  
startRTPPull 1682, 1683  
startServer 1316  
startStartupStreams 1754  
startStats 1327  
startStream 896, 1518, 1529  
startsWith 39  
startTransmit 994  
startVHost 1295, 1317  
startVHosts 1295, 1317  
startVideoData 1563  
startVideoPacket 1438  
STAT\_connect\_application\_not\_available 753  
STAT\_connect\_application\_not\_found 754  
STAT\_connect\_bad\_gateway 755  
STAT\_connect\_internal\_error 754  
STAT\_connect\_license\_limit 754  
STAT\_connect\_pending\_wating 753  
STAT\_connect\_redirect 754  
STAT\_connect\_rejected\_by\_application 754  
STAT\_connect\_rejected\_by\_module 754  
STAT\_connect\_resource\_limit 754  
STAT\_connect\_service\_unavailable 755  
STAT\_connect\_successful 753  
STAT\_connect\_unknown\_protocol 754  
STAT\_general\_internal\_error 757  
STAT\_general\_successful 757  
STAT\_play\_bad\_request 755  
STAT\_play\_internal\_error 756  
STAT\_play\_rejected\_by\_application 755  
STAT\_play\_rejected\_by\_module 755  
STAT\_play\_stream\_not\_found 755  
STAT\_play\_successful 755  
STAT\_play\_unsupported\_media\_type 755  
STAT\_publish\_bad\_request 756  
STAT\_publish\_in\_use 756  
STAT\_publish\_internal\_error 756  
STAT\_publish\_rejected\_by\_application 756  
STAT\_publish\_successful 756  
STAT\_publish\_unsupported\_media\_type 756  
STAT\_stop\_client\_disconnect 756  
STAT\_stop\_successful 756  
stateChange 524  
statusCodeToStr 76  
stop 1695  
stopAdminAgent 1300, 1315

stopCommandInterface 1294, 1316  
stopMediaCasterStream 301  
stopName 1409, 1470  
stopPublishing 1432  
stopRecorder 669  
stopRecording 520, 532, 661, 1521  
stopRTPPull 1682  
stopServer 1315  
stopStartupStreams 1754  
stopStats 1327  
stopStream 896  
stopTransmit 994  
stopVHost 1295, 1316  
stopVHosts 1295, 1316  
storeChunks 522, 533  
storeOnMetadata 522, 533  
Stream 1579  
stream 1042  
streamCloseWaitTime 1077  
streamCodecToString 59  
streamExists 896, 1262  
streamId 1135  
STREAMINFO\_SESSIONATTRIBUTES 1256  
STREAMINFO\_SESSIONBANDWIDTH 1256  
STREAMINFO\_SESSIONCONNECTIONDATA 1256  
STREAMINFO\_SESSIONEMAILADDRESS 1256  
STREAMINFO\_SESSIONINFORMATION 1255  
STREAMINFO\_SESSIONNAME 1255  
STREAMINFO\_SESSIONPHONENUMBER 1256  
STREAMINFO\_SESSIONPORIGIN 1255  
STREAMINFO\_SESSIONPROTOCOLVERSION 1255  
STREAMINFO\_SESSIONREPEATTIMES 1256  
STREAMINFO\_SESSIONTIMEZONES 1256  
STREAMINFO\_SESSIONTIMING 1256  
STREAMINFO\_SESSIONURI 1255  
StreamItem 1765  
streamLicenses 1500  
StreamList 1768  
streamLock 1501  
streamMonitorAudioStartTimeout 1004  
streamMonitorAudioTCNegTolerance 1003  
streamMonitorAudioTCPoSTolerance 1003  
streamMonitorAudioTCToleranceEnable 1003  
streamMonitorAudioTimeout 1004  
streamMonitorAVSyncTolerance 1004  
streamMonitorAVSyncToleranceEnable 1004  
streamMonitorDataTCNegTolerance 1003  
streamMonitorDataTCPoSTolerance 1003  
streamMonitorDataTCToleranceEnable 1003  
streamMonitorDebug 1004  
streamMonitorResetNameGroups 1004  
streamMonitorStreamStartTimeout 1002  
streamMonitorStreamTimeout 1002  
streamMonitorVideoStartTimeout 1004  
streamMonitorVideoTCNegTolerance 1003  
streamMonitorVideoTCPoSTolerance 1003  
streamMonitorVideoTCToleranceEnable 1003  
streamMonitorVideoTimeout 1004  
streamNames 1500  
streamNamesLock 1501  
streamNameToGroup 1500  
streamNameToValidFilename 50  
streamPacketizers 1220  
StreamRecorderFileVersionDelegate 694  
StreamRecorderParameters 700  
StreamRecorderSimpleFileVersionDelegate 702  
streams 1216, 1500  
streamsOrder 1216  
streamSrcToMediaCaster 903  
streamsToPublish 1123  
STREAMTIMEOUTREASON\_GOOD 863  
STREAMTIMEOUTREASON\_MISSING 864  
STREAMTIMEOUTREASON\_NORTSPSESSION 863  
STREAMTIMEOUTREASON\_NOSESSION 863  
STREAMTIMEOUTREASON\_NOSTREAM 863  
STREAMTIMEOUTREASON\_NOTIMEOUT 863  
STREAMTIMEOUTREASON\_NOURL 863  
STREAMTIMEOUTREASON\_RECONNECTRUNNING 863  
STREAMTIMEOUTREASON\_UNKNOWN 863  
streamToFileForRead 1453  
streamToFileForWrite 1454  
streamToIndex 1504  
StreamUtils 1687  
stringToMilliseconds 100  
StringUtils 99  
suspendAllHostPorts 1751  
suspendAllVHosts 1300, 1317  
suspendCommandInterface 1316  
suspendEncoding 29

suspendServer 1300, 1317  
switchName 1408, 1470  
switchPlay 1471  
switchRecorder 668  
switchSetupToMPEGTS 1275  
SystemUtils 103

T

TAG\_APIC 780  
TAG\_COMM 787  
TAG\_LINK 780  
TAG\_PRIV 780  
TAG\_RBUF 780  
TAG\_SYLT 780  
TAG\_TALB 780  
TAG\_TBPM 781  
TAG\_TCOM 781  
TAG\_TCON 781  
TAG\_TCOP 781  
TAG\_TDEN 781  
TAG\_TDLY 781  
TAG\_TDOR 781  
TAG\_TDRC 781  
TAG\_TDRL 782  
TAG\_TDTG 782  
TAG\_TENC 782  
TAG\_TEXT 782  
TAG\_TFLT 782  
TAG\_TIPL 782  
TAG\_TIT1 782  
TAG\_TIT2 782  
TAG\_TIT3 782  
TAG\_TKEY 783  
TAG\_TLAN 783  
TAG\_TLEN 783  
TAG\_TMCL 783  
TAG\_TMED 783  
TAG\_TMOO 783  
TAG\_TOAL 783  
TAG\_TOFN 783  
TAG\_TOLY 784  
TAG\_TOPE 784  
TAG\_TOWN 784  
TAG\_TPE1 784  
TAG\_TPE2 784  
TAG\_TPE3 784  
TAG\_TPE4 784  
TAG\_TPOS 784  
TAG\_TPRO 785  
TAG\_TPUB 785  
TAG\_TRCK 785  
TAG\_TRSN 785  
TAG\_TRSO 785  
TAG\_TSOA 785  
TAG\_TSOP 785  
TAG\_TSOT 785  
TAG\_TSRC 785  
TAG\_TSSE 786  
TAG\_TSST 786  
TAG\_TXXX 787  
TAG\_UNKN 780  
TAG\_WCOM 786  
TAG\_WCOP 786  
TAG\_WOAF 786  
TAG\_WOAR 786  
TAG\_WOAS 786  
TAG\_WORS 787  
TAG\_WPAY 787  
TAG\_WPUB 787  
TAG\_WXXX 787  
TAGS\_TEXTINFORMATION 786  
TEMP\_CONSOLE\_APPENDER\_NAME 707  
TEMP\_LIST\_APPENDER\_NAME 707  
terminate 1771  
testFlashVersion 375  
testNextByte 131  
TEXTENCODING\_ISO\_8859\_1 779  
TEXTENCODING\_UTF16 779  
TEXTENCODING\_UTF16BE 779  
TEXTENCODING\_UTF8 780  
ThreadPool 1770  
triggerReadAhead 842  
TIME\_MAP\_TYPE 476  
timecodesOutOfOrderThreshold 1074  
timeCreated 1218  
TIMED\_TEXT\_TYPE\_CAPTION 1639  
timeReset 507  
TIMESTAMP\_RULE\_FORMAT 706  
TIMESTAMPFORMAT\_MILLISECONDS 807

TIMESTAMPFORMAT\_MPEG 807  
title 920  
toArray 153  
toByteBuffer 153  
toDvr 542  
toHex 43  
toLong 44  
toNTPTime 104  
toSMILString 921, 929, 937  
toString 149, 155, 171, 185, 189, 202, 211, 223, 323, 395, 397, 410, 413, 416, 418, 854, 883, 921, 929, 936, 1194, 1554, 1705, 1767, 1774  
toStringList 101  
totalIOPerformance2Last 1218  
totalIOPerformanceLast 1218  
totalPacketsSent 1077  
touch 303, 357, 605, 1223, 1280, 1374, 1518, 1530  
touchApplicationInstance 1751  
toValidFilename 50  
trait 196  
TRANSCODERTYPE\_LIVE 1537  
TRANSCODERTYPE\_VOD 1537  
transportFindBestMatch 1278  
traverseDirectory 51  
triggerAMF3Switch 136  
trim 1427, 1495  
trimTrailingZero 788  
truncatePacket 221  
tryConnect 1075  
tryToConnect 1083  
type 131, 403, 927, 1042  
TYPE\_ADMIN 1703  
TYPE\_STREAMING 1703

U

UDPAppender 716  
unbind 1202  
unbindAllHostPorts 1751  
unbindAllVHosts 1300, 1317  
uncacheRTPStream 1190  
unlock 839, 1342, 1366  
unlockRepeaterStreams 1262  
unpublish 1562  
unregisterCallback 1427  
unregisterLiveStreamTranscoder 1324  
unregisterOnPlayStatus 1428  
unregisterOnStatus 1428  
unregisterPlayer 866, 892, 903  
updateBlockMask 854  
updateIOPerformance 1317  
updateLoggingDuration 1319, 1434, 1743  
updateLoggingValues 568, 610, 637, 644, 651, 1229, 1434, 1471  
updateOnCuePointTimecode 70  
updateSDPDestination 1681  
uri 1219  
URL\_SAFE 18  
urlToId 109  
URLUtils 108  
userAgent 1217  
userExists 334  
userName 1027, 1114  
UTC\_TIME 469  
utcTime 403  
utcToDvr 542  
UTF8\_BRITISH\_POUND 1626  
UTF8\_BULLET 1603  
UTF8\_CENTS 1626  
UTF8\_CLOSING\_DOUBLE\_QUOTE 1603  
UTF8\_CLOSING\_GUILLEMETS 1603  
UTF8\_CONNECTING\_VERTICAL\_BAR 1603  
UTF8\_COPYRIGHT 1603  
UTF8\_CURRENCY 1603  
UTF8\_DASH 1618  
UTF8\_DEGREE 1603  
UTF8\_DOUBLEQUOTE 1618  
UTF8\_EM\_DASH 1603  
UTF8\_EXCLAMATION 1618  
UTF8\_GREATERTHAN 1619  
UTF8\_INVERTED\_EXCLAMATION 1604  
UTF8\_LEFT\_SINGLE\_QUOTATION 1604  
UTF8\_LEFTPAREN 1618  
UTF8\_LESSTHAN 1618  
UTF8\_LOWER\_A 1619  
UTF8\_LOWER\_A\_WITH\_CIRCUMFLEX 1625  
UTF8\_LOWER\_A\_WITH\_DIAERESIS 1604  
UTF8\_LOWER\_A\_WITH\_GRAVE 1626  
UTF8\_LOWER\_A\_WITH\_RING 1605  
UTF8\_LOWER\_A\_WITH\_TILDE 1605  
UTF8\_LOWER\_B 1619

UTF8\_LOWER\_C 1619  
UTF8\_LOWER\_D 1619  
UTF8\_LOWER\_E 1619  
UTF8\_LOWER\_E\_WITH\_CIRCUMFLEX 1625  
UTF8\_LOWER\_E\_WITH\_DIAERESIS 1605  
UTF8\_LOWER\_E\_WITH\_GRAVE 1626  
UTF8\_LOWER\_F 1620  
UTF8\_LOWER\_G 1620  
UTF8\_LOWER\_H 1620  
UTF8\_LOWER\_I 1620  
UTF8\_LOWER\_I\_WITH\_CIRCUMFLEX 1625  
UTF8\_LOWER\_I\_WITH\_DIAERESIS 1605  
UTF8\_LOWER\_I\_WITH\_GRAVE 1605  
UTF8\_LOWER\_J 1620  
UTF8\_LOWER\_K 1620  
UTF8\_LOWER\_L 1620  
UTF8\_LOWER\_M 1620  
UTF8\_LOWER\_N 1621  
UTF8\_LOWER\_O 1621  
UTF8\_LOWER\_O\_WITH\_CIRCUMFLEX 1626  
UTF8\_LOWER\_O\_WITH\_DIAERESIS 1605  
UTF8\_LOWER\_O\_WITH\_GRAVE 1605  
UTF8\_LOWER\_O\_WITH\_STROKE 1605  
UTF8\_LOWER\_O\_WITH\_TILDE 1606  
UTF8\_LOWER\_P 1621  
UTF8\_LOWER\_Q 1621  
UTF8\_LOWER\_R 1621  
UTF8\_LOWER\_S 1621  
UTF8\_LOWER\_T 1621  
UTF8\_LOWER\_U 1621  
UTF8\_LOWER\_U\_WITH\_CIRCUMFLEX 1626  
UTF8\_LOWER\_U\_WITH\_DIAERESIS 1606  
UTF8\_LOWER\_U\_WITH\_GRAVE 1606  
UTF8\_LOWER\_V 1621  
UTF8\_LOWER\_W 1622  
UTF8\_LOWER\_X 1622  
UTF8\_LOWER\_Y 1622  
UTF8\_LOWER\_Z 1622  
UTF8\_MINUSHYPHEN 1619  
UTF8\_MUSICAL\_NOTE 1626  
UTF8\_NBSP 1626  
UTF8\_NOTMATCH 1617  
UTF8\_ONE\_HALF 1627  
UTF8\_OPENING\_DOUBLE\_QUOTE 1604  
UTF8\_OPENING\_GUILLEMETS 1604  
UTF8\_PERIOD 1617  
UTF8\_PIPE 1619  
UTF8\_QUESTION 1618  
UTF8\_REGISTERED\_MARK 1604  
UTF8\_REVERSE\_QUERY 1627  
UTF8\_RIGHTPAREN 1618  
UTF8\_SERVICE\_MARK 1604  
UTF8\_SINGLEQUOTE 1617  
UTF8\_SPACE 1618  
UTF8\_TRADEMARK 1627  
UTF8\_UNDERSCORE 1618  
UTF8\_UPPER\_A 1622  
UTF8\_UPPER\_A\_WITH\_ACUTE 1606  
UTF8\_UPPER\_A\_WITH\_CIRCUMFLEX 1606  
UTF8\_UPPER\_A\_WITH\_DIAERESIS 1606  
UTF8\_UPPER\_A\_WITH\_GRAVE 1606  
UTF8\_UPPER\_A\_WITH\_RING 1606  
UTF8\_UPPER\_A\_WITH\_TILDE 1606  
UTF8\_UPPER\_B 1622  
UTF8\_UPPER\_C 1622  
UTF8\_UPPER\_C\_WITH\_CEDILLA 1607  
UTF8\_UPPER\_D 1622  
UTF8\_UPPER\_E 1623  
UTF8\_UPPER\_E\_WITH\_ACUTE 1607  
UTF8\_UPPER\_E\_WITH\_CIRCUMFLEX 1607  
UTF8\_UPPER\_E\_WITH\_DIAERESIS 1607  
UTF8\_UPPER\_E\_WITH\_GRAVE 1607  
UTF8\_UPPER\_F 1623  
UTF8\_UPPER\_G 1623  
UTF8\_UPPER\_H 1623  
UTF8\_UPPER\_I 1623  
UTF8\_UPPER\_I\_WITH\_CIRCUMFLEX 1607  
UTF8\_UPPER\_I\_WITH\_DIAERESIS 1607  
UTF8\_UPPER\_I\_WITH\_GRAVE 1607  
UTF8\_UPPER\_I\_WITH\_TILDE 1608  
UTF8\_UPPER\_J 1623  
UTF8\_UPPER\_K 1623  
UTF8\_UPPER\_L 1623  
UTF8\_UPPER\_M 1624  
UTF8\_UPPER\_N 1624  
UTF8\_UPPER\_O 1624  
UTF8\_UPPER\_O\_WITH\_ACUTE 1608  
UTF8\_UPPER\_O\_WITH\_CIRCUMFLEX 1608  
UTF8\_UPPER\_O\_WITH\_DIAERESIS 1608  
UTF8\_UPPER\_O\_WITH\_GRAVE 1608

UTF8\_UPPER\_O\_WITH\_STROKE 1608  
UTF8\_UPPER\_O\_WITH\_TILDE 1608  
UTF8\_UPPER\_P 1624  
UTF8\_UPPER\_Q 1624  
UTF8\_UPPER\_R 1624  
UTF8\_UPPER\_S 1624  
UTF8\_UPPER\_T 1624  
UTF8\_UPPER\_U 1624  
UTF8\_UPPER\_U\_WITH\_ACUTE 1608  
UTF8\_UPPER\_U\_WITH\_CIRCUMFLEX 1609  
UTF8\_UPPER\_U\_WITH\_DIAERESIS 1609  
UTF8\_UPPER\_U\_WITH\_GRAVE 1609  
UTF8\_UPPER\_V 1625  
UTF8\_UPPER\_W 1625  
UTF8\_UPPER\_X 1625  
UTF8\_UPPER\_Y 1625  
UTF8\_UPPER\_Z 1625  
UTF8\_YEN 1604  
UTF8MAP 1627  
UTF8MAP\_INDEX\_OFFSET 1627

## V

validate 840  
validateHTTPStreamerRequest 1758  
validateNewConnection 340  
validateRTSPRequest 1762  
validationFrequency 1076  
validStreamDomainToString 616  
valueOf 100, 469  
values 469  
VERSION\_FILE 686  
versionFile 51  
versioningOption 696  
vhost 327, 1216  
VHOST\_DEFAULT 1725  
VHostItem 1773  
VHostList 1777  
VHostSingleton 1780  
videoCodecId 928  
videoCodecStringToId 60  
videoCodecToString 60  
videoCodecTypeToString 94  
videoHost 1153

videoPacketizers 1220  
videoPort 1153  
VIDEOSAMPLE\_ACCESS\_ALL 354  
VIDEOSAMPLE\_ACCESS\_NONE 354  
VIDEOSAMPLEACCESS 1405  
vodTranscodeNGRP 929

## W

waitForMetadataAudio 1078  
waitForMetadataAudioDatarate 1078  
waitForMetadataAvailable 1078  
waitForMetadataReady 1078  
waitForMetadataStartTime 1078  
waitForMetadataTimeout 1078  
waitForMetadataVideo 1078  
waitForMetadataVideoDatarate 1078  
warn 728, 729  
WEEKDAY 1694  
width 928  
WMSLogger 722  
WMSLoggerFactory 731  
WMSLoggerIDs 760  
WMSProperties 321  
wowzaAudioOnly 928  
wowzaKeyFrameOnly 928  
wowzaVideoOnly 928  
WOWZDEBUGHEADERSIZE 209  
wrap 153  
write 29, 1173, 1174  
WRITE\_ACCESS\_ALL 354  
WRITE\_ACCESS\_NONE 355  
WRITEACCESS 1358, 1405  
writeAppConfig 246  
writeAppInstConfig 300  
writeChunk 64  
writeCodecConfig 1684  
writeConfig 1301, 1323  
writeDeleteError 1341, 1363  
writeDuration 67  
writeError 1363  
writeGeneratedKeyFrame 1385  
writeHeader 63  
writeHeaderFlags 71  
writePackets 66, 67, 1384, 1385, 1484

writeSDPFiles 1153  
writeSetValueError 1340, 1363  
writeShortHeader 63  
writeString 163  
writeXMLConfig 1323

## X

XMLUtils 112

## Y

YEAR 1694