



---

Wowza Media Server® 2

# Server Side API

---

# Wowza Media Server 2: Server Side API



---

Copyright © 2006 – 2010 Wowza Media Systems, Inc.  
<http://www.wowzamedia.com>

---

Copyright © 2006 – 2010 Wowza Media Systems, Inc. All rights reserved.

### **Third-Party Information**

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

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

### **Trademarks**

Wowza, Wowza Media Systems, Wowza Media Server and related logos are trademarks of Wowza Media Systems, Inc., and may be registered in the United States or in other jurisdictions including internationally.

Adobe and Flash are registered trademarks of Adobe Systems Incorporated, and may be registered in the United States or in other jurisdictions including internationally.

Silverlight is either a registered trademark or a trademark of Microsoft Corporation in the United States and/or other countries.

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

Other product names, logos, designs, titles, words, or phrases mentioned may be trademarks, service marks or trade names of other entities and may be registered in certain jurisdictions including internationally.

### **Third Party Copyright Notices**

Log4j and Mina: Copyright © 2006 The Apache Software Foundation

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

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

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

---

Package

**com.wowza.io**

## com.wowza.io Interface IRandomAccessReader

public interface **IRandomAccessReader**  
extends

### Field Summary

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

### Method Summary

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

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

## Fields

### FORWARD

```
public static final int FORWARD
```

Constant value: **1**

### REVERSE

```
public static final int REVERSE
```

Constant value: **-1**

## Methods

### init

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

Intialize RandomAccessReader

#### Parameters:

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

### open

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

Open the media asset

#### Throws:

IOException

## close

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

Close the media asset

**Throws:**

IOException

---

## isOpen

```
public boolean isOpen()
```

Is the media asset open

**Returns:**

true if media asset is open

---

## getFilePointer

```
public long getFilePointer()
```

Get the current byte location in the media asset

**Returns:**

current byte location in the media asset

---

## seek

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

Seek to a position in the media asset

**Parameters:**

pos - position to seek to

---

## read

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

Read bytes from the media asset

**Parameters:**

buf - buffer to fill

off - offset in buffer

size - size of block to read

**Returns:**

number of bytes read, -1 if failure

---

## getDirecton

```
public int getDirecton()
```

(continued on next page)

---

(continued from last page)

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

**Returns:**

current direction hint for the random access reader

---

## **setDirecton**

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

Set the current direction hint

**Parameters:**

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

---

## **getBasePath**

```
public String getBasePath()
```

Get the basePath for the random access reader

**Returns:**

basePath for the random access reader

---

## **getMediaName**

```
public String getMediaName()
```

Get the media name

**Returns:**

media name

---

## **getMediaExtension**

```
public String getMediaExtension()
```

Get the media extension

**Returns:**

media extension

---

## **exists**

```
public boolean exists()
```

Does the media asset exist

**Returns:**

true if media assets exists

---

## **lastModified**

```
public long lastModified()
```

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

**Returns:**

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

---



## length

```
public long length()
```

Get the media asset length in bytes

**Returns:**

media asset length in bytes

---

## getPath

```
public String getPath()
```

Get the unique path to the media asset item

**Returns:**

unique path to the media asset item

---

---

Package

**com.wowza.util**

## com.wowza.util Class AMFUtils

java.lang.Object

└─com.wowza.util.AMFUtils

public class **AMFUtils**  
extends Object

Utilities for the conversion between Java and AMF

### Constructor Summary

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

### Method Summary

static <a href="#">AMFData[]</a>	<a href="#">convertParams</a> (Object[] params) Converts an array of Java native data values and class to AMF data types.
----------------------------------	--

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

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

### Constructors

#### AMFUtils

public **AMFUtils**()

### Methods

#### convertParams

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

(continued from last page)

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

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

**Parameters:**

params - Array of Java

**Returns:**

Array of AMFData[] objects

## com.wowza.util Class 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

### Constructor Summary

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

### Method Summary

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

static String	<a href="#"><code>byteArrayToString</code></a> (byte[] b) Convert a byte array to a String (UTF-8 encoding assumed)
static String	<a href="#"><code>byteArrayToString</code></a> (byte[] b, int offset, int count) Convert a byte array to a String (UTF-8 encoding assumed)
static byte[]	<a href="#"><code>decodeHexString</code></a> (String hexStr) Decode a string as a byte array
static int	<a href="#"><code>doCRC32</code></a> (int crc, byte[] buffer, int offset, int len) Calculate an IEEE CRC32 value for MPEG transport stream from a starting crc value
static String	<a href="#"><code>encodeHexString</code></a> (byte[] bytes) Encode a byte array as a string
static String	<a href="#"><code>encodeHexString</code></a> (byte[] bytes, int offset, int len) Encode a byte array as a string
static int	<a href="#"><code>getUnsignedShort</code></a> ( java.nio.ByteBuffer buffer)
static byte[]	<a href="#"><code>intToByteArray</code></a> (int value) Convert a int value to a byte array in network order
static void	<a href="#"><code>intToByteArray</code></a> (int value, byte[] buffer, int offset, int size) Convert a int value to a byte array in network order
static void	<a href="#"><code>intToByteArray</code></a> (int value, byte[] buffer, int offset, int size, boolean isReverse) Convert a int value to a byte array in network order
static byte[]	<a href="#"><code>intToByteArray</code></a> (int value, int size) Convert a int value to a byte array in network order
static byte[]	<a href="#"><code>longToByteArray</code></a> (long value) Convert a long value to a byte array in network order
static void	<a href="#"><code>longToByteArray</code></a> (long value, byte[] buffer, int offset, int size) Convert a long value to a byte array in network order
static void	<a href="#"><code>longToByteArray</code></a> (long value, byte[] buffer, int offset, int size, boolean isReverse) Convert a long value to a byte array in network order
static byte[]	<a href="#"><code>longToByteArray</code></a> (long value, int size) Convert a long value to a byte array in network order

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

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

## Constructors

### BufferUtils

```
public BufferUtils()
```

(continued from last page)

## 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

**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

(continued from last page)

## 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

---

## byteArrayToLong

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

Conver byte array to long with offset

**Parameters:**

b - byte array (8 bytes)  
offset - offset

**Returns:**

long value

---

## byteArrayToLong

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

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

**Parameters:**

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

**Returns:**

long value

---



## byteArrayToLong

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

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

**Parameters:**

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

**Returns:**

long value

---

## byteArrayToInt

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

Convert byte array to int

**Parameters:**

b - byte array (4 bytes)

**Returns:**

int value

---

## byteArrayToInt

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

Conver byte array to int with offset

**Parameters:**

b - byte array (4 bytes)  
offset - offset

**Returns:**

int value

---

## byteArrayToInt

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

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

**Parameters:**

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

**Returns:**

int value

---

## byteArrayToInt

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

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

**Parameters:**

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

**Returns:**

int value

---

## byteArrayToShort

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

Convert byte array to int

**Parameters:**

b - byte array (2 bytes)

**Returns:**

int value

---

## byteArrayToShort

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

Conver byte array to int with offset

**Parameters:**

b - byte array (2 bytes)  
offset - offset

**Returns:**

int value

---

## byteArrayToShort

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

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

**Parameters:**

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

**Returns:**

(continued from last page)

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

### 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

(continued from last page)

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:

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)
```

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

---

## 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 String	<a href="#">formatBytes</a> (byte[] data) 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="#">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="#">toHex</a> (byte value) Format a byte value to a 0xff format
static String	<a href="#">toHex</a> (int value) Format a byte value to a 0xffffffff format
static String	<a href="#">toLong</a> (long value) Format a long value to a 0xffffffffffffffff format

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

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

### Constructors

(continued from last page)

## DebugUtils

```
public DebugUtils()
```

## Methods

### formatBytesShort

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

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

**Parameters:**

data - byte array

**Returns:**

formatted string

### formatBytesShort

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

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

**Parameters:**

data - byte array

offset - offset in array

len - data len

**Returns:**

formatted string

### formatBytes

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

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

**Parameters:**

data - byte array

**Returns:**

formatted string

### formatBytes

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

(continued from last page)

offset - start index in array  
len - length to format

**Returns:**

formatted string

---

## formatBytesStruct

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

---

## formatBytesStruct

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

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

**Parameters:**

data - byte array

**Returns:**

formatted string

---

## toHex

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

Format a byte value to a 0xff format

**Parameters:**

value - byte value

**Returns:**

return string

---

## toHex

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

Format a byte value to a 0xffffffff format

**Parameters:**

value - int value

**Returns:**

return string

---

## toLong

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

Format a long value to a 0xffffffffffffffff format

**Parameters:**

value - long value

---



(continued from last page)

**Returns:**

return string

## com.wowza.util Class ElapsedTimer

java.lang.Object

└─com.wowza.util.ElapsedTimer

```
public class ElapsedTimer
    extends Object
```

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

### Constructor Summary

public	<a href="#">ElapsedTimer()</a> Construct a new ElapsedTimer and start the clock
--------	--

### Method Summary

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

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

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

### Constructors

#### ElapsedTimer

```
public ElapsedTimer()
```

Construct a new ElapsedTimer and start the clock

### Methods

(continued from last page)

## getDate

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

Get the date the object was created

**Returns:**

date object created

---

## getDateString

```
public String getDateString()
```

Get the date object was created as formatted String

**Returns:**

date object created as formatted String

---

## getTime

```
public long getTime()
```

Get elapsed time object in existence (milliseconds)

**Returns:**

elapsed time (milliseconds)

---

## getTimeSeconds

```
public double getTimeSeconds()
```

Get elapsed time object in seconds

**Returns:**

elapsed time in seconds

---

## getTimeString

```
public String getTimeString()
```

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

**Returns:**

elapsed time as formatted String

---

## com.wowza.util Class FileUtils

java.lang.Object

└─com.wowza.util.FileUtils

public class **FileUtils**  
extends Object

FileUtils: File utilities

### Constructor Summary

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

### Method Summary

static void	<a href="#">copyFile</a> (java.io.File fromFile, java.io.File toFile) Simple file copy routine
static void	<a href="#">copyFile2</a> (java.io.File in, java.io.File out)
static boolean	<a href="#">deleteDirectory</a> (java.io.File path)
static String	<a href="#">streamNameToValidFilename</a> (String name) Encode a stream name (deal with path elements) to a valid filename.
static String	<a href="#">toValidFilename</a> (String name) Encode a name to a valid filename.
static void	<a href="#">traverseDirectory</a> (java.io.File dir, <a href="#">IFileProcess</a> fileNotify) Traverse a directory recursively calling fileNotify for each file and folder encountered
static java.io.File	<a href="#">versionFile</a> (java.io.File newFile) Rename a file using the file format [original-name]_#[.ext].

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

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

### Constructors

#### FileUtils

public **FileUtils**()

(continued from last page)

## Methods

### streamNameToValidFilename

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

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

**Parameters:**

name - stream name

**Returns:**

encoded filename

---

### toValidFilename

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

Encode a name to a valid filename.

**Parameters:**

name

**Returns:**

encoded filename

---

### copyFile

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

Simple file copy routine

**Parameters:**

fromFile - from file

toFile - to file

---

### copyFile2

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

---

### deleteDirectory

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

---

### versionFile

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

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

**Parameters:**

(continued from last page)

newFile - input file

**Returns:**

renamed file

---

## traverseDirectory

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

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

**Parameters:**

dir - starting directory

fileNotify - file notify object

## com.wowza.util Class FLVUtils

java.lang.Object

└─com.wowza.util.FLVUtils

public final class **FLVUtils**  
extends Object

FLVUtils: utility for reading and writing .flv files.

### Field Summary

public static final	<a href="#"><u>FLV_CHUNKHEADER_BUFFER_SIZE</u></a> Size of temporary buffer needed for flv reading (byte[]) Value: <b>13</b>
public static final	<a href="#"><u>FLV_CHUNKHEADER_FIRST_BYTE</u></a> Header values: first byte of packet data Value: <b>3</b>
public static final	<a href="#"><u>FLV_CHUNKHEADER_HEADER_SIZE</u></a> Size of packet header (byte[]) Value: <b>11</b>
public static final	<a href="#"><u>FLV_CHUNKHEADER_I_SIZE</u></a> Header values: packet size Value: <b>1</b>
public static final	<a href="#"><u>FLV_CHUNKHEADER_I_TIMECODE</u></a> Header values: timecode (milliseconds) Value: <b>2</b>
public static final	<a href="#"><u>FLV_CHUNKHEADER_I_TYPE</u></a> Header values: packet type Value: <b>0</b>
public static final	<a href="#"><u>FLV_CHUNKHEADER_SECOND_BYTE</u></a> Header values: second byte of packet data Value: <b>4</b>
public static final	<a href="#"><u>FLV_CHUNKHEADER_VALUES_SIZE</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 String	<a href="#">audioCodecToString</a> (int codec) Get a printable string representation of the audio codecs defined as IVHost.CODEC_AUDIO_*
static String	<a href="#">frameTypeToString</a> (int frameType)
static int	<a href="#">getAudioCodec</a> ( <a href="#">AMFPacket</a> packet) Get the codec id for this audio packet.
static int	<a href="#">getAudioCodec</a> (int value) Return the codec portion of the first byte of an audio packet.
static int	<a href="#">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</a> (int[] values) Given the headers values (including first byte of the packet), determine the type of video frame (FLV_*FRAME)
static long	<a href="#">getLastTC</a> (java.io.File file) Get the duration of an .flv file.
static int	<a href="#">getVideoCodec</a> ( <a href="#">AMFPacket</a> packet) Get the codec id for this video packet.



static int	<a href="#"><code>getVideoCodec</code></a> (int value) Return the codec portion of the first byte of an video packet.
static java.util.List	<a href="#"><code>interleavePackets</code></a> (java.util.List audioPackets, java.util.List videoPackets, java.util.List dataPackets, java.util.List audioTCs, java.util.List videoTCs, java.util.List dataTCs, java.util.List dataTypes, long[] currentTCs) This is a utility function primarily used for IMediaWriters.
static java.util.List	<a href="#"><code>interleavePackets</code></a> (java.util.List audioPackets, java.util.List videoPackets, java.util.List dataPackets, java.util.List audioTCs, java.util.List videoTCs, java.util.List dataTCs, long[] currentTCs) This is a utility function primarily used for IMediaWriters.
static boolean	<a href="#"><code>isAudioCodecConfig</code></a> ( <a href="#"><code>AMFPacket</code></a> packet) Returns true if the packet is a video codec config packet
static boolean	<a href="#"><code>isVideoCodecConfig</code></a> ( <a href="#"><code>AMFPacket</code></a> packet) Returns true if the packet is a video codec config packet
static boolean	<a href="#"><code>isVideoKeyFrame</code></a> ( <a href="#"><code>AMFPacket</code></a> packet) Returns true if the packet is a video key frame
static boolean	<a href="#"><code>isVideoKeyFrame</code></a> (byte[] buffer) Returns true if the packet is a video key frame
static boolean	<a href="#"><code>isVideoKeyFrame</code></a> (java.nio.ByteBuffer buffer) Returns true if the packet is a video key frame
static boolean	<a href="#"><code>isVideoKeyFrame</code></a> (int[] chunkHeaderValues) Returns true if the packet is a video key frame
static <a href="#"><code>AMFPacket</code></a>	<a href="#"><code>readChunk</code></a> (java.io.DataInput is) Read a packets worth of .flv data from an InputStream and return as an AMFPacket
static <a href="#"><code>AMFPacket</code></a>	<a href="#"><code>readChunk</code></a> (java.io.InputStream is) Read a packets worth of .flv data from an InputStream and return as an AMFPacket
static void	<a href="#"><code>readChunkHeader</code></a> (java.io.RandomAccessFile is, byte[] buffer, int[] values) Read packet header.
static boolean	<a href="#"><code>readHeader</code></a> (java.io.DataInput is) Read file header.
static boolean	<a href="#"><code>readHeader</code></a> (java.io.InputStream is) Read file header.
static void	<a href="#"><code>readPrevChunkHeader</code></a> (java.io.RandomAccessFile is, byte[] buffer, int[] values) Back up one packet from current position in the file and read the packet header.
static String	<a href="#"><code>streamCodecToString</code></a> (int codec) Get a printable string representation of the stream codecs defined as IVHost.CODEC_STREAM_*
static <a href="#"><code>AMFPacket</code></a>	<a href="#"><code>updateOnCuePointTimecode</code></a> ( <a href="#"><code>AMFPacket</code></a> packet, long timecode)
static java.nio.ByteBuffer	<a href="#"><code>updateOnCuePointTimecode</code></a> (java.nio.ByteBuffer data, int dataType, long timecode)

static String	<a href="#">videoCodecToString</a> (int codec) Get a printable string representation of the video codecs defined as IVHost.CODEC_VIDEO_*
static void	<a href="#">writeChunk</a> (java.io.DataOutput ds, java.nio.ByteBuffer data, int size, long timecode, byte type)
static void	<a href="#">writeChunk</a> (java.io.OutputStream ds, java.nio.ByteBuffer data, int size, long timecode, byte type) Write a packets worth of data.
static void	<a href="#">writeDuration</a> (java.io.File file, double duration) Write the duration to an existing .flv file.
static void	<a href="#">writeHeader</a> (java.io.OutputStream ds, double duration, java.util.Map extraMetadata) Write file header including onMetaData packet.
static void	<a href="#">writePackets</a> (java.io.OutputStream ds, java.util.List audioPackets, java.util.List videoPackets, java.util.List dataPackets, java.util.List audioTCs, java.util.List videoTCs, java.util.List dataTCs, java.util.List dataTypes, long[] currentTCs) Write a bunch of packets to .flv file all at once.
static void	<a href="#">writePackets</a> (java.io.OutputStream ds, java.util.List audioPackets, java.util.List videoPackets, java.util.List dataPackets, java.util.List audioTCs, java.util.List videoTCs, java.util.List dataTCs, java.util.List dataTypes, long[] currentTCs, <a href="#">IFLVWriterAdjustTimecode</a> dataPacketTimecodeAdjuster) Write a bunch of packets to .flv file all at once.
static void	<a href="#">writePackets</a> (java.io.OutputStream ds, java.util.List audioPackets, java.util.List videoPackets, java.util.List dataPackets, java.util.List audioTCs, java.util.List videoTCs, java.util.List dataTCs, long[] currentTCs) Write a bunch of packets to .flv file all at once.
static void	<a href="#">writePackets</a> (java.io.OutputStream ds, java.util.List packetList, long tcOffset) Write audio/video/data packets to an .flv file.
static void	<a href="#">writeShortHeader</a> ( java.io.DataOutput ds)
static void	<a href="#">writeShortHeader</a> ( java.io.OutputStream ds) Write just the FLV file header (without the metadata packet)

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

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

## Fields

### FLV\_CHUNKHEADER\_ITYPE

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

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

---

## FLV\_CHUNKHEADER\_ISIZE

```
public static final int FLV_CHUNKHEADER_ISIZE
```

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

---

## FLV\_CHUNKHEADER\_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**

---

(continued from last page)

## FLV\_KFRAME

```
public static final int FLV_KFRAME
```

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

## FLV\_DFRAME

```
public static final int FLV_DFRAME
```

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

## FLV\_PFRAME

```
public static final int FLV_PFRAME
```

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

## FLV\_TCINDEXAUDIO

```
public static final int FLV_TCINDEXAUDIO
```

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:**

(continued from last page)

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

**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

---

## videoCodecToString

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

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

**Parameters:**

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

**Returns:**codec name

---

## frameTypeToString

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

---

## 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

---

(continued from last page)

**Returns:**

AMFPacket of data or null if unsuccessful or end of file

---

**readPrevChunkHeader**

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

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

**Parameters:**

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

---

**readChunkHeader**

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

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

**Parameters:**

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

---

**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.

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

extraMetadata - Map of name/value pairs of metadata that will be appended to the onMetaData block

---

## writeChunk

```
public static void writeChunk(java.io.OutputStream ds,  
    java.nio.ByteBuffer data,  
    int size,  
    long timecode,  
    byte type)
```

Write a packets worth of data.

### Parameters:

ds - OutputStream

data - packet data

size - size of the packet

timecode - timecode (milliseconds)

type - type of packet IVHost.CONTENT\*TYPE\_\*

---

## writeChunk

```
public static void writeChunk(java.io.DataOutput ds,  
    java.nio.ByteBuffer data,  
    int size,  
    long timecode,  
    byte type)
```

---



(continued from last page)

## interleavePackets

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

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

### Parameters:

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

### Returns:

List of AMFPacket objects sorted by timecodes

## interleavePackets

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

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

### Parameters:

audioPackets - list of audio packets (ByteBuffer)  
 videoPackets - list of video packets (ByteBuffer)  
 dataPackets - list of data packets (ByteBuffer)  
 audioTCs - list of relative audio timecodes (Long)  
 videoTCs - list of relative video timecodes (Long)  
 dataTCs - list of relative data timecodes (Long)  
 dataTypes - list of integer packets types (IVHost.CONTENTTYPE\_DATA0, IVHost.CONTENTTYPE\_DATA3) - if null assumed to be IVHost.CONTENTTYPE\_DATA0  
 currentTCs - array of longs containing last TCs written for each packet type FLV\_TCINDEX\* (long[3]). Last timecode written for each packet type will be returned in this same array.

### Returns:

List of AMFPacket objects sorted by timecodes

---

## writePackets

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

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

### Parameters:

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

---

## writePackets

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

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

### Parameters:

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

---

## writePackets

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

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

### Parameters:

ds - OutputStream  
 audioPackets - list of audio packets (ByteBuffer)  
 videoPackets - list of video packets (ByteBuffer)  
 dataPackets - list of data packets (ByteBuffer)  
 audioTCs - list of relative audio timecodes (Long)  
 videoTCs - list of relative video timecodes (Long)  
 dataTCs - list of relative data timecodes (Long)  
 dataTypes - list of integer packets types (IVHost.CONTENTTYPE\_DATA0, IVHost.CONTENTTYPE\_DATA3) - if null assumed to be IVHost.CONTENTTYPE\_DATA0

---

(continued from last page)

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

---

## writePackets

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

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

### Parameters:

ds - OutputStream  
 audioPackets - list of audio packets (ByteBuffer)  
 videoPackets - list of video packets (ByteBuffer)  
 dataPackets - list of data packets (ByteBuffer)  
 audioTCs - list of relative audio timecodes (Long)  
 videoTCs - list of relative video timecodes (Long)  
 dataTCs - list of relative data timecodes (Long)  
 dataTypes - list of integer packets types (IVHost.CONTENTTYPE\_DATA0, IVHost.CONTENTTYPE\_DATA3) - if null assumed to be IVHost.CONTENTTYPE\_DATA0  
 currentTCs - array of longs containing last TCs written for each packet type FLV\_TCINDEX\* (long[3]). Last timecode written for each packet type will be returned  
 dataPacketTimecodeAdjuster - class that implements the IFLVWriterAdjustTimecode interface for adjusting timecodes in this same array.

---

## adjustFirstPacketTCs

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

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

### Parameters:

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

### Returns:

lowest of three absolute timecodes

---

## writeDuration

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

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

### Parameters:

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

## getLastTC

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

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

**Parameters:**

file

**Returns:**

duration (milliseconds)

---

## isVideoKeyFrame

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

Returns true if the packet is a video key frame

**Parameters:**

packet - packet

**Returns:**

true if is video key frame

---

## isVideoCodecConfig

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

Returns true if the packet is a video codec config packet

**Parameters:**

packet

**Returns:**

true if the packet is a video codec config packet

---

## isAudioCodecConfig

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

Returns true if the packet is a video codec config packet

**Parameters:**

packet

**Returns:**

true if the packet is a video codec config packet

---

## 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

---

(continued from last page)

**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\_\*

---

## isVideoKeyFrame

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

Returns true if the packet is a video key frame

**Parameters:**

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

**Returns:**

true if is video key frame

---

## isVideoKeyFrame

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

Returns true if the packet is a video key frame

**Parameters:**

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

**Returns:**

true if is video key frame

---

## isVideoKeyFrame

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

Returns true if the packet is a video key frame

**Parameters:**

chunkHeaderValues - chunk header values returned by FLVUtils.readChunkHeader

**Returns:**

true if is video key frame

---

## updateOnCuePointTimecode

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

## updateOnCuePointTimecode

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

## com.wowza.util Class HTTPUtils

java.lang.Object

└─com.wowza.util.HTTPUtils

public class **HTTPUtils**  
extends Object

HTTPUtils: utility class for making http requests.

### Field Summary

public static final	<a href="#">COOKIEFORMAT</a>
---------------------	------------------------------

### Constructor Summary

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

### Method Summary

static String	<a href="#">assembleQueryStr</a> (java.util.Map queryMap)
---------------	---

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)
----------------------	---

static String	<a href="#">statusCodeToStr</a> (int statusCode)
---------------	--

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

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

## Fields

### COOKIEDATEFORMAT

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

## Constructors

### HTTPUtils

`public` **HTTPUtils**()

## Methods

### HTTPRequestToFile

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

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

**Parameters:**

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

**Returns:**

true if successful

### HTTPRequestToByteArray

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

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

**Parameters:**

`inUrl` - url (will work with http and https)  
`method` - method (POST, GET)  
`data` - post data



(continued from last page)

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)
```

---

## splitQueryStr

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

---

## 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)
```

## com.wowza.util Interface IBandwidthThrottler

public interface **IBandwidthThrottler**  
extends

### Method Summary

long	<a href="#"><code>getBytesAllocation</code></a> (long request) Requests bytes from the bandwidth throttler interface.
------	--

### Methods

#### **getBytesAllocation**

public long **getBytesAllocation**(long request)

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

**Parameters:**

request - request number of bytes

**Returns:**

allocated number of bytes

## com.wowza.util Interface IFasterByteArrayDynamicStreaming

public interface **IFasterByteArrayDynamicStreaming**  
extends

### Method Summary

void	<a href="#">addAfterBlock</a> (byte[] buffer)
void	<a href="#">addBeforeBlock</a> (byte[] buffer)
java.util.List	<a href="#">getAfterBlocks</a> ()
java.util.List	<a href="#">getBeforeBlocks</a> ()

### Methods

#### **addBeforeBlock**

public void **addBeforeBlock**(byte[] buffer)

#### **addAfterBlock**

public void **addAfterBlock**(byte[] buffer)

#### **getBeforeBlocks**

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

#### **getAfterBlocks**

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

## com.wowza.util

# Class IFasterByteArrayOutputStream

```

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

```

### All Implemented Interfaces:

java.io.Flushable, java.io.Closeable

public abstract class **IFasterByteArrayOutputStream**  
 extends java.io.OutputStream

## Constructor Summary

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

## Method Summary

abstract void	<a href="#">encrypt</a> (IRTMPDecrypt encryptHandler, boolean skipFirst)
abstract int	<a href="#">getBlockSize</a> ()
abstract java.util.List	<a href="#">getBuffers</a> ()
abstract void	<a href="#">reset</a> ()
abstract void	<a href="#">setBlockSize</a> (int blockSize)
abstract int	<a href="#">size</a> ()
abstract byte[]	<a href="#">toByteArray</a> ()
abstract org.apache.mina.commo n.ByteBuffer	<a href="#">toByteBuffer</a> ()
abstract String	<a href="#">toString</a> ()
abstract String	<a href="#">toString</a> (String enc)
abstract int	<a href="#">write</a> (byte[] b, int off, int len, int poffset, int amfIndex, int chunksize)

### 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

### **IFasterByteArrayOutputStream**

```
public IFasterByteArrayOutputStream()
```

## Methods

### **write**

```
public abstract int write(byte[] b,  
    int off,  
    int len,  
    int poffset,  
    int amfIndex,  
    int chunksize)
```

---

### **size**

```
public abstract int size()
```

---

### **reset**

```
public abstract void reset()
```

---

### **toByteArray**

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

---

### **toByteBuffer**

```
public abstract org.apache.mina.common.ByteBuffer toByteBuffer()
```

(continued from last page)

---

## toString

```
public abstract String toString()
```

---

## toString

```
public abstract String toString(String enc)  
    throws java.io.UnsupportedEncodingException
```

---

## getBlockSize

```
public abstract int getBlockSize()
```

---

## setBlockSize

```
public abstract void setBlockSize(int blockSize)
```

---

## getBuffers

```
public abstract java.util.List getBuffers()
```

---

## encrypt

```
public abstract void encrypt(IRTMPEncrypt encryptHandler,  
    boolean skipFirst)
```



## com.wowza.util Interface IFasterByteArrayWriteControl

public interface **IFasterByteArrayWriteControl**  
extends

### Field Summary

public static final	<a href="#"><u>WRITECONTROL_ENHANCEDSEEK_START</u></a> Value: <b>1</b>
public static final	<a href="#"><u>WRITECONTROL_ENHANCEDSEEK_STOP</u></a> Value: <b>2</b>
public static final	<a href="#"><u>WRITECONTROL_MEDIAPACKET</u></a> Value: <b>3</b>

### Method Summary

void	<a href="#"><u>writeControl</u></a> (int src, boolean isBefore, int writeControlCode, Object extra)
------	---

### Fields

#### WRITECONTROL\_ENHANCEDSEEK\_START

public static final int **WRITECONTROL\_ENHANCEDSEEK\_START**

Constant value: **1**

#### WRITECONTROL\_ENHANCEDSEEK\_STOP

public static final int **WRITECONTROL\_ENHANCEDSEEK\_STOP**

Constant value: **2**

#### WRITECONTROL\_MEDIAPACKET

public static final int **WRITECONTROL\_MEDIAPACKET**

Constant value: **3**

### Methods

(continued from last page)

**writeControl**

```
public void writeControl(int src,  
    boolean isBefore,  
    int writeControlCode,  
    Object extra)
```

---

## com.wowza.util

# Interface IFasterByteArrayWriteReference

---

public interface **IFasterByteArrayWriteReference**  
extends

---

### Method Summary

void	<a href="#">write</a> (byte[] b, int off, int len, boolean reference)
------	---

---

### Methods

#### **write**

```
public void write(byte[] b,  
    int off,  
    int len,  
    boolean reference)
```

## com.wowza.util Interface IFastIntBuffer

public interface **IFastIntBuffer**  
extends

### Method Summary

void	<a href="#"><u>allocateBuffer()</u></a>
int	<a href="#"><u>binarySearch</u></a> (int value)
int	<a href="#"><u>binarySearch</u></a> (int value, int element)
void	<a href="#"><u>generateIndex</u></a> ()
void	<a href="#"><u>generateIndex</u></a> (int element)
int	<a href="#"><u>get</u></a> (int index)
int	<a href="#"><u>get</u></a> (int index, int element)
int	<a href="#"><u>length</u></a> ()
void	<a href="#"><u>setBufferSize</u></a> (int bufferSize)
void	<a href="#"><u>setData</u></a> (byte[] data)
void	<a href="#"><u>setDataSize</u></a> (int dataSize)
void	<a href="#"><u>setEntrySize</u></a> (int entrySize)
void	<a href="#"><u>setRandomAccessReader</u></a> ( <a href="#"><u>IRandomAccessReader</u></a> randomAccessReader)
void	<a href="#"><u>setRandomAccessReaderStartPos</u></a> (long randomAccessReaderStartPos)

### Methods

#### get

public int **get**(int index)

(continued from last page)

**get**

```
public int get(int index,  
              int element)
```

---

**binarySearch**

```
public int binarySearch(int value)
```

---

**binarySearch**

```
public int binarySearch(int value,  
                        int element)
```

---

**generateIndex**

```
public void generateIndex()
```

---

**generateIndex**

```
public void generateIndex(int element)
```

---

**length**

```
public int length()
```

---

**allocateBuffer**

```
public void allocateBuffer()
```

---

**setBufferSize**

```
public void setBufferSize(int bufferSize)
```

---

**setRandomAccessReaderStartPos**

```
public void setRandomAccessReaderStartPos(long randomAccessReaderStartPos)
```

---

(continued from last page)

## setRandomAccessReader

```
public void setRandomAccessReader(IRandomAccessReader randomAccessReader)
```

---

## setEntrySize

```
public void setEntrySize(int entrySize)
```

---

## setDataSize

```
public void setDataSize(int dataSize)
```

---

## setData

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

---

## com.wowza.util Interface IFastLongBuffer

public interface **IFastLongBuffer**  
extends

### Method Summary

void	<a href="#"><u>allocateBuffer()</u></a>
int	<a href="#"><u>binarySearch</u></a> (long value)
int	<a href="#"><u>binarySearch</u></a> (long value, int element)
void	<a href="#"><u>generateIndex()</u></a>
void	<a href="#"><u>generateIndex</u></a> (int element)
long	<a href="#"><u>get</u></a> (int index)
long	<a href="#"><u>get</u></a> (int index, int element)
int	<a href="#"><u>length()</u></a>
void	<a href="#"><u>setBufferSize</u></a> (int bufferSize)
void	<a href="#"><u>setData</u></a> (byte[] data)
void	<a href="#"><u>setDataSize</u></a> (int dataSize)
void	<a href="#"><u>setEntrySize</u></a> (int entrySize)
void	<a href="#"><u>setRandomAccessReader</u></a> ( <a href="#"><u>IRandomAccessReader</u></a> randomAccessReader)
void	<a href="#"><u>setRandomAccessReaderStartPos</u></a> (long randomAccessReaderStartPos)

### Methods

#### **get**

public long **get**(int index)

(continued from last page)

**get**

```
public long get(int index,  
               int element)
```

---

**binarySearch**

```
public int binarySearch(long value)
```

---

**binarySearch**

```
public int binarySearch(long value,  
                        int element)
```

---

**generateIndex**

```
public void generateIndex()
```

---

**generateIndex**

```
public void generateIndex(int element)
```

---

**length**

```
public int length()
```

---

**allocateBuffer**

```
public void allocateBuffer()
```

---

**setBufferSize**

```
public void setBufferSize(int bufferSize)
```

---

**setRandomAccessReaderStartPos**

```
public void setRandomAccessReaderStartPos(long randomAccessReaderStartPos)
```

---



(continued from last page)

## setRandomAccessReader

```
public void setRandomAccessReader(IRandomAccessReader randomAccessReader)
```

---

## setEntrySize

```
public void setEntrySize(int entrySize)
```

---

## setDataSize

```
public void setDataSize(int dataSize)
```

---

## setData

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

---

## com.wowza.util Interface IFileProcess

public interface **IFileProcess**  
extends

Used by FileUtils.traverseDirectory

### Method Summary

void	<a href="#">onFile</a> (java.io.File file) Triggered for each file encountered in FileUtils.traverseDirectory
------	--

### Methods

#### onFile

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

Triggered for each file encountered in FileUtils.traverseDirectory

#### Parameters:

file - file descriptor

## com.wowza.util Interface IFLVWriterAdjustTimecode

public interface **IFLVWriterAdjustTimecode**  
extends

### Method Summary

java.nio.ByteBuffer	<a href="#"><u>adjustDataTimecode</u></a> (java.nio.ByteBuffer data, int dataType, long timecode)
---------------------	---

### Methods

#### **adjustDataTimecode**

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

---

## com.wowza.util Interface IGenericPoolFactory

---

public interface **IGenericPoolFactory**  
extends

---

### Method Summary

Object	<a href="#"><u>createInstance()</u></a>
void	<a href="#"><u>destroyInstance</u></a> (Object instance)

---

### Methods

#### **createInstance**

public Object **createInstance**()

---

#### **destroyInstance**

public void **destroyInstance**(Object instance)

## com.wowza.util Class IOPerformanceCounter

java.lang.Object

└─com.wowza.util.IOPerformanceCounter

Direct Known Subclasses:

[IOPerformanceCounterDebug](#)

```
public class IOPerformanceCounter
extends Object
```

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

### Constructor Summary

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

### Method Summary

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

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

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

Methods inherited from class `java.lang.Object`

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

## Constructors

### IOPerformanceCounter

```
public IOPerformanceCounter()
```

Create an empty performance counter.

## Methods

### clone

```
public IOPerformanceCounter clone()
```

Create a deep clone (copy) if this object.

### addDifference

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

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

#### Parameters:

current - current counter

last - last counter

### clear

```
public void clear()
```

(continued from last page)

Clear or reset this counter back to zero.

---

## 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()
```

---

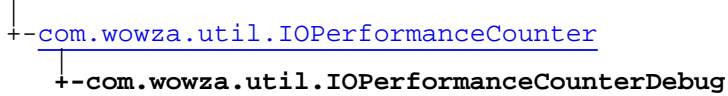
**setDebugLog**

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

---

## com.wowza.util Class IOPerformanceCounterDebug

java.lang.Object



public class **IOPerformanceCounterDebug**  
extends [IOPerformanceCounter](#)

### Constructor Summary

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

### Method Summary

void	<a href="#">add</a> ( <a href="#">IOPerformanceCounter</a> value)
void	<a href="#">addDifference</a> ( <a href="#">IOPerformanceCounter</a> current, <a href="#">IOPerformanceCounter</a> last)
void	<a href="#">doSet</a> ( <a href="#">IOPerformanceCounter</a> value)
long	<a href="#">incrementBytesIn</a> (long bytes)
long	<a href="#">incrementBytesOut</a> (long bytes)
long	<a href="#">incrementMessagesIn</a> (long bytes)
long	<a href="#">incrementMessagesIn</a> (long bytes, long count)
long	<a href="#">incrementMessagesOut</a> (long bytes)
long	<a href="#">incrementMessagesOut</a> (long bytes, long count)

#### Methods inherited from class [com.wowza.util.IOPerformanceCounter](#)

[add](#), [addDifference](#), [clear](#), [clone](#), [doSet](#), [dummy](#), [getFileInBytes](#), [getFileInBytesRate](#), [getFileOutBytes](#), [getFileOutBytesRate](#), [getMessagesInBytes](#), [getMessagesInBytesRate](#), [getMessagesInCount](#), [getMessagesInCountRate](#), [getMessagesLossBytes](#), [getMessagesLossBytesRate](#), [getMessagesLossCount](#), [getMessagesLossCountRate](#), [getMessagesOutBytes](#), [getMessagesOutBytesRate](#), [getMessagesOutCount](#), [getMessagesOutCountRate](#), [incrementBytesIn](#), [incrementBytesLoss](#), [incrementBytesOut](#), [incrementFileIn](#), [incrementFileOut](#), [incrementMessagesIn](#), [incrementMessagesIn](#), [incrementMessagesIn](#), [incrementMessagesLoss](#), [incrementMessagesLoss](#), [incrementMessagesLoss](#), [incrementMessagesOut](#), [incrementMessagesOut](#), [incrementMessagesOut](#), [isDebugLog](#), [setDebugLog](#)

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

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

## Constructors

### IOPerformanceCounterDebug

```
public IOPerformanceCounterDebug()
```

## Methods

### 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.

### add

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

Add (value) to this counter.

### doSet

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

Set this object to value.

### incrementMessagesOut

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

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

### incrementMessagesOut

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

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

### incrementBytesOut

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

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

### incrementMessagesIn

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

(continued from last page)

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

---

## **incrementMessagesIn**

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

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

---

## **incrementBytesIn**

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

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



## com.wowza.util Interface NoMBean

---

public interface **NoMBean**  
extends Annotation

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

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

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

---

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

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

---

## com.wowza.util Class StringUtils

java.lang.Object

└─com.wowza.util.StringUtils

public class **StringUtils**  
extends Object

StringUtils: utility class of String utilities.

### Constructor Summary

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

### Method Summary

static String	<a href="#">stampToString</a> (long stamp) Convert a duration (milliseconds) to a formatted string.
---------------	--

Methods inherited from class java.lang.Object

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

### Constructors

#### StringUtils

public **StringUtils**()

### Methods

#### 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)

## 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>
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>
static String	<a href="#">expandEnvironmentVariables</a> (String inValue) Expand system level Java properties in a String in the form \${property-name}
static String	<a href="#">expandEnvironmentVariables</a> (String inValue, java.util.Map valueMap) Expand system level Java properties in a String in the form \${property-name}.
static byte[]	<a href="#">getMACAddress()</a> Get MAC address of localhost interface (only works on Java 6 or greater)
static long	<a href="#">toNTPTime</a> (long t)

Methods inherited from class java.lang.Object

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

## Fields

### defaultTimeZone

```
public static final java.util.TimeZone defaultTimeZone
```

### 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()
```

## 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

(continued from last page)

## 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()
```

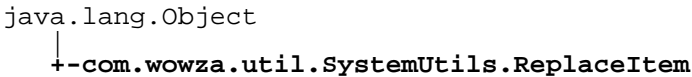
---

## toNTPTime

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

com.wowza.util

# Class SystemUtils.ReplaceItem



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

## Field Summary

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

## Constructor Summary

public	<a href="#">SystemUtils.ReplaceItem</a> (int start, int stop, String newValue)
--------	--

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

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

## Fields

### start

public int **start**

### end

public int **end**

### newValue

public java.lang.String **newValue**

## Constructors

(continued from last page)

## SystemUtils.ReplaceItem

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

## com.wowza.util Class URLUtils

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

```
public final class URLUtils
extends Object
```

Class to deal with URLs

### Constructor Summary

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

### Method Summary

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

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

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

### Constructors

#### URLUtils

```
public URLUtils()
```



(continued from last page)

## Methods

### pathToFileURL

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

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

**Parameters:**

basePath - path

**Returns:**

file url

### urlToId

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

Create a MD5 message digest hash of a url

**Parameters:**

url - url to hash

**Returns:**

hashed url

### appendParamsToUrl

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

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

**Parameters:**

url - input url

params - parameters in the form param1=val1&param2=val2

**Returns:**

appended url

### encodeValue

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

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

**Parameters:**

val - value to encode

**Returns:**

encoded value

### decodeValue

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

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

(continued from last page)

**Parameters:**

val - value to decode

**Returns:**decoded value

---

## parseQueryStr

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

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

**Parameters:**

queryString - query string

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

**Returns:**Map of Lists

---

## getParamValue

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

Helper funtion to get single value from multiple value parameter Map

**Parameters:**

params - params Map

key - key string

**Returns:**first value

---

## com.wowza.util Class XMLUtils

java.lang.Object

└─com.wowza.util.XMLUtils

public class **XMLUtils**  
extends Object

XMLUtils: utility class for parsing XML files.

### Constructor Summary

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

### Method Summary

static org.w3c.dom.Node	<a href="#">getNodeByTagName</a> (org.w3c.dom.Element node, String name) Return a child Node by tag name.
static String	<a href="#">getNodeValue</a> (org.w3c.dom.Node node) Return the text value of a node.
static String	<a href="#">getNodeValueByTagName</a> (org.w3c.dom.Element node, String name) Get a child Node value by tag name.
static boolean	<a href="#">getXMLPropertyBool</a> (javax.xml.xpath.XPath xpath, String xpathStr, org.w3c.dom.Element root, boolean defaultVal) Get a boolean int value pointed to by xpath or defaultVal if not found.
static double	<a href="#">getXMLPropertyDouble</a> (javax.xml.xpath.XPath xpath, String xpathStr, org.w3c.dom.Element root, double defaultVal) Get a single double value pointed to by xpath or defaultVal if not found.
static boolean	<a href="#">getXMLPropertyExists</a> (javax.xml.xpath.XPath xpath, String xpathStr, org.w3c.dom.Element root)
static int	<a href="#">getXMLPropertyInt</a> (javax.xml.xpath.XPath xpath, String xpathStr, org.w3c.dom.Element root, int defaultVal) Get a single int value pointed to by xpath or defaultVal if not found.
static long	<a href="#">getXMLPropertyLong</a> (javax.xml.xpath.XPath xpath, String xpathStr, org.w3c.dom.Element root, long defaultVal) Get a single long value pointed to by xpath or defaultVal if not found.
static String	<a href="#">getXMLPropertyStr</a> (javax.xml.xpath.XPath xpath, String xpathStr, org.w3c.dom.Element root) Get a single string value pointed to by xpath or null if not found.
static String	<a href="#">getXMLPropertyStr</a> (javax.xml.xpath.XPath xpath, String xpathStr, org.w3c.dom.Element root, String defaultVal) Get a single string value pointed to by xpath or defaultVal if not found.

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

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

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

## Constructors

### XMLUtils

public **XMLUtils**()

## Methods

### getNodeValueByTagName

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

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

#### Parameters:

node - parent node  
name - tag name

#### Returns:

node value or null if does not exist

### getNodeByTagName

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

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

#### Parameters:

node - parent node  
name - tag name

#### Returns:

Node or null if does not exist

(continued from last page)

## getNodeValue

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

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

**Parameters:**

node - parent node

**Returns:**

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

---

## loadConfigProperties

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

Loads <Properties> elemnt by xpath into properties object.

**Parameters:**

root - root node

propertiesXPath - xpath to search for <Properties> element

properties - properties to add values to

---

## loadConfigProperties

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

Given a nodeList load children as properties.

**Parameters:**

resultList - node list

properties - properties to add values to

---

## getXMLPropertyStr

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

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

**Parameters:**

xpath - XPath interface

xpathStr - xpath string

root - root node to start search

**Returns:**

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

---

## getXMLPropertyStr

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

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

(continued from last page)

**Parameters:**

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

**Returns:**

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

---

## getXMLPropertyExists

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

---

## getXMLPropertyInt

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

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

**Parameters:**

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

**Returns:**

int value or defaultVal if not found

---

## getXMLPropertyLong

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

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

**Parameters:**

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

**Returns:**

long value or defaultVal if not found

---

## getXMLPropertyDouble

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

(continued from last page)

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

**Parameters:**

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

**Returns:**

double value or defaultVal if not found

---

## getXMLPropertyBool

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

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

**Parameters:**

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

**Returns:**

boolean value or defaultVal if not found

---

## newXPathFactory

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

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

**Returns:**

XPathFactory or null if not found

---

---

Package

**com.wowza.wms.amf**



## com.wowza.wms.amf

### Class AMF3Utils

java.lang.Object

└─com.wowza.wms.amf.AMF3Utils

public class **AMF3Utils**  
extends Object

#### Constructor Summary

public	<a href="#">AMF3Utils()</a>
--------	-----------------------------

#### Method Summary

static java.util.Date	<a href="#">deserializeDate</a> (java.nio.ByteBuffer data)
static int	<a href="#">deserializeInt</a> (java.nio.ByteBuffer data)
static String	<a href="#">deserializeString</a> (java.nio.ByteBuffer data)
static String	<a href="#">deserializeString</a> (java.nio.ByteBuffer data, <a href="#">AMFDataContextDeserialize</a> context)
static String	<a href="#">deserializeString</a> (java.nio.ByteBuffer data, int utflen)
static int	<a href="#">serializeDate</a> (java.io.DataOutputStream out, java.util.Date date)
static int	<a href="#">serializeInt</a> (java.io.DataOutputStream out, int val)
static int	<a href="#">serializeString</a> (java.io.DataOutputStream out, String str)
static int	<a href="#">serializeStringNoLength</a> (java.io.DataOutputStream out, String str)
static void	<a href="#">serializeZeroLengthString</a> (java.io.DataOutputStream out)

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

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

#### Constructors

(continued from last page)

## AMF3Utils

```
public AMF3Utils()
```

## Methods

### deserializeDate

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

### serializeDate

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

### deserializeInt

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

### serializeInt

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

### deserializeString

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

### deserializeString

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

### deserializeString

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

(continued from last page)

### **serializeZeroLengthString**

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

---

### **serializeStringNoLength**

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

---

### **serializeString**

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

## com.wowza.wms.amf

### Class AMFData

java.lang.Object

└─com.wowza.wms.amf.AMFData

Direct Known Subclasses:

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

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

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

#### Field Summary

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

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

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

## Constructor Summary

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

## Method Summary

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

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

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

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

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

## Fields

### DATA\_TYPE\_UNKNOWN

```
public static final byte DATA_TYPE_UNKNOWN
```

Constant value: **-1**

### DATA\_TYPE\_NUMBER

```
public static final byte DATA_TYPE_NUMBER
```

Constant value: **0**

### DATA\_TYPE\_BOOLEAN

```
public static final byte DATA_TYPE_BOOLEAN
```

Constant value: **1**

### DATA\_TYPE\_STRING

```
public static final byte DATA_TYPE_STRING
```



(continued from last page)

Constant value: **2**

---

## DATA\_TYPE\_OBJECT

```
public static final byte DATA_TYPE_OBJECT
```

Constant value: **3**

---

## DATA\_TYPE\_MOVIE\_CLIP

```
public static final byte DATA_TYPE_MOVIE_CLIP
```

Constant value: **4**

---

## DATA\_TYPE\_NULL

```
public static final byte DATA_TYPE_NULL
```

Constant value: **5**

---

## DATA\_TYPE\_UNDEFINED

```
public static final byte DATA_TYPE_UNDEFINED
```

Constant value: **6**

---

## DATA\_TYPE\_REFERENCE\_OBJECT

```
public static final byte DATA_TYPE_REFERENCE_OBJECT
```

Constant value: **7**

---

## DATA\_TYPE\_MIXED\_ARRAY

```
public static final byte DATA_TYPE_MIXED_ARRAY
```

Constant value: **8**

---

## DATA\_TYPE\_OBJECT\_END

```
public static final byte DATA_TYPE_OBJECT_END
```

Constant value: **9**

---

## DATA\_TYPE\_ARRAY

```
public static final byte DATA_TYPE_ARRAY
```

Constant value: **10**

(continued from last page)

---

## DATA\_TYPE\_DATE

```
public static final byte DATA_TYPE_DATE
```

Constant value: **11**

---

## DATA\_TYPE\_LONG\_STRING

```
public static final byte DATA_TYPE_LONG_STRING
```

Constant value: **12**

---

## DATA\_TYPE\_AS\_OBJECT

```
public static final byte DATA_TYPE_AS_OBJECT
```

Constant value: **13**

---

## DATA\_TYPE\_RECORDSET

```
public static final byte DATA_TYPE_RECORDSET
```

Constant value: **14**

---

## DATA\_TYPE\_XML

```
public static final byte DATA_TYPE_XML
```

Constant value: **15**

---

## DATA\_TYPE\_CUSTOM\_CLASS

```
public static final byte DATA_TYPE_CUSTOM_CLASS
```

Constant value: **16**

---

## DATA\_TYPE\_AMF3

```
public static final byte DATA_TYPE_AMF3
```

Constant value: **17**

---

## DATA\_TYPE\_INTEGER

```
public static final byte DATA_TYPE_INTEGER
```

Constant value: **32**

---

## DATA\_TYPE\_BYTEARRAY

```
public static final byte DATA_TYPE_BYTEARRAY
```

---

(continued from last page)

Constant value: **33**

---

## DATA\_TYPE\_XML\_TOP

```
public static final byte DATA_TYPE_XML_TOP
```

Constant value: **34**

---

## DATA\_TYPE\_AMF3\_UNDEFINED

```
public static final byte DATA_TYPE_AMF3_UNDEFINED
```

Constant value: **0**

---

## DATA\_TYPE\_AMF3\_NULL

```
public static final byte DATA_TYPE_AMF3_NULL
```

Constant value: **1**

---

## DATA\_TYPE\_AMF3\_BOOLEAN\_FALSE

```
public static final byte DATA_TYPE_AMF3_BOOLEAN_FALSE
```

Constant value: **2**

---

## DATA\_TYPE\_AMF3\_BOOLEAN\_TRUE

```
public static final byte DATA_TYPE_AMF3_BOOLEAN_TRUE
```

Constant value: **3**

---

## DATA\_TYPE\_AMF3\_INTEGER

```
public static final byte DATA_TYPE_AMF3_INTEGER
```

Constant value: **4**

---

## DATA\_TYPE\_AMF3\_NUMBER

```
public static final byte DATA_TYPE_AMF3_NUMBER
```

Constant value: **5**

---

## DATA\_TYPE\_AMF3\_STRING

```
public static final byte DATA_TYPE_AMF3_STRING
```

Constant value: **6**

---

## DATA\_TYPE\_AMF3\_XML\_LEGACY

```
public static final byte DATA_TYPE_AMF3_XML_LEGACY
```

Constant value: **7**

---

## DATA\_TYPE\_AMF3\_DATE

```
public static final byte DATA_TYPE_AMF3_DATE
```

Constant value: **8**

---

## DATA\_TYPE\_AMF3\_ARRAY

```
public static final byte DATA_TYPE_AMF3_ARRAY
```

Constant value: **9**

---

## DATA\_TYPE\_AMF3\_OBJECT

```
public static final byte DATA_TYPE_AMF3_OBJECT
```

Constant value: **10**

---

## DATA\_TYPE\_AMF3\_XML\_TOP

```
public static final byte DATA_TYPE_AMF3_XML_TOP
```

Constant value: **11**

---

## DATA\_TYPE\_AMF3\_BYTEARRAY

```
public static final byte DATA_TYPE_AMF3_BYTEARRAY
```

Constant value: **12**

---

## AMF\_LEVEL0

```
public static final byte AMF_LEVEL0
```

Constant value: **0**

---

## AMF\_LEVEL3

```
public static final byte AMF_LEVEL3
```

Constant value: **3**

---

(continued from last page)

## MILLS\_PER\_HOUR

```
public static final int MILLS_PER_HOUR
```

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

## type

```
protected int type
```

## Constructors

### AMFData

```
public AMFData()
```

## Methods

### getType

```
public int getType()
```

Returns the data type for this object

**Returns:**

object type DATA\_TYPE\_\*

### setType

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

Sets the the data type for this object

**Parameters:**

type - type DATA\_TYPE\_\*

### testNextByte

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

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

**Parameters:**

data - binary data being deserialized  
test - value being tested

**Returns:**

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

### peekByte

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

(continued from last page)

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

**Parameters:**

data - binary data being deserialized

**Returns:**

next byte in buffer

---

## skipByte

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

Skip forward one byte in the byte buffer

**Parameters:**

data - binary data being deserialized

**Returns:**

next byte in buffer

---

## isObjStart

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

Returns true if next byte in data is object start

**Parameters:**

data - binary data being deserialized  
context - deserialization context

**Returns:**

Returns true if next byte in data is object start

---

## isAMF3Start

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

Returns true if next byte in data is AMF3 start

**Parameters:**

data - binary data being deserialized  
context - deserialization context

**Returns:**

Returns true if next byte in data is object start

---

## isArrayStart

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

Returns true if next byte in data is an array start

**Parameters:**

data - binary data being deserialized  
context - deserialization context

---

(continued from last page)

**Returns:**

Returns true if next byte in data is and array start

---

## isMixedArrayStart

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

Returns true if next byte in data is mixed array start

**Parameters:**

data - binary data being deserialized  
context - deserialization context

**Returns:**

Returns true if next byte in data is mixed array start

---

## isObjEnd

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

Returns true if next byte is object end

**Parameters:**

data - binary data being deserialized  
context - deserialization context

**Returns:**

Returns true if next byte in object end

---

## isByteArrayStart

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

Returns true if next byte is ByteArray start

**Parameters:**

data - binary data being deserialized  
context - deserialization context

**Returns:**

true if next byte in object end

---

## getReference

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

Get AMF3 object reference

**Parameters:**

data - binary data being deserialized  
context - deserialization context

**Returns:**

null if not a reference else referenced object

## deserializeInnerObject

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

Deserialize next item

**Parameters:**

data - binary data being deserialized  
context - deserialization context

**Returns:**

deserialized object

**Throws:**

IOException

---

## createContextSerialize

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

Create an AMF3 serialization context

**Returns:**

AMF3 serialization context

---

## createContextSerialize

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

Create an AMF3 serialization context

**Parameters:**

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

**Returns:**

AMF3 serialization context

---

## createContextDeserialize

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

Create an AMF3 deserialization context

**Returns:**

AMF3 deserialization context

---

## createContextDeserialize

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

Create an AMF3 deserialization context

**Parameters:**

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

**Returns:**



(continued from last page)

## AMF3 deserialization context

**serialize**

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

Serialize object to output stream

**Parameters:**

out - Output stream

**serialize**

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

Serialize object to output stream

**Parameters:**

out - Output stream

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

**serialize**

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

Serialize object to output stream

**Parameters:**

out - Output stream

context - serialization context used by AMF3

**serialize**

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

Serial object to byte array

**Returns:**

serialized byte array

**serialize**

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

Serial object to byte array

**Parameters:**

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

**Returns:**

serialized byte array

**serialize**

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

Serial object to byte array

---

(continued from last page)

**Parameters:**

context - serialization context used by AMF3

**Returns:**

serialized byte array

---

## deserialize

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

Deserialize data in byte buffer

**Parameters:**

data - binary data

---

## deserialize

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

Deserialize data in byte buffer

**Parameters:**

data - binary data

context - deserialization context used by AMF3

---

## getValue

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

Convert object to Java native class

**Returns:**

java native class

---

## triggerAMF3Switch

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

Return true if the object is serialized differently in AMF3

**Parameters:**

data - AMF object

**Returns:**

true if the object is serialized differently in AMF3

---

## com.wowza.wms.amf Class AMFDataArray

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

```
public class AMFDataArray
extends AMFData
```

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

### Create Array of Strings

```
AMFDataArray amfDataArray = new AMFDataArray();

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

### Iterate Items In Array

```
AMFDataArray amfDataArray;

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

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

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

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

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

## Constructor Summary

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

## Method Summary

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

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

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

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

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

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

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

## Constructors

(continued from last page)

## AMFDataArray

```
public AMFDataArray()
```

Create empty AMFDataArray object

---

## AMFDataArray

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

Deserialize entire data array and create AMFDataArray object

**Parameters:**

data - binary data

---

## AMFDataArray

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

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

**Parameters:**

data - binary data

offset - starting offset into data

size - size of data to deserialize

---

## AMFDataArray

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

Deserialize entire data array and create AMFDataArray object

**Parameters:**

data - binary data

---

## AMFDataArray

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

---

## Methods

### remove

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

Remove an item from the array

**Parameters:**

index - index

**Returns:**

delete item or null if not found

## size

```
public int size()
```

Returns the number of items in array

**Returns:**

number of items in array

---

## add

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

Append a new item onto the array

**Parameters:**

data - AMFData object

---

## add

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

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

**Parameters:**

data - string value

---

## add

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

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

**Parameters:**

data - double value

---

## add

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

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

**Parameters:**

data - int value

---

## add

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

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

**Parameters:**

data - long value

---

## add

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

---



(continued from last page)

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

**Parameters:**

data - date value

---

**add**

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

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

**Parameters:**

data - boolean value

---

**add**

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

Insert an item into the array

**Parameters:**

index - index

data - AMFData object

---

**add**

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

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

**Parameters:**

index - index

data - string value

---

**add**

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

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

**Parameters:**

index - index

data - double value

---

**add**

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

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

**Parameters:**

index - index

data - int value

---

## add

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

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

**Parameters:**

index - index  
data - long value

---

## add

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

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

**Parameters:**

index - index  
data - date value

---

## add

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

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

**Parameters:**

index - index  
data - boolean value

---

## set

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

Set an array item

**Parameters:**

index - index  
data - AMFData object

---

## set

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

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

**Parameters:**

index - index  
data - string value

---

## set

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

---

(continued from last page)

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

**Parameters:**

index - index  
data - double value

---

**set**

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

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

**Parameters:**

index - index  
data - int value

---

**set**

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

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

**Parameters:**

index - index  
data - long value

---

**set**

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

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

**Parameters:**

index - index  
data - date value

---

**set**

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

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

**Parameters:**

index - index  
data - boolean value

---

**get**

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

Get item at index

**Parameters:**

index

---

(continued from last page)

**Returns:**

Returns AMFData object or null if out of bounds

---

## getString

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

Get item at index return as String

**Parameters:**

index

**Returns:**

Return item as String or null if out of bounds

---

## getInt

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

Get item at index return as int

**Parameters:**

index

**Returns:**

Return item as int or 0 if out of bounds

---

## getLong

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

Get item at index return as long

**Parameters:**

index

**Returns:**

Return item as long or 0 if out of bounds

---

## getShort

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

Get item at index return as short

**Parameters:**

index

**Returns:**

Return item as short or 0 if out of bounds

---

## getByte

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

Get item at index return as byte

**Parameters:**

(continued from last page)

index

**Returns:**

Return item as byte or 0 if out of bounds

---

## getBoolean

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

Get item at index return as boolean

**Parameters:**

index

**Returns:**

Return item as boolean or false if out of bounds

---

## getDate

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

Get item at index return as Date

**Parameters:**

index

**Returns:**

Return item as Date or null if out of bounds

---

## getObject

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

Get item at index return as AMFDataObj

**Parameters:**

index

**Returns:**

Return item as AMFDataObj or null if out of bounds

---

## getDouble

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

Get item at index return as double

**Parameters:**

index

**Returns:**

Return item as double or 0 if out of bounds

---

## getFloat

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

Get item at index return as float

(continued from last page)

**Parameters:**

index

**Returns:**

Return item as float or 0 if out of bounds

---

**deserialize**

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

Deserialize data in byte buffer

---

**deserialize**

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

Deserialize data in byte buffer

---

**serialize**

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

Serialize object to output stream

---

**serialize**

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

Serialize object to output stream

---

**serialize**

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

Serialize object to output stream

---

**serialize**

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

Serial object to byte array

---

**serialize**

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

Serial object to byte array

---

**serialize**

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

Serial object to byte array

(continued from last page)

## **getValue**

```
public Object getValue()
```

Convert object to Java native class

---

## **toString**

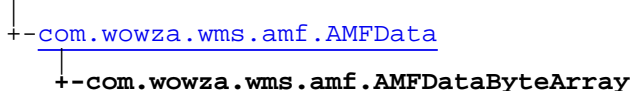
```
public String toString()
```

Return object as formatted string

## com.wowza.wms.amf

### Class AMFDataByteArray

java.lang.Object



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

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

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

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

#### Constructor Summary

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

#### Method Summary

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



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

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

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

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

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

## Constructors

(continued from last page)

## AMFDataByteArray

```
public AMFDataByteArray( )
```

Create empty AMFDataByteArray object

---

## AMFDataByteArray

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

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

**Parameters:**

data - binary data

---

## AMFDataByteArray

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

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

**Parameters:**

data - binary data

offset - starting offset into data

size - size of data to deserialize

---

## AMFDataByteArray

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

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

**Parameters:**

data - binary data

---

## AMFDataByteArray

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

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

**Parameters:**

data - binary data

context - deserialization context (used for AMF3 decoding)

## Methods

(continued from last page)

## size

```
public int size()
```

Returns the number of bytes in the byte array

**Returns:**

number of bytes in the array

---

## toArray

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

Returns the underlying data buffer (not a copy)

**Returns:**

data buffer (not a copy)

---

## toByteBuffer

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

Wraps the underlying data buffer with a ByteBuffer object.

**Returns:**

byte[] wrapped as ByteBuffer

---

## wrap

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

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

**Parameters:**

data - byte[] data

**Returns:**

wrapped byte[]

---

## wrap

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

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

**Parameters:**

data - ByteBuffer data

**Returns:**

wrapped ByteBuffer

---

## getValue

```
public Object getValue()
```

Returns the underlying byte[] data buffer

## deserialize

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

Deserialize data in byte buffer

---

## deserialize

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

Deserialize data in byte buffer

---

## serialize

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

Serialize object to output stream

---

## serialize

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

Serialize object to output stream

---

## serialize

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

Serialize object to output stream

---

## serialize

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

Serial object to byte array

---

## serialize

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

Serial object to byte array

---

## serialize

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

Serial object to byte array

---

## compress

```
public int compress()
```

Compress the internal buffer using the ZLIB compression library

**Returns:**

---

(continued from last page)

size of compressed buffer

---

## decompress

```
public int decompress()
```

Decompress the internal buffer using the ZLIB compression library

**Returns:**

size of decompressed buffer

---

## toString

```
public String toString()
```

Return object as formatted string

## com.wowza.wms.amf

### Class AMFDataContextDeserialize

java.lang.Object

└─com.wowza.wms.amf.AMFDataContextDeserialize

public class **AMFDataContextDeserialize**  
extends Object

#### Constructor Summary

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

#### Method Summary

void	<a href="#">addObject</a> (Object obj)
void	<a href="#">addString</a> (String str)
void	<a href="#">addTrait</a> ( <a href="#">AMFDataTrait</a> obj)
int	<a href="#">clearIntData</a> ()
int	<a href="#">getIntData</a> ()
Object	<a href="#">getObject</a> (int index)
int	<a href="#">getObjectEncoding</a> ()
String	<a href="#">getString</a> (int index)
<a href="#">AMFDataTrait</a>	<a href="#">getTrait</a> (int index)
boolean	<a href="#">isAMF0</a> ()
boolean	<a href="#">isAMF3</a> ()
boolean	<a href="#">isIntData</a> ()
void	<a href="#">setIntData</a> (int intData)
void	<a href="#">setObjectEncoding</a> (int objectEncoding)

Methods inherited from class java.lang.Object

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

---

## Constructors

### AMFDataContextDeserialize

```
public AMFDataContextDeserialize()
```

---

### AMFDataContextDeserialize

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

---

## Methods

### isIntData

```
public boolean isIntData()
```

---

### setIntData

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

---

### getIntData

```
public int getIntData()
```

---

### clearIntData

```
public int clearIntData()
```

---

### getObjectEncoding

```
public int getObjectEncoding()
```

---

### setObjectEncoding

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

---

---

## isAMF3

```
public boolean isAMF3()
```

---

---

## isAMF0

```
public boolean isAMF0()
```

---

---

## addString

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

---

---

## getString

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

---

---

## addObject

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

---

---

## getObject

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

---

---

## addTrait

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

---

---

## getTrait

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

---



## com.wowza.wms.amf

### Class AMFDataContextSerialize

java.lang.Object

└─com.wowza.wms.amf.AMFDataContextSerialize

public class **AMFDataContextSerialize**  
extends Object

#### Constructor Summary

public	<a href="#">AMFDataContextSerialize()</a>
public	<a href="#">AMFDataContextSerialize(int objectEncoding)</a>

#### Method Summary

int	<a href="#">getObjectEncoding()</a>
int	<a href="#">getObjectReference(Object obj)</a>
int	<a href="#">getStringReference(String str)</a>
int	<a href="#">getTargetEncoding()</a>
int	<a href="#">getTraitReference(AMFDataTrait obj)</a>
boolean	<a href="#">isAMF0()</a>
boolean	<a href="#">isAMF3()</a>
void	<a href="#">setObjectEncoding(int objectEncoding)</a>
void	<a href="#">setTargetEncoding(int targetEncoding)</a>
void	<a href="#">writeString(java.io.DataOutputStream out, String str)</a>

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

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

#### Constructors

(continued from last page)

---

## AMFDataContextSerialize

```
public AMFDataContextSerialize()
```

---

## AMFDataContextSerialize

```
public AMFDataContextSerialize(int objectEncoding)
```

## Methods

### getObjectEncoding

```
public int getObjectEncoding()
```

---

### setObjectEncoding

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

---

### getTargetEncoding

```
public int getTargetEncoding()
```

---

### setTargetEncoding

```
public void setTargetEncoding(int targetEncoding)
```

---

### isAMF3

```
public boolean isAMF3()
```

---

### isAMF0

```
public boolean isAMF0()
```

---

### getStringReference

```
public int getStringReference(String str)
```

---

(continued from last page)

## getObjectReference

```
public int getObjectReference(Object obj)
```

---

## getTraitReference

```
public int getTraitReference(AMFDataTrait obj)
```

---

## writeString

```
public void writeString(java.io.DataOutputStream out,  
                        String str)
```

## com.wowza.wms.amf Class AMFDataItem

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

```
public class AMFDataItem
extends AMFData
```

AMFDataItem: class for marshalling data between Wowza Pro server and Flash client. The type wraps native Java data types.

- DATA\_TYPE\_NUMBER = int, long, short, double, float
- DATA\_TYPE\_STRING = String
- DATA\_TYPE\_BOOLEAN = boolean
- DATA\_TYPE\_DATE = Date
- DATA\_TYPE\_NULL = 'null'

### Create Native Java Types

```
AMFDataItem amfDataString = new AMFDataItem("here is my string"); // String
AMFDataItem amfDataLong = new AMFDataItem(1234L); // long
AMFDataItem amfDataDouble = new AMFDataItem(1.234); // double
AMFDataItem amfDataBoolean = new AMFDataItem(true); // boolean
AMFDataItem amfDataNull = new AMFDataItem(); // null
```

### Get Native Java Types

```
String dataString = amfDataString.getType()==AMFData.DATA_TYPE_LONG_STRING?
    amfDataString.toString():"";

long dataLong = amfDataLong.getType()==AMFData.DATA_TYPE_NUMBER?
    amfDataLong.longValue():0;

double dataDouble = amfDataDouble.getType()==AMFData.DATA_TYPE_NUMBER?
    amfDataLong.doubleValue():0.0;

boolean dataBoolean = amfDataBoolean.getType()==AMFData.DATA_TYPE_BOOLEAN?
    amfDataLong.booleanValue():false;

Object dataNull = amfDataNull.getType()==AMFData.DATA_TYPE_NULL?
    null:null;
```

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

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

## Constructor Summary

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

## Method Summary

boolean	<a href="#">booleanValue</a> () Return object as boolean.
byte	<a href="#">byteValue</a> () Return object as byte.

java.util.Date	<a href="#"><u>dateValue()</u></a> Return object as Date.
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)
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()</u></a> ( <a href="#"><u>AMFDataContextSerialize</u></a> context)
void	<a href="#"><u>serialize()</u></a> (java.io.DataOutputStream out)
void	<a href="#"><u>serialize()</u></a> (java.io.DataOutputStream out, <a href="#"><u>AMFDataContextSerialize</u></a> context)
void	<a href="#"><u>serialize()</u></a> (java.io.DataOutputStream out, int objectEncoding)
byte[]	<a href="#"><u>serialize()</u></a> (int objectEncoding)
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](#), [createContextDeserialize](#), [createContextSerialize](#), [createContextSerialize](#), [deserialize](#), [deserialize](#), [deserializeInnerObject](#), [getReference](#), [getType](#), [getValue](#), [isAMF3Start](#), [isArrayStart](#), [isByteArrayStart](#), [isMixedArrayStart](#), [isObjEnd](#), [isObjStart](#), [peekByte](#), [serialize](#), [serialize](#), [serialize](#), [serialize](#), [serialize](#), [serialize](#), [set](#), [setType](#), [skipByte](#), [testNextByte](#), [triggerAMF3Switch](#)

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

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

## 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)
```

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

---

(continued from last page)

## 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)
```

Deserialize entire data array and create AMFDataItem object

**Parameters:**

data - binary data

---

## AMFDataItem

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

---

## Methods

### longValue

```
public long longValue()
```

Return object as long. Valid object types are DATA\_TYPE\_NUMBER and DATA\_TYPE\_STRING.

**Returns:**

long value or 0 if failure



(continued from last page)

## 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()
```

(continued from last page)

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

---

(continued from last page)

**serialize**

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

Serial object to byte array

---

**serialize**

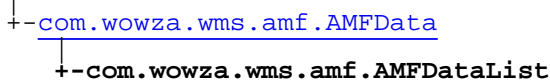
```
public byte[] serialize(AMFDataContextSerialize context)
```

Serial object to byte array

## com.wowza.wms.amf

### Class AMFDataList

java.lang.Object



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

AMFDataItem: class for marshalling data between Wowza Pro server and Flash client. This class is for internal server marshalling of AMF event messages between the Flash client and Wowza Pro server.

**NOTE:** There is a slight difference between this class and AMFDataArray. This class when serialized/deserialized does NOT include the DATA\_TYPE\_ARRAY header (byte) and array size (int). AMF formatted functions use this class since they do not include these elements.

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

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

### Constructor Summary

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

### Method Summary

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

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

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

void	<a href="#">set</a> (int index, int data) Set an int value (will be wrapped in an AMFDataItem object)
void	<a href="#">set</a> (int index, long data) Set an long value (will be wrapped in an AMFDataItem object)
void	<a href="#">set</a> (int index, String data) Set an string value (will be wrapped in an AMFDataItem object)
int	<a href="#">size</a> () Returns the number of items in array
String	<a href="#">toString</a> () Return object as formatted string

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

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

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

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

## Constructors

### AMFDataList

```
public AMFDataList()
```

Create empty AMFDataList object

### AMFDataList

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

Deserialize entire data array and create AMFDataList object

#### Parameters:

data - binary data

### AMFDataList

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

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

#### Parameters:

data - binary data

offset - starting offset into data

(continued from last page)

size - size of data to deserialize

---

## AMFDataList

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

Deserialize entire data array and create AMFDataList object

**Parameters:**

data - binary data

---

## AMFDataList

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

## Methods

### remove

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

Remove an element from the AMFDataList object

**Parameters:**

index - index

**Returns:**

removed element

---

### size

```
public int size()
```

Returns the number of items in array

**Returns:**

number of items in array

---

### add

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

Append a new item onto the array

**Parameters:**

data - AMFData object

---

### add

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

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

**Parameters:**

data - string value

---



## add

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

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

**Parameters:**

data - double value

---

## add

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

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

**Parameters:**

data - int value

---

## add

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

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

**Parameters:**

data - long value

---

## add

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

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

**Parameters:**

data - date value

---

## add

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

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

**Parameters:**

data - boolean value

---

## add

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

Insert an item into the array

**Parameters:**

index - index

data - AMFData object

---

(continued from last page)

---

## add

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

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

### Parameters:

index - index  
data - string value

---

## add

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

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

### Parameters:

index - index  
data - double value

---

## add

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

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

### Parameters:

index - index  
data - int value

---

## add

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

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

### Parameters:

index - index  
data - long value

---

## add

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

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

### Parameters:

index - index  
data - date value

---

## add

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

---

(continued from last page)

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

**Parameters:**

index - index  
data - boolean value

---

**set**

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

Set an array item

**Parameters:**

data - AMFData object

---

**set**

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

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

**Parameters:**

index - index  
data - string value

---

**set**

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

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

**Parameters:**

index - index  
data - double value

---

**set**

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

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

**Parameters:**

index - index  
data - int value

---

**set**

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

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

**Parameters:**

index - index  
data - long value

## set

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

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

### Parameters:

index - index  
data - date value

---

## set

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

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

### Parameters:

index - index  
data - boolean value

---

## getType

```
public int getType(int index)
```

Get type of item at index. Return AMFData.DATA\_TYPE\_UNKNOWN if item does not exist

### Parameters:

index

### Returns:

type of item at index

---

## get

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

Get item at index

### Parameters:

index

### Returns:

Returns AMFData object or null if out of bounds

---

## getString

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

Get item at index return as String

### Parameters:

index

### Returns:

Return item as String or null if out of bounds

---

## getInt

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

Get item at index return as int

**Parameters:**

index

**Returns:**

Return item as int or 0 if out of bounds

---

## getLong

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

Get item at index return as long

**Parameters:**

index

**Returns:**

Return item as long or 0 if out of bounds

---

## getDouble

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

Get item at index return as double

**Parameters:**

index

**Returns:**

Return item as double or 0 if out of bounds

---

## getFloat

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

Get item at index return as float

**Parameters:**

index

**Returns:**

Return item as float or 0 if out of bounds

---

## getShort

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

Get item at index return as short

**Parameters:**

index

---

(continued from last page)

**Returns:**

Return item as short or 0 if out of bounds

---

## getBytes

```
public byte getBytes(int index)
```

Get item at index return as byte

**Parameters:**

index

**Returns:**

Return item as byte or 0 if out of bounds

---

## getBoolean

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

Get item at index return as boolean

**Parameters:**

index

**Returns:**

Return item as boolean or false if out of bounds

---

## getDate

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

Get item at index return as Date

**Parameters:**

index

**Returns:**

Return item as Date or null if out of bounds

---

## getObject

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

Get item at index return as AMFDataObj

**Parameters:**

index

**Returns:**

Return item as AMFDataObj or null if out of bounds

---

## deserialize

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

Deserialize data in byte buffer

(continued from last page)

## deserialize

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

Deserialize data in byte buffer

---

## serialize

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

Serialize object to output stream

---

## serialize

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

Serialize object to output stream

---

## serialize

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

Serialize object to output stream

---

## serialize

```
public void serialize(java.io.DataOutputStream out,  
    AMFDataContextSerialize context,  
    byte[] prepend)
```

## serialize

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

Serial object to byte array

---

## serialize

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

Serial object to byte array

---

## serialize

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

Serial object to byte array

---

## serialize

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

---

## getValue

```
public Object getValue()
```

Convert object to Java native class

---

## toString

```
public String toString()
```

Return object as formatted string



## com.wowza.wms.amf

### Class AMFDataMixedArray

```

java.lang.Object
├── com.wowza.wms.amf.AMFData
│   ├── com.wowza.wms.amf.AMFDataObj
│   └── com.wowza.wms.amf.AMFDataMixedArray

```

```

public class AMFDataMixedArray
extends AMFDataObj

```

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

### Create Array of Strings

```

AMFDataMixedArray amfDataMixedArray = new AMFDataMixedArray();

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

```

### Iterate Mixed Array

```

AMFDataMixedArray amfDataMixedArray;

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

```

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

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

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

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

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

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

## Constructor Summary

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

## Method Summary

void	<a href="#">deserialize</a> (java.nio.ByteBuffer data)
void	<a href="#">deserialize</a> (java.nio.ByteBuffer data, <a href="#">AMFDataContextDeserialize</a> context)
void	<a href="#">serialize</a> (java.io.DataOutputStream out)
void	<a href="#">serialize</a> (java.io.DataOutputStream out, <a href="#">AMFDataContextSerialize</a> context)
void	<a href="#">serialize</a> (java.io.DataOutputStream out, int objectEncoding)
String	<a href="#">toString()</a> Return object as formatted string

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

[containsKey](#), [deserialize](#), [deserialize](#), [get](#), [get](#), [getBoolean](#), [getBoolean](#), [getBytes](#), [getBytes](#), [getClassName](#), [getDate](#), [getDate](#), [getDouble](#), [getDouble](#), [getFloat](#), [getFloat](#), [getInt](#), [getInt](#), [getKey](#), [getKeys](#), [getLong](#), [getLong](#), [getObject](#), [getObject](#), [getShort](#), [getShort](#), [getString](#), [getString](#), [getTrait](#), [getValue](#), [put](#), [put](#), [put](#), [put](#), [put](#), [put](#), [put](#), [remove](#), [remove](#), [serialize](#), [serialize](#), [serialize](#), [serialize](#), [serialize](#), [serialize](#), [setClassName](#), [size](#), [toString](#)

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

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

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

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

## 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

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

(continued from last page)

**Parameters:**

data - binary data

---

## AMFDataMixedArray

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

## Methods

### deserialize

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

Deserialize data in byte buffer

---

### deserialize

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

Deserialize data in byte buffer

---

### serialize

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

Serialize object to output stream

---

### serialize

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

Serialize object to output stream

---

### serialize

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

Serialize object to output stream

---

### toString

```
public String toString()
```

Return object as formatted string

## com.wowza.wms.amf

### Class AMFDataObj

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

Direct Known Subclasses:  
[AMFDataMixedArray](#)

```
public class AMFDataObj
extends AMFData
```

AMFDataObj: class for marshalling data between Wowza Pro server and Flash client. Object with attributes. Implementation is very similar to a java.util.Map. Each parameter is an item in the map.

### Create AMFDataObj

```
AMFDataObj amfDataObj = new AMFDataObj();

amfDataObj.put("key1", "item1");
amfDataObj.put("key2", "item2");
amfDataObj.put("key3", "item3");
```

### Iterate AMFDataObj

```
AMFDataObj amfDataObj;

List keys = amfDataObj.getKeys();
Iterator iter = keys.iterator();
while(iter.hasNext())
{
    String key = (String)iter.next();
    AMFData value = amfDataObj.get(key);
    int itemType = value.getType();
    WMSLoggerFactory.getLogger(null).debug(key+"="+value.toString()+"
(type:"+itemType+")");
}
```

### Direct Access To Attributes

```

AMFDataObj amfDataObj;

// If you know the type you can access it directly
String dataString = amfDataObj.getString("stringData");
long dataLong = amfDataObj.getLong("longData");
double dataDouble = amfDataObj.getDouble("doubleData");
boolean dataBoolean = amfDataObj.getBoolean("booleanData");

// This illustrate how to decode the value if
// you don't know the type
AMFData myItemKey1 = amfDataObj.get("theData");
switch (myItemKey1.getType())
{
default:
case AMFDataItem.DATA_TYPE_UNDEFINED:
case AMFDataItem.DATA_TYPE_UNKNOWN:
case AMFDataItem.DATA_TYPE_NULL:
    // the value is null or undefined
    break;
case AMFDataItem.DATA_TYPE_NUMBER:
    double amfDataDouble = ((AMFDataItem)myItemKey1).doubleValue();
    break;
case AMFDataItem.DATA_TYPE_BOOLEAN:
    boolean amfDataBoolean = ((AMFDataItem)myItemKey1).booleanValue();
    break;
case AMFDataItem.DATA_TYPE_STRING:
    String amfDataString = ((AMFDataItem)myItemKey1).toString();
    break;
case AMFDataItem.DATA_TYPE_DATE:
    Date amfDataDate = ((AMFDataItem)myItemKey1).dateValue();
    break;
case AMFDataItem.DATA_TYPE_OBJECT:
    AMFDataObj amfDataValObj = (AMFDataObj)myItemKey1;
    break;
case AMFDataItem.DATA_TYPE_MIXED_ARRAY:
    AMFDataMixedArray amfDataMixedArray = (AMFDataMixedArray)myItemKey1;
    break;
case AMFDataItem.DATA_TYPE_ARRAY:
    AMFDataArray amfDataArray = (AMFDataArray)myItemKey1;
    break;
}

```

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

## Field Summary

public static final	<a href="#">DECODE_OBJ_REF</a> Value: <b>1</b>
public static final	<a href="#">DECODE_TRAITS</a> Value: <b>4</b>
public static final	<a href="#">DECODE_TRAITS_EXT</a> Value: <b>3</b>
public static final	<a href="#">DECODE_TRAITS_REF</a> Value: <b>2</b>
public static final	<a href="#">DECODE_UNDEFINED</a> Value: <b>0</b>
protected	<a href="#">members</a>
protected	<a href="#">order</a>
protected	<a href="#">trait</a>

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

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

## Constructor Summary

public	<a href="#">AMFDataObj()</a> Create empty AMFDataObj object
public	<a href="#">AMFDataObj(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(java.nio.ByteBuffer data, <a href="#">AMFDataContextDeserialize</a> context)</a>

## Method Summary

boolean	<a href="#"><u>containsKey</u></a> (String name) Return true if the object/array contains key
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) Return the object at a particular index.
<a href="#"><u>AMFData</u></a>	<a href="#"><u>get</u></a> (String name) Return the object at a particular key.
boolean	<a href="#"><u>getBoolean</u></a> (int index) Get item at index return as boolean
boolean	<a href="#"><u>getBoolean</u></a> (String name) Get item at key return as boolean
byte	<a href="#"><u>getBytes</u></a> (int index) Get item at index return as byte
byte	<a href="#"><u>getBytes</u></a> (String name) Get item at key return as byte
String	<a href="#"><u>getClassName</u></a> ( )
java.util.Date	<a href="#"><u>getDate</u></a> (int index) Get item at index return as Date
java.util.Date	<a href="#"><u>getDate</u></a> (String name) Get item at key return as Date
double	<a href="#"><u>getDouble</u></a> (int index) Get item at index return as double
double	<a href="#"><u>getDouble</u></a> (String name) Get item at key return as double
float	<a href="#"><u>getFloat</u></a> (int index) Get item at index return as float
float	<a href="#"><u>getFloat</u></a> (String name) Get item at key return as float
int	<a href="#"><u>getInt</u></a> (int index) Get item at index return as int
int	<a href="#"><u>getInt</u></a> (String name) Get item at key return as int
String	<a href="#"><u>getKey</u></a> (int index) Return the key at a particular index.
java.util.List	<a href="#"><u>getKeys</u></a> ( ) Return a list of all the keys (the list is a copy)



long	<a href="#"><u>getLong</u></a> (int index) Get item at index return as long
long	<a href="#"><u>getLong</u></a> (String name) Get item at key return as long
<a href="#"><u>AMFDataObj</u></a>	<a href="#"><u>getObject</u></a> (int index) Get item at index return as AMFDataObj
<a href="#"><u>AMFDataObj</u></a>	<a href="#"><u>getObject</u></a> (String name) Get item at key return as AMFDataObj
short	<a href="#"><u>getShort</u></a> (int index) Get item at index return as short
short	<a href="#"><u>getShort</u></a> (String name) Get item at key return as short
String	<a href="#"><u>getString</u></a> (int index) Get item at index return as String
String	<a href="#"><u>getString</u></a> (String name) Get item at key return as String
<a href="#"><u>AMFDataTrait</u></a>	<a href="#"><u>getTrait</u></a> ()
Object	<a href="#"><u>getValue</u></a> ()
void	<a href="#"><u>put</u></a> (String name, <a href="#"><u>AMFData</u></a> data) Put or replace object at key
void	<a href="#"><u>put</u></a> (String name, boolean data) Put or replace boolean value at key (data will be wrapped in an AMFDataItem object)
void	<a href="#"><u>put</u></a> (String name, java.util.Date data) Put or replace date value at key (data will be wrapped in an AMFDataItem object)
void	<a href="#"><u>put</u></a> (String name, double data) Put or replace double value at key (data will be wrapped in an AMFDataItem object)
void	<a href="#"><u>put</u></a> (String name, int data) Put or replace int value at key (data will be wrapped in an AMFDataItem object)
void	<a href="#"><u>put</u></a> (String name, long data) Put or replace long value at key (data will be wrapped in an AMFDataItem object)
void	<a href="#"><u>put</u></a> (String name, String data) Put or replace string value at key (data will be wrapped in an AMFDataItem object)
<a href="#"><u>AMFData</u></a>	<a href="#"><u>remove</u></a> (int index) Remove element by index
<a href="#"><u>AMFData</u></a>	<a href="#"><u>remove</u></a> (String name) Remove element by key
byte[]	<a href="#"><u>serialize</u></a> ()
byte[]	<a href="#"><u>serialize</u></a> ( <a href="#"><u>AMFDataContextSerialize</u></a> context)

void	<a href="#">serialize</a> (java.io.DataOutputStream out)
void	<a href="#">serialize</a> (java.io.DataOutputStream out, <a href="#">AMFDataContextSerialize</a> context)
void	<a href="#">serialize</a> (java.io.DataOutputStream out, int objectEncoding)
byte[]	<a href="#">serialize</a> (int objectEncoding)
void	<a href="#">setClassName</a> (String className)
int	<a href="#">size</a> () Return the number of members of this object/array
String	<a href="#">toString</a> () Return object as formatted string

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

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

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

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

## Fields

### 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
```

## Constructors

### AMFDataObj

```
public AMFDataObj()
```

Create empty AMFDataObj object

### AMFDataObj

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

Deserialize entire data array and create AMFDataObj object

#### Parameters:

data - binary data

### AMFDataObj

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

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

#### Parameters:

data - binary data

(continued from last page)

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

---

## AMFDataObj

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

Deserialize entire data array and create AMFDataObj object

**Parameters:**

data - binary data

---

## AMFDataObj

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

## Methods

### size

```
public int size()
```

Return the number of members of this object/array

**Returns:**

number of members

---

### containsKey

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

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)

---

(continued from last page)

**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)
```

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

---

## 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

---

(continued from last page)

**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:**



(continued from last page)

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

**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

(continued from last page)

**Parameters:**

index - index

**Returns:**

Return item as short or 0 if out of bounds

---

## getBytes

```
public byte getBytes(int index)
```

Get item at index return as byte

**Parameters:**

index - 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:**

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)
```

(continued from last page)

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

---

## deserialize

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

Deserialize data in byte buffer

---

## deserialize

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

Deserialize data in byte buffer

---

## serialize

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

Serialize object to output stream

---

## serialize

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

Serialize object to output stream

---

## serialize

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

Serialize object to output stream

---

## serialize

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

Serial object to byte array

---

## serialize

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

Serial object to byte array

---

## serialize

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

Serial object to byte array

---

## getValue

```
public Object getValue()
```

Convert object to Java native class

---

## toString

```
public String toString()
```

Return object as formatted string

---

## getClassName

```
public String getClassName()
```

---

## setClassName

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

---

## getTrait

```
public AMFDataTrait getTrait()
```

---

## com.wowza.wms.amf

### Class AMFDataTrait

java.lang.Object

└─com.wowza.wms.amf.AMFDataTrait

public class **AMFDataTrait**  
extends Object

#### Constructor Summary

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

#### Method Summary

void	<a href="#">addMember</a> (String member)
<a href="#">AMFDataTrait</a>	<a href="#">clone</a> ()
String	<a href="#">getClassName</a> ()
<a href="#">AMFData</a>	<a href="#">getInnerObj</a> ()
String	<a href="#">getMember</a> (int i)
int	<a href="#">getMemberCount</a> ()
java.util.List	<a href="#">getMembers</a> ()
boolean	<a href="#">isDynamic</a> ()
boolean	<a href="#">isMember</a> (String member)
void	<a href="#">setClassName</a> (String className)
void	<a href="#">setDynamic</a> (boolean isDynamic)
void	<a href="#">setInnerObj</a> ( <a href="#">AMFData</a> innerObj)

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

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

#### Constructors

(continued from last page)

## AMFDataTrait

```
public AMFDataTrait()
```

## Methods

### clone

```
public AMFDataTrait clone()
```

---

### addMember

```
public void addMember(String member)
```

---

### isMember

```
public boolean isMember(String member)
```

---

### getMembers

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

---

### getMemberCount

```
public int getMemberCount()
```

---

### getMember

```
public String getMember(int i)
```

---

### getClassName

```
public String getClassName()
```

---

### setClassName

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

---

(continued from last page)

## isDynamic

```
public boolean isDynamic()
```

---

## setDynamic

```
public void setDynamic(boolean isDynamic)
```

---

## getInnerObj

```
public AMFData getInnerObj()
```

---

## setInnerObj

```
public void setInnerObj(AMFData innerObj)
```

---

## 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.

### Constructor Summary

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

### Method Summary

long	<a href="#">getAbsTimecode()</a> Get the absolute time code
int	<a href="#">getByteContainerLevel()</a> Fake container for processing
int	<a href="#">getId()</a> Get channel id
int	<a href="#">getObjectEncoding()</a>
int	<a href="#">getSize()</a> Get packet size
int	<a href="#">getSrc()</a> Get stream id (0 if not stream data)
long	<a href="#">getTimecode()</a> Get timecode (milliseconds) sometimes relative
int	<a href="#">getType()</a> Get content type IVHost.CONTENTTYPE_*
long	<a href="#">incAbsTimecode(long absTimecode)</a> Increment the absolution timecode
void	<a href="#">incByteContainerLevel(int byteContainerLevel)</a> Fake container for processing
boolean	<a href="#">isByteContainerEmpty()</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>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(int objectEncoding)</u></a>
void	<a href="#"><u>setSize(int size)</u></a> Set packet size
void	<a href="#"><u>setSrc(int src)</u></a> Set stream id (0 if not stream data)
void	<a href="#"><u>setTimecode(long timecode)</u></a> Set timecode (milliseconds) sometimes relative
void	<a href="#"><u>setType(int type)</u></a> 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, wait

## Constructors

### AMFObj

```
public AMFObj(int id)
```

Create new AMFObj for a given channel (id)

**Parameters:**

id - channel id

---

### AMFObj

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

Create new AMFObj for a given channel (id)

**Parameters:**

id - channel id

objectEncoding - object encoding level (AMF0 or AMF3)

---

## Methods

### getId

```
public int getId()
```

Get channel id

**Returns:**

channel id

---

### setId

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

Set channel id

**Parameters:**

id - channel id

---

### getSize

```
public int getSize()
```

Get packet size

**Returns:**

packet size

---

### setSize

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

Set packet size

---

(continued from last page)

**Parameters:**

size - packet size

---

**getType**

```
public int getType()
```

Get content type IVHost.CONTENTTYPE\_\*

**Returns:**

content type

---

**setType**

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

Set content type IVHost.CONTENTTYPE\_\*

**Parameters:**

type - content type

---

**getSrc**

```
public int getSrc()
```

Get stream id (0 if not stream data)

**Returns:**

stream id

---

**setSrc**

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

Set stream id (0 if not stream data)

**Parameters:**

src - stream id

---

**toString**

```
public String toString()
```

Return object as formatted string

---

**getTimecode**

```
public long getTimecode()
```

Get timecode (milliseconds) sometimes relative

**Returns:**

timecode (milliseconds)

---

**setTimecode**

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

---

(continued from last page)

Set timecode (milliseconds) sometimes relative

**Parameters:**

timecode - timecode (milliseconds)

---

## isNew

```
public boolean isNew( )
```

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

**Returns:**

is new packet

---

## setNew

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

Set is new packet

**Parameters:**

isNew - is new packet

---

## getAbsTimecode

```
public long getAbsTimecode( )
```

Get the absolute time code

**Returns:**

absolute timecode

---

## setAbsTimecodeLong

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

Set the absolute timecode

**Parameters:**

absTimecode - absolute timecode

**Returns:**

absolute timecode

---

## setAbsTimecodeShort

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

Set the absolute timecode

**Parameters:**

absTimecode

**Returns:**

absolute timecode

---

(continued from last page)

## incAbsTimecode

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

Increment the absolute timecode

**Parameters:**

absTimecode - absolute timecode

**Returns:**

absolute timecode

---

## getByteContainerLevel

```
public int getByteContainerLevel()
```

Fake container for processing

**Returns:**

current container level

---

## setByteContainerLevel

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

Fake container for processing

**Parameters:**

byteContainerLevel - current container level

---

## incByteContainerLevel

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

Fake container for processing

**Parameters:**

byteContainerLevel - current container level

---

## isByteContainerEmpty

```
public boolean isByteContainerEmpty()
```

Fake container for processing

**Returns:**

is container full

---

## isByteContainerFull

```
public boolean isByteContainerFull()
```

Fake container for processing

**Returns:**

is container full

---

(continued from last page)

## isLongTimecode

```
public boolean isLongTimecode()
```

Get is a 32 bit timecode

**Returns:**

true if 32 bit timecode

---

## setLongTimecode

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

Set is a 32 bit timecode

**Parameters:**

isLongTimecode - is a 32 bit timecode

---

## isLastSentAbsTimecode

```
public boolean isLastSentAbsTimecode()
```

---

## setLastSentAbsTimecode

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

---

## isObjectEncodingAMF3

```
public boolean isObjectEncodingAMF3()
```

---

## isObjectEncodingAMF0

```
public boolean isObjectEncodingAMF0()
```

---

## setObjectEncoding

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

---

## getObjectEncoding

```
public int getObjectEncoding()
```

---

## com.wowza.wms.amf

### Class AMPacket

java.lang.Object

└─com.wowza.wms.amf.AMPacket

public class **AMFPacket**  
extends Object

AMFPacket: data container for data being transferred to and from the server from the Flash client. AMPacket is also used to store data read/written to/from an flv file.

### Constructor Summary

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

### Method Summary

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

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

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

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



## Constructors

### AMFPacket

```
public AMFPacket()
```

Create new empty packet

---

### AMFPacket

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

Create new packet with given values

**Parameters:**

type - packet content type: IVHost.CONTENT\*TYPE\_\*

src - stream id

size - packet size (bytes)

## Methods

### clone

```
public AMFPacket clone()
```

---

### getSize

```
public int getSize()
```

Get packet size

**Returns:**

packet size

---

### setSize

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

Set packet size

**Parameters:**

size - packet size

---

### truncatePacket

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

---

### getMissing

```
public int getMissing()
```

---

(continued from last page)

Get the number of bytes remaining unfilled in the packet

**Returns:**

number of bytes unfilled in packet

---

## setDataBuffer

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

Set the data for this packet

**Parameters:**

data - byte buffer that holds packet data

---

## setDataBuffer

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

Set the data buffer to a byte array

**Parameters:**

data - data buffer byte array

---

## addData

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

Add data to the packet

**Parameters:**

data - byte buffer with data  
offset - offset in byte buffer  
size - size of data

**Returns:**

number of bytes unfilled in packet

---

## getType

```
public int getType()
```

Get content type IVHost.CONTENTTYPE\_\*

**Returns:**

content type

---

## setType

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

Set content type IVHost.CONTENTTYPE\_\*

**Parameters:**

type - content type

---

(continued from last page)

## getSrc

```
public int getSrc()
```

Get stream id (0 if not stream data)

**Returns:**

stream id

---

## setSrc

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

Set stream id (0 if not stream data)

**Parameters:**

src - stream id

---

## toString

```
public String toString()
```

Return object as formatted string

---

## getTimecode

```
public long getTimecode()
```

Get timecode (milliseconds) relative

**Returns:**

timecode (milliseconds)

---

## setTimecodes

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

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

**Parameters:**

timecode - relative timecode (milliseconds)

absTimecode - absolute timecode (milliseconds)

---

## setTimecode

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

Set timecode (milliseconds) relative

**Parameters:**

timecode - timecode (milliseconds)

---

## getAbsTimecode

```
public long getAbsTimecode()
```

Get absolute timecode (milliseconds)

(continued from last page)

**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

(continued from last page)

## getData

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

Get data as byte[]

**Returns:**

data as byte[] null if no data

---

## getFirstByte

```
public int getFirstByte()
```

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

**Returns:**

first byte of data in packet

---

## getSecondByte

```
public int getSecondByte()
```

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

**Returns:**

second byte of data in packet

---

## calcTotalPacketSize

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

Calculate the total packet size for given packet parameters

**Parameters:**

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

**Returns:**

total bytes

---

## addDataEx

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

Add data to the packet

**Parameters:**

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

(continued from last page)

`srcBytes` - - size of data to copy

**Returns:**

number of bytes unfilled in packet

---

Package

**com.wowza.wms.application**

## com.wowza.wms.application Interface IApplication

public interface **IApplication**  
extends

IApplication: public interface to Application object

### Field Summary

public static final	<a href="#"><u>DEFAULT_APPLICATION_NAME</u></a> Value: <b>_defapp_</b>
---------------------	---

### Method Summary

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



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

## Fields

### DEFAULT\_APPLICATION\_NAME

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

Constant value: **\_defapp\_**

## Methods

### shutdown

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

shutdown application

#### Parameters:

isServerShutdown - true if due to shutdown of server

(continued from last page)

## getApplicationPath

```
public String getApplicationPath()
```

Get the root path for application

**Returns:**

root path for application

---

## getConfigPath

```
public String getConfigPath()
```

Get full path to Application.xml file

**Returns:**

full path to Application.xml file

---

## getAppInstance

```
public IApplicationInstance getAppInstance(String name)
```

Get applicationInstance object by name

**Parameters:**

name - applicationInstance name

**Returns:**

appliationInstance object

---

## isAppInstanceLoaded

```
public boolean isAppInstanceLoaded(String name)
```

Return true if application instance is loaded

**Parameters:**

name - applicationInstance name

**Returns:**

true if application instance is loaded

---

## getName

```
public String getName()
```

Get the name of application

**Returns:**

name of application

---

## setName

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

Set name of application

**Parameters:**

(continued from last page)

name - name of application

---

## getVHost

```
public IVHost getVHost( )
```

Get the parent vHost object

**Returns:**

parent vHost

---

## getProperties

```
public WMSProperties getProperties( )
```

Get application properties

**Returns:**

application properties

---

## addApplicationInstanceListener

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

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

**Parameters:**

applicationInstanceListener - applicationInstance listener

---

## removeApplicationInstanceListener

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

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

**Parameters:**

applicationInstanceListener - applicationInstance listener

---

## getConnectionCounter

```
public ConnectionCounter getConnectionCounter( )
```

Get the connectionCounter for application

**Returns:**

connectionCounter for application

---

## getConnectionCounter

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

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

**Parameters:**

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

**Returns:**

(continued from last page)

connection ocunter

---

## getIoPerformanceCounter

```
public IoPerformanceCounter getIoPerformanceCounter()
```

Get the performance counter for application

**Returns:**

performance counter for application

---

## getIoPerformanceCounter

```
public IoPerformanceCounter getIoPerformanceCounter(int counterIndex)
```

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

**Parameters:**

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

**Returns:**

performance counter

---

## getDateStarted

```
public String getDateStarted()
```

Get date application started

**Returns:**

date application started

---

## getTimeRunning

```
public String getTimeRunning()
```

Get time application running

**Returns:**

time application running

---

## getTimeRunningSeconds

```
public double getTimeRunningSeconds()
```

Get time running in seconds

**Returns:**

time running in seconds

---

## getAppInstanceNames

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

Get a list of application instance names

**Returns:**

list of application instance names

---

## removeAppInstance

public void **removeAppInstance**([IApplicationInstance](#) appInstance)

Disconnect all clients connected to an application instance and remove it from the IApplication application list. The proper way to call this is:

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

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

### Parameters:

appInstance - application instance to remove

---

## shutdownAppInstance

public void **shutdownAppInstance**(String appInstanceName)

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

### Parameters:

appInstanceName - application instance name

---

## readAppConfig

public String **readAppConfig**(String sName)

Method to read xml config file..

---

(continued from last page)

## **writeAppConfig**

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

Method to write xml config file..

## 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>
---------------------	--

### Method Summary

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

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



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

<a href="#"><u>IOPerformanceCounter</u></a>	<a href="#"><u>getIOPerformanceCounter</u></a> ( ) Get the performance counter for applicationInstance
<a href="#"><u>IOPerformanceCounter</u></a>	<a href="#"><u>getIOPerformanceCounter</u></a> (int counterIndex) Get the performance counter for applicationInstance for a specific technology (see IVHost.COUNTER_*)
long	<a href="#"><u>getLastTouchTime</u></a> ( ) Get the last time the instance was touched (milliseconds)
ILiveStreamPacketizer Control	<a href="#"><u>getLiveStreamPacketizerControl</u></a> ( ) Get the Live Stream Packetizer Contoller.
String	<a href="#"><u>getLiveStreamPacketizerList</u></a> ( ) Get the comma separated list of LiveStreamPacketizers names being used by this application (see conf/LiveStreamPacketizers.xml)
<a href="#"><u>WMSProperties</u></a>	<a href="#"><u>getLiveStreamPacketizerProperties</u></a> ( ) Get the property collection of LiveStreamPacketizer settings that are specific to this application instance
int	<a href="#"><u>getMaximumPendingWriteBytes</u></a> ( ) Get maximum number a bytes a client connection can have waiting to be sent before the connection is terminated.
int	<a href="#"><u>getMaximumSetBufferTime</u></a> ( ) Get maximum number of milliseconds allowed for the NetStream.setBufferTime(secs) call.
int	<a href="#"><u>getMaxStorageDirDepth</u></a> ( ) Maximum folder depth allowed for the StreamStorageDir and SharedObjectStorageDir paths
<a href="#"><u>WMSProperties</u></a>	<a href="#"><u>getMediaCasterProperties</u></a> ( ) Get the property collection of media caster settings that are specific to this application instance
<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>IMediaCasterValidateM ediaCaster</u></a>	<a href="#"><u>getMediaCasterValidator</u></a> ( ) Get the MediaCaster validator interface for this application instance
<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 reader settings that are specific to this application instance
ModuleFunctions	<a href="#"><u>getModFunctions</u></a> ( ) Get list of application modules
Object	<a href="#"><u>getModuleInstance</u></a> (String name) Get the instance of the module class for this application instance.
ModuleList	<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</code></a> (String streamName) 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</code></a> (String name) Get a list of media streams that are listening for published stream.
<a href="#"><code>WMSProperties</code></a>	<a href="#"><code>getProperties</code></a> ( ) Get applicationInstance properties
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.
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.AVSYNCMETHODS_SENDERREPORT, RTPStream.AVSYNCMETHODS_SYSTEMCLOCK, RTPStream.AVSYNCMETHODS_RTPTIMECODE)
int	<a href="#"><code>getRTPIdleFrequency</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="#"><code>getRTPPlayAuthenticationMethod</code></a> ( ) Get the RTP play authentication method (as defined in conf/Authentication.xml)
<a href="#"><code>WMSProperties</code></a>	<a href="#"><code>getRTPProperties</code></a> ( ) Get the property collection of RTP settings that are specific to this application instance
String	<a href="#"><code>getRTPPublishAuthenticationMethod</code></a> ( ) Get the RTP publish authentication method (as defined in conf/Authentication.xml)
int	<a href="#"><code>getRTPSessionCount</code></a> ( ) Get the number of RTP sessions running under this application instance
int	<a href="#"><code>getRTPSessionCount</code></a> (String streamName) Get the number of RTP player streams playing a given stream name

java.util.Map	<a href="#"><u>getRTPSessionCountsByName()</u></a> Get a map of stream names and session counts of RTP sessions
java.util.List	<a href="#"><u>getRTPSessions()</u></a> Get a list of RTP sessions running under this application instance
java.util.List	<a href="#"><u>getRTPSessions(String streamName)</u></a> Get a list of RTP sessions running under this application instance playing a given stream name
String	<a href="#"><u>getRTSPBindIpAddress()</u></a> Set the IP address to which UDP ports will be bound for RTSP/RTP sessions
String	<a href="#"><u>getRTSPConnectionAddressType()</u></a> Get the connection IP address type (IP4) to used in the Session Description Protocol data exchanged for an RTSP/RTP session
String	<a href="#"><u>getRTSPConnectionIpAddress()</u></a> Get the connection IP address to used in the Session Description Protocol data exchanged for an RTSP/RTP session
int	<a href="#"><u>getRTSPMaximumPendingWriteBytes()</u></a> Get the maximum number of pending write bytes for an RTSP session
String	<a href="#"><u>getRTSPOriginAddressType()</u></a> Get the origin IP address type (IP4) to used in the Session Description Protocol data exchanged for an RTSP/RTP session
String	<a href="#"><u>getRTSPOriginIpAddress()</u></a> Get the origin IP address to used in the Session Description Protocol data exchanged for an RTSP/RTP session
int	<a href="#"><u>getRTSPSessionTimeout()</u></a> Get the RTSP session timeout (milliseconds)
String	<a href="#"><u>getSharedObjectReadAccess()</u></a> Get the default shared object read access
<a href="#"><u>ISharedObjects</u></a>	<a href="#"><u>getSharedObjects()</u></a> Get non-persistent shared object collection
<a href="#"><u>ISharedObjects</u></a>	<a href="#"><u>getSharedObjects(boolean isPersistent)</u></a> Get either persistent or non-persistent shared object collection
String	<a href="#"><u>getSharedObjectWriteAccess()</u></a> Get the default shared object write access
String	<a href="#"><u>getStreamAudioSampleAccess()</u></a> Get the default stream audio sample access
int	<a href="#"><u>getStreamCount()</u></a> Get the total number of open streams attached to this application instance
<a href="#"><u>IMediaStreamFileMapper</u></a>	<a href="#"><u>getStreamFileMapper()</u></a> Get the stream file mapper.
String	<a href="#"><u>getStreamKeyDir()</u></a> Get the stream key path
String	<a href="#"><u>getStreamKeyPath()</u></a> Get the resolved key path to the MediaStreams encryption keys

<a href="#"><u>IMediaStreamNameAliasProvider</u></a>	<a href="#"><u>getStreamNameAliasProvider()</u></a> Get the stream name alias provider
<a href="#"><u>WMSPProperties</u></a>	<a href="#"><u>getStreamProperties()</u></a> Get the property collection of stream settings that are specific to this application instance
String	<a href="#"><u>getStreamReadAccess()</u></a> Get the default stream read access
<a href="#"><u>MediaStreamMap</u></a>	<a href="#"><u>getStreams()</u></a> Get all the mediaStream objects attached to this applicationInstance
String	<a href="#"><u>getStreamStorageDir()</u></a> Get stream storage path
String	<a href="#"><u>getStreamStoragePath()</u></a> Get the resolved storage path to the MediaStreams
String	<a href="#"><u>getStreamType()</u></a> Get default streamType for application.
String	<a href="#"><u>getStreamVideoSampleAccess()</u></a> Get the default stream video sample access
String	<a href="#"><u>getStreamWriteAccess()</u></a> Get the default stream write access
String	<a href="#"><u>getTimeRunning()</u></a> Get time applicationInstance running
double	<a href="#"><u>getTimeRunningSeconds()</u></a> Get time running in seconds
int	<a href="#"><u>getValidationFrequency()</u></a> Get time between validation pings (milliseconds)
<a href="#"><u>IVHost</u></a>	<a href="#"><u>getVHost()</u></a> Get parent vHost
void	<a href="#"><u>incClientCountTotal()</u></a> Increment the total number of connected client counter by one
boolean	<a href="#"><u>isAcceptConnection()</u></a> Is auto accept connection on/off
boolean	<a href="#"><u>isRTPIncomingDatagramPortValid(int port)</u></a> Check a port number to be sure it is a valid RTP UDP port for this application instance
boolean	<a href="#"><u>isValidateFMLEConnections()</u></a> Returns true if validating FMLE connection (default is false)
void	<a href="#"><u>notifyLiveStreamPacketizerCreate(ILiveStreamPacketizer liveStreamPacketizer, String streamName)</u></a> Notify Live Stream Packetizer Create
void	<a href="#"><u>notifyLiveStreamPacketizerDestroy(ILiveStreamPacketizer liveStreamPacketizer)</u></a> Notify Live Stream Packetizer Destory

void	<a href="#"><u>notifyLiveStreamPacketizerInit</u></a> ( <a href="#"><u>ILiveStreamPacketizer</u></a> liveStreamPacketizer, String streamName) Notify Live Stream Packetizer Init
void	<a href="#"><u>notifyMediaReaderClose</u></a> ( <a href="#"><u>IMediaReader</u></a> mediaReader, <a href="#"><u>IMediaStream</u></a> stream) Notify media reader notifyMediaReaderClose
void	<a href="#"><u>notifyMediaReaderCreate</u></a> ( <a href="#"><u>IMediaReader</u></a> mediaReader) Notify media reader notifyMediaReaderCreate
void	<a href="#"><u>notifyMediaReaderExtractMetaData</u></a> ( <a href="#"><u>IMediaReader</u></a> mediaReader, <a href="#"><u>IMediaStream</u></a> stream) Notify media reader notifyMediaReaderExtractMetaData
void	<a href="#"><u>notifyMediaReaderInit</u></a> ( <a href="#"><u>IMediaReader</u></a> mediaReader, <a href="#"><u>IMediaStream</u></a> stream) Notify media reader notifyMediaReaderInit
void	<a href="#"><u>notifyMediaReaderOpen</u></a> ( <a href="#"><u>IMediaReader</u></a> mediaReader, <a href="#"><u>IMediaStream</u></a> stream) Notify media reader notifyMediaReaderOpen
void	<a href="#"><u>notifyMediaWriterOnFLVAddMetadata</u></a> ( <a href="#"><u>IMediaStream</u></a> stream, java.util.Map extraMetadata) Notify all MediaWriter listeners of onFLVAddMetadata
void	<a href="#"><u>notifyMediaWriterOnWriteComplete</u></a> ( <a href="#"><u>IMediaStream</u></a> stream, java.io.File file) Notify all MediaWriter listeners of onWriteComplete
void	<a href="#"><u>parseAllowDomains</u></a> (String domainFilterStr) Parse a comma delimited list of domain names used to control access to this application.
String	<a href="#"><u>readAppInstConfig</u></a> (String sName) Method to read xml config file..
void	<a href="#"><u>registerPlayRTPSession</u></a> ( <a href="#"><u>RTPSession</u></a> rtpSession) Register an RTP session as a play session
void	<a href="#"><u>removeClientListener</u></a> ( <a href="#"><u>IClientNotify</u></a> clientListener) Remove client listener.
void	<a href="#"><u>removeHTTPStreamerSession</u></a> ( <a href="#"><u>IHTTPStreamerSession</u></a> httpStreamerSession) Remove a HTTPStreamerSession from this application instance
void	<a href="#"><u>removeLiveStreamPacketizerListener</u></a> ( <a href="#"><u>ILiveStreamPacketizerActionNotify</u></a> liveStreamPacketizerListener) Remove a Live Stream Packetizer listener (see: <a href="#"><u>ILiveStreamPacketizerActionNotify</u></a> )
void	<a href="#"><u>removeMediaCasterListener</u></a> ( <a href="#"><u>IMediaCasterNotify</u></a> mediaCasterListener) Remove mediaCaster listener.
void	<a href="#"><u>removeMediaReaderListener</u></a> ( <a href="#"><u>IMediaReaderActionNotify</u></a> mediaReaderListener) Remove media reader listener.
void	<a href="#"><u>removeMediaStreamListener</u></a> ( <a href="#"><u>IMediaStreamNotify</u></a> mediaStreamListener) Remove mediaStream listener.
void	<a href="#"><u>removeMediaWriterListener</u></a> ( <a href="#"><u>IMediaWriterActionNotify</u></a> listener) remove MediaWriter listener class.
void	<a href="#"><u>removeModuleListener</u></a> ( <a href="#"><u>IModuleNotify</u></a> moduleListener) Remove module listener

void	<a href="#"><u>removePlayStreamByName</u></a> ( <a href="#"><u>IMediaStream</u></a> stream) Remove media stream from the list of streams that are listening for a published stream
void	<a href="#"><u>removePublisher</u></a> ( <a href="#"><u>Publisher</u></a> publisher) Remove a server side publisher from this application instance
void	<a href="#"><u>removeRTPSession</u></a> ( <a href="#"><u>RTPSession</u></a> rtpSession) Remove an RTP session from this application instance
void	<a href="#"><u>removeSharedObjectListener</u></a> ( <a href="#"><u>ISharedObjectNotify</u></a> sharedObjectListener, boolean isPersistent) Remove sharedObject listener.
boolean	<a href="#"><u>resetMediaCasterStream</u></a> (String streamName) Reset a media caster stream
boolean	<a href="#"><u>resetMediaCasterStream</u></a> (String streamName, String streamExt) Reset a media caster stream
void	<a href="#"><u>setAcceptConnection</u></a> (boolean acceptConnection) Set is auto accept connection
void	<a href="#"><u>setAllowDomains</u></a> (String[] domainFilter) Set the list of domain names used to control access to this application.
void	<a href="#"><u>setApplicationInstanceTouchTimeout</u></a> (int applicationInstanceTouchTimeout) Set the application instance touch timeout (milliseconds).
void	<a href="#"><u>setApplicationTimeout</u></a> (int applicationTimeout) Set application timeout (milliseconds)
void	<a href="#"><u>setClientIdleFrequency</u></a> (int clientIdleFrequency) Set default client idle frequency (milliseconds)
void	<a href="#"><u>setHTTPStreamerList</u></a> (String httpStreamerList) Set the comma separated list of HTTPStreamer names being used by this application (see conf/HTTPStreamers.xml)
void	<a href="#"><u>setLiveStreamPacketizerControl</u></a> ( <a href="#"><u>ILiveStreamPacketizerControl</u></a> liveStreamPacketizerControl) Set the Live Stream Packetizer Contoller.
void	<a href="#"><u>setLiveStreamPacketizerList</u></a> (String liveStreamPacketizerList) Set the comma separated list of LiveStreamPacketizers names being used by this application (see conf/LiveStreamPacketizers.xml)
void	<a href="#"><u>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>setMediaCasterValidator</u></a> ( <a href="#"><u>IMediaCasterValidateMediaCaster</u></a> mediaCasterValidator) Set the MediaCaster validator interface for this application instance

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



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

## Fields

### 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

---

### setName

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

Set applicationInstance name

**Parameters:**

(continued from last page)

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

---

## getStreamType

```
public String getStreamType()
```

Get default streamType for application.

**Returns:**

streamType name

---

(continued from last page)

## setStreamType

```
public void setStreamType(String streamType)
```

Set default stream type for application.

**Parameters:**

streamType - streamType name

---

## isAcceptConnection

```
public boolean isAcceptConnection()
```

Is auto accept connection on/off

**Returns:**

auto accept connection

---

## setAcceptConnection

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

Set is auto accept connection

**Parameters:**

acceptConnection - auto accept connection

---

## getClientCountTotal

```
public int getClientCountTotal()
```

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

**Returns:**

number of client connections

---

## incClientCountTotal

```
public void incClientCountTotal()
```

Increment the total number of connected client counter by one

---

## getClientCount

```
public int getClientCount()
```

Get number of client connections currently connected to applicationInstance

**Returns:**

number of client connections

---

## getClientById

```
public IClient getClientById(int index)
```

Get a client connection by the client Id

**Parameters:**

(continued from last page)

index - client Id

**Returns:**

client connection

---

**getClients**

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

Get the set of clients currently connected to this application instance (replaces getClient(index))

**Returns:**

set of clients

---

**getClient**

```
public IClient getClient(int index)
```

**Deprecated.** *Get the client connection at index. This method is deprecated. It is best to use getClient() to return a List objects.*

**Parameters:**

index - index

**Returns:**

client connection

---

**getSharedObjects**

```
public ISharedObjects getSharedObjects()
```

Get non-persistent shared object collection

**Returns:**

collection of non-persistent shared objects

---

**getSharedObjects**

```
public ISharedObjects getSharedObjects(boolean isPersistent)
```

Get either persistent or non-persistent shared object collection

**Parameters:**

isPersistent

**Returns:**

collection of shared objects

---

**addClientListener**

```
public void addClientListener(IClientNotify clientListener)
```

(continued from last page)

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

## Add a Client Listener

```
IApplicationInstance appInstance;

class ClientListener implements IClientNotify
{
    public void onClientConnect(IClient client)
    {
        WMSLoggerFactory.getLogger(null).debug("onClientConnect: "+
            client.getClientId());
    }

    public void onClientDisconnect(IClient client)
    {
        WMSLoggerFactory.getLogger(null).debug("onClientDisconnect: "+
            client.getClientId());
    }

    public void onClientAccept(IClient client)
    {
        WMSLoggerFactory.getLogger(null).debug("onClientAccept: "+
            client.getClientId());
    }

    public void onClientReject(IClient client)
    {
        WMSLoggerFactory.getLogger(null).debug("onClientReject: "+
            client.getClientId());
    }
}

appInstance.addClientListener(new ClientListener());
```

### Parameters:

clientListener - client listener

---

## removeClientListener

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

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

### Parameters:

clientListener - client listener

## addMediaStreamListener

public void **addMediaStreamListener**([IMediaStreamNotify](#) mediaStreamListener)

Add mediaStream listener. Listens for create and destroy

### Add a MediaStream Listener

```
IApplicationInstance appInstance;

class MediaStreamListener implements IMediaStreamNotify
{
    public void onMediaStreamCreate(IMediaStream stream)
    {
        WMSLoggerFactory.getLogger(null).debug("onMediaStreamCreate: "+
            stream.getSrc());
    }

    public void onMediaStreamDestroy(IMediaStream stream)
    {
        WMSLoggerFactory.getLogger(null).debug("onMediaStreamDestroy: "+
            stream.getSrc());
    }
}

appInstance.addMediaStreamListener(new MediaStreamListener());
```

#### Parameters:

mediaStreamListener - mediaStream listener

---

## removeMediaStreamListener

public void **removeMediaStreamListener**([IMediaStreamNotify](#) mediaStreamListener)

Remove mediaStream listener. Listens for create and destroy

#### Parameters:

mediaStreamListener - mediaStream listener

---

## addSharedObjectListener

public void **addSharedObjectListener**([ISharedObjectNotify](#) sharedObjectListener,  
boolean isPersistent)

(continued from last page)

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

## Add SharedObject Listener

```
IApplicationInstance appInstance;

class SharedObjectListener implements ISharedObjectNotify
{
    public void onSharedObjectCreate(ISharedObject sharedObject)
    {
        WMSLoggerFactory.getLogger(null).debug("onSharedObjectCreate: "+
            sharedObject.getName());
    }

    public void onSharedObjectDestroy(ISharedObject sharedObject)
    {
        WMSLoggerFactory.getLogger(null).debug("onSharedObjectDestroy: "+
            sharedObject.getName());
    }

    public void onSharedObjectConnect(ISharedObject sharedObject, IClient client)
    {
        WMSLoggerFactory.getLogger(null).debug("onSharedObjectConnect: "+
            sharedObject.getName());
    }

    public void onSharedObjectDisconnect(ISharedObject sharedObject, IClient client)
    {
        WMSLoggerFactory.getLogger(null).debug("onSharedObjectDisconnect: "+
            sharedObject.getName());
    }
}

appInstance.addSharedObjectListener(new SharedObjectListener(), false);
```

### Parameters:

sharedObjectListener - sharedObject listener  
isPersistent

## removeSharedObjectListener

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

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

### Parameters:



(continued from last page)

sharedObjectListener - sharedObject listener  
isPersistent

---

## addMediaCasterListener

```
public void addMediaCasterListener(IMediaCasterNotify mediaCasterListener)
```

Add mediaCaster listener. Listens for create, destroy, registerPlayer, unregisterPlayer, setSourceStream

**Parameters:**

mediaCasterListener

---

## addMediaCasterListener

```
public void addMediaCasterListener(IMediaCasterNotify2 mediaCasterListener)
```

Add mediaCaster listener. Listens for create, destroy, registerPlayer, unregisterPlayer, setSourceStream

**Parameters:**

mediaCasterListener

---

## removeMediaCasterListener

```
public void removeMediaCasterListener(IMediaCasterNotify mediaCasterListener)
```

Remove mediaCaster listener. Listens for create, destroy, registerPlayer, unregisterPlayer, setSourceStream

**Parameters:**

mediaCasterListener

---

## getConnectionCounter

```
public ConnectionCounter getConnectionCounter( )
```

Get the connectionCounter for applicationInstance

**Returns:**

connection counter

---

## getConnectionCounter

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

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

**Parameters:**

counterIndex - counter index

**Returns:**

connection counter

---

## getDateStarted

```
public String getDateStarted( )
```

Get date applicationInstance started

**Returns:**

(continued from last page)

date applicationInstance started

---

## getTimeRunning

```
public String getTimeRunning()
```

Get time applicationInstance running

**Returns:**

time applicationInstance running

---

## getTimeRunningSeconds

```
public double getTimeRunningSeconds()
```

Get time running in seconds

**Returns:**

time running in seconds

---

## broadcastMsg

```
public void broadcastMsg(java.util.List clientList,  
                          String handlerName)
```

Broadcast a message to a specific list of clients connected to this application instance

**Parameters:**

clientList - list of client

handlerName - handler name

---

## broadcastMsg

```
public void broadcastMsg(java.util.List clientList,  
                          String handlerName,  
                          Object[] params)
```

Broadcast a message to a specific list of clients connected to this application instance

**Parameters:**

clientList - list of client

handlerName - handler name

params - parameters

---

## broadcastMsg

```
public void broadcastMsg(String handlerName,  
                          Object[] params)
```

(continued from last page)

Broadcast a message to all clients connected to this applicationInstance

## Broadcast Message to All Clients

```
IApplicationInstance appInstance;
appInstance.broadcastMsg("onNotify", "Hello World", 1.2345, false, new Date());
```

### Parameters:

handlerName - handler name

params - variable list of arguments (Java primitive and Strings will be wrapped in AMFData objects)

---

## getIOPerformanceCounter

```
public IOPerformanceCounter getIOPerformanceCounter()
```

Get the performance counter for applicationInstance

### Returns:

io performance counter

---

## getIOPerformanceCounter

```
public IOPerformanceCounter getIOPerformanceCounter(int counterIndex)
```

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

### Parameters:

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

### Returns:

connection counter

---

## addPlayStreamByName

```
public void addPlayStreamByName(IMediaStream stream,
    String name)
```

Add a media stream to the list of streams that are listening for a published stream

### Parameters:

stream - media stream

name - stream name

---

## removePlayStreamByName

```
public void removePlayStreamByName(IMediaStream stream)
```

Remove media stream from the list of streams that are listening for a published stream

### Parameters:

(continued from last page)

stream - media stream

---

## getPlayStreamCountsByName

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

Get a map of stream names to number of Flash players playing the stream name

**Returns:**

map of stream names to number of Flash players playing the stream name

---

## getPlayStreamCount

```
public int getPlayStreamCount(String streamName)
```

Get the number of Flash players playing a given stream name

**Parameters:**

streamName - stream name

**Returns:**

number of players

---

## getPlayStreamsByName

```
public java.util.List getPlayStreamsByName(String name)
```

Get a list of media streams that are listening for published stream.

**Parameters:**

name - stream name

**Returns:**

list of streams or null if no listeners

---

## getMediaCasterStreams

```
public MediaCasterStreamMap getMediaCasterStreams( )
```

Get the media caster streams attached to this application instance

**Returns:**

media caster streams attached to this application instance

---

## getStreamCount

```
public int getStreamCount( )
```

Get the total number of open streams attached to this application instance

**Returns:**

the total number of open streams attached to this application instance

---

## getModFunctions

```
public ModuleFunctions getModFunctions( )
```

Get list of application modules

---

(continued from last page)

**Returns:**

list of application modules

---

**addModuleListener**

```
public void addModuleListener(IModuleNotify moduleListener)
```

Add module listener. Listens for onModuleLoad and onModuleUnload events. See IModuleNotify.

**Parameters:**

moduleListener - module listener

---

**removeModuleListener**

```
public void removeModuleListener(IModuleNotify moduleListener)
```

Remove module listener

**Parameters:**

moduleListener - module listener

---

**getModuleList**

```
public ModuleList getModuleList()
```

Get the list of loaded modules.

**Returns:**

list of loaded modules

---

**getModuleInstance**

```
public Object getModuleInstance(String name)
```

Get the instance of the module class for this application instance.

**Parameters:**

name - module name as defined in Application.xml

**Returns:**

instance of class for this application instance

---

**getApplicationTimeout**

```
public int getApplicationTimeout()
```

Get application timeout (milliseconds)

**Returns:**

application timeout (milliseconds)

---

**setApplicationTimeout**

```
public void setApplicationTimeout(int applicationTimeout)
```

Set application timeout (milliseconds)

(continued from last page)

**Parameters:**

applicationTimeout - application timeout (milliseconds)

---

**getPingTimeout**

```
public int getPingTimeout()
```

Get ping timeout (milliseconds)

**Returns:**

ping timeout (milliseconds)

---

**setPingTimeout**

```
public void setPingTimeout(int pingTimeout)
```

Set ping timeout (milliseconds)

**Parameters:**

pingTimeout - ping timeout (milliseconds)

---

**getValidationFrequency**

```
public int getValidationFrequency()
```

Get time between validation pings (milliseconds)

**Returns:**

time between validation pings (milliseconds)

---

**setValidationFrequency**

```
public void setValidationFrequency(int validationFrequency)
```

Set time between validation pings (milliseconds)

**Parameters:**

validationFrequency - time between validation pings (milliseconds)

---

**getMaximumPendingWriteBytes**

```
public int getMaximumPendingWriteBytes()
```

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

**Returns:**

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

---

**setMaximumPendingWriteBytes**

```
public void setMaximumPendingWriteBytes(int maximumPendingWriteBytes)
```

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

**Parameters:**

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

---

## setMaximumSetBufferTime

```
public void setMaximumSetBufferTime(int maximumSetBufferTime)
```

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

**Parameters:**

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

---

## getMaximumSetBufferTime

```
public int getMaximumSetBufferTime()
```

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

**Returns:**

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

---

## getRepeaterOriginUrl

```
public String getRepeaterOriginUrl()
```

Get the Repeater Origin URL used by the Live Stream Repeater

**Returns:**

URL used by the Live Stream Repeater

---

## setRepeaterOriginUrl

```
public void setRepeaterOriginUrl(String repeaterOriginUrl)
```

Set the Repeater Origin URL used by the Live Stream Repeater

**Parameters:**

repeaterOriginUrl - URL used by the Live Stream Repeater

---

## getRepeaterQueryString

```
public String getRepeaterQueryString()
```

Get the Repeater query string that is used to connect to the origin. This value can be used to pass secure URL parameters to the origin for security validation.

**Returns:**

Repeater query string

---

## setRepeaterQueryString

```
public void setRepeaterQueryString(String repeaterQueryString)
```

Set the Repeater query string that is used to connect to the origin. This value can be used to pass secure URL parameters to the origin for security validation.

**Parameters:**

repeaterQueryString - Repeater query string

---

## getAllowDomains

```
public String[] getAllowDomains()
```

Get the list of domain names used to control access to this application. Upon connection, if this list is non-null the client.referrer value is checked to make sure the referrer is from a domain in this list.

**Returns:**

list of domain names used to control access to this application

---

## setAllowDomains

```
public void setAllowDomains(String[] domainFilter)
```

Set the list of domain names used to control access to this application. Upon connection, if this list is non-null the client.referrer value is checked to make sure the referrer is from a domain in this list.

**Parameters:**

domainFilter - list of domain names used to control access to this application

---

## parseAllowDomains

```
public void parseAllowDomains(String domainFilterStr)
```

Parse a comma delimited list of domain names used to control access to this application. Upon connection, if this list is non-null the client.referrer value is checked to make sure the referrer is from a domain in this list.

**Parameters:**

domainFilterStr - comma delimited list of domain names

---

## getClientIdleFrequency

```
public int getClientIdleFrequency()
```

Get default client idle frequency (milliseconds)

**Returns:**

default client idle frequency (milliseconds)

---

## setClientIdleFrequency

```
public void setClientIdleFrequency(int clientIdleFrequency)
```

Set default client idle frequency (milliseconds)

**Parameters:**

clientIdleFrequency - default client idle frequency (milliseconds)

---

## getRTPIdeFrequency

```
public int getRTPIdeFrequency()
```

Set the default RTP idle frequency (milliseconds)

**Returns:**

default RTP idle frequency (milliseconds)

---



(continued from last page)

## 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)
```

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)
```

(continued from last page)

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

**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\(String\)](#)

(continued from last page)

## 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

**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()
```

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

---

(continued from last page)

**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

---

## 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

---

(continued from last page)

**Returns:**

the number of RTP sessions

---

**getRTPSessionCount**

```
public int getRTPSessionCount()
```

Get the number of RTP sessions running under this application instance

**Returns:**

the number of RTP sessions running under this application instance

---

**addRTPSession**

```
public void addRTPSession(RTPSession rtpSession)
```

Add an RTP session to this application instance

**Parameters:**

rtpSession - RTP session to add

---

**registerPlayRTPSession**

```
public void registerPlayRTPSession(RTPSession rtpSession)
```

Register an RTP session as a play session

**Parameters:**

rtpSession - RTP session to register

---

**removeRTPSession**

```
public void removeRTPSession(RTPSession rtpSession)
```

Remove an RTP session from this application instance

**Parameters:**

rtpSession - RTP session to remove

---

**getClientsLockObj**

```
public edu.emory.mathcs.backport.java.util.concurrent.locks.WMSReadWriteLock  
getClientsLockObj()
```

Get the read/write lock for this application instance

**Returns:**

read/write lock for this application instance

---

**getStreamProperties**

```
public WMSProperties getStreamProperties()
```

Get the property collection of stream settings that are specific to this application instance

**Returns:**

property collection of stream settings

## getMediaCasterProperties

```
public WMSProperties getMediaCasterProperties( )
```

Get the property collection of media caster settings that are specific to this application instance

**Returns:**

property collection of media caster settings

---

## getMediaReaderProperties

```
public WMSProperties getMediaReaderProperties( )
```

Get the property collection of media reader settings that are specific to this application instance

**Returns:**

property collection of media reader settings

---

## getMediaWriterProperties

```
public WMSProperties getMediaWriterProperties( )
```

Get the property collection of media reader settings that are specific to this application instance

**Returns:**

property collection of media reader settings

---

## getRTPProperties

```
public WMSProperties getRTPProperties( )
```

Get the property collection of RTP settings that are specific to this application instance

**Returns:**

property collection of RTP settings

---

## getLiveStreamPacketizerProperties

```
public WMSProperties getLiveStreamPacketizerProperties( )
```

Get the property collection of LiveStreamPacketizer settings that are specific to this application instance

**Returns:**

property collection of LiveStreamPacketizer 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( )
```



(continued from last page)

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)
```

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:**

(continued from last page)

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:

httpStreamer - HTTP Streamer name

### Returns:

true is this type of streaming is allowed

---

## containsLiveStreamPacketizer

```
public boolean containsLiveStreamPacketizer(String liveStreamPacketizer)
```

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

### Parameters:

liveStreamPacketizer - live stream packetizer name

### Returns:

true if contains reference to it

---

## getStreamNameAliasProvider

```
public IMediaStreamNameAliasProvider getStreamNameAliasProvider()
```

Get the stream name alias provider

### Returns:

stream name alias provider

(continued from last page)

---

## setStreamNameAliasProvider

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

Set the stream name alias provider

**Parameters:**

streamNameAliasProvider - stream name alias provider

---

## getPublishers

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

Get the set of server side publishers

**Returns:**

set of server side publishers

---

## getPublisherCount

```
public int getPublisherCount()
```

Get the current number of server side publishers

**Returns:**

number of server side publishers

---

## addPublisher

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

Add a server side publisher to this application instance

**Parameters:**

publisher - server side publisher

---

## removePublisher

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

Remove a server side publisher from this application instance

**Parameters:**

publisher - server side publisher

---

## getHTTPStreamerSessions

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

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

**Parameters:**

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

**Returns:**

(continued from last page)

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

---

(continued from last page)

**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 (HTTPStreamerSession.SESSIONPROTOCOL\_\*) for protocols

**Parameters:**

protocol - streaming protocol (HTTPStreamerSession.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 (HTTPStreamerSession.SESSIONPROTOCOL\_\*) for protocols

**Parameters:**

protocol - streaming protocol (HTTPStreamerSession.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

---

(continued from last page)

---

## 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

**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..

---

(continued from last page)

## getLiveStreamPacketizerControl

```
public ILiveStreamPacketizerControl getLiveStreamPacketizerControl()
```

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

**Returns:**

Live Stream Packetizer Contoller

---

## setLiveStreamPacketizerControl

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

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

**Parameters:**

liveStreamPacketizerControl - Live Stream Packetizer Contoller

---

## resetMediaCasterStream

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

Reset a media caster stream

**Parameters:**

streamName - stream name

**Returns:**

true if successful

---

## resetMediaCasterStream

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

Reset a media caster stream

**Parameters:**

streamName - stream name

streamExt - stream extension

**Returns:**

true if successful

---

## startMediaCasterStream

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

Start a media caster stream

**Parameters:**

streamName - stream name

streamExt - stream extension

mediaCasterType - media caster stream type

(continued from last page)

**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.

**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



(continued from last page)

**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

**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()
```

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)
```

(continued from last page)

Add media reader listener. see [IMediaReaderActionNotify](#)

**Parameters:**

`mediaReaderListener` - media reader listener

---

**removeMediaReaderListener**

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

Remove media reader listener. see [IMediaReaderActionNotify](#)

**Parameters:**

`mediaReaderListener` - media reader listener

---

**notifyMediaReaderCreate**

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

Notify media reader `notifyMediaReaderCreate`

**Parameters:**

`mediaReader` - media reader

---

**notifyMediaReaderInit**

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

Notify media reader `notifyMediaReaderInit`

**Parameters:**

`mediaReader` - media reader

`stream` - media stream

---

**notifyMediaReaderOpen**

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

Notify media reader `notifyMediaReaderOpen`

**Parameters:**

`mediaReader` - media reader

`stream` - media stream

---

**notifyMediaReaderExtractMetaData**

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

Notify media reader `notifyMediaReaderExtractMetaData`

**Parameters:**

`mediaReader` - media reader

`stream` - media stream

---

(continued from last page)

## 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

### 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:**

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

---

(continued from last page)

---

## removeLiveStreamPacketizerListener

```
public void removeLiveStreamPacketizerListener( ILiveStreamPacketizerActionNotify  
liveStreamPacketizerListener)
```

Remove a Live Stream Packetizer listener (see: ILiveStreamPacketizerActionNotify)

### Parameters:

liveStreamPacketizerListener - Live Stream Packetizer listener

---

## notifyLiveStreamPacketizerCreate

```
public void notifyLiveStreamPacketizerCreate( ILiveStreamPacketizer  
liveStreamPacketizer,  
String streamName)
```

Notify Live Stream Packetizer Create

### Parameters:

liveStreamPacketizer - Live Stream Packetizer listener

---

## notifyLiveStreamPacketizerDestroy

```
public void notifyLiveStreamPacketizerDestroy( ILiveStreamPacketizer  
liveStreamPacketizer)
```

Notify Live Stream Packetizer Destory

### Parameters:

liveStreamPacketizer - Live Stream Packetizer listener

---

## notifyLiveStreamPacketizerInit

```
public void notifyLiveStreamPacketizerInit( ILiveStreamPacketizer liveStreamPacketizer,  
String streamName)
```

Notify Live Stream Packetizer Init

### Parameters:

liveStreamPacketizer - Live Stream Packetizer listener

---

## isValidateFMLEConnections

```
public boolean isValidateFMLEConnections()
```

Returns true if validating FMLE connection (default is false)

### 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

---

## com.wowza.wms.application Interface IApplicationInstanceNotify

public interface **IApplicationInstanceNotify**  
extends

IApplicationInstanceNotify: listener interface used by IApplication addApplicationInstanceListener

### Method Summary

void	<a href="#">onApplicationInstanceCreate</a> ( <a href="#">IApplicationInstance</a> applicationInstance) Triggered when applicationInstance created
void	<a href="#">onApplicationInstanceDestroy</a> ( <a href="#">IApplicationInstance</a> applicationInstance) Triggered when applicationInstance destroyed

### Methods

#### onApplicationInstanceCreate

public void **onApplicationInstanceCreate**([IApplicationInstance](#) applicationInstance)

Triggered when applicationInstance created

**Parameters:**

applicationInstance - applicationInstance

#### onApplicationInstanceDestroy

public void **onApplicationInstanceDestroy**([IApplicationInstance](#) applicationInstance)

Triggered when applicationInstance destroyed

**Parameters:**

applicationInstance - applicationInstance

## com.wowza.wms.application Interface IApplicationNotify

---

public interface **IApplicationNotify**  
extends

IApplicationNotify: listener interface used by IVHost addApplicationListener

---

### Method Summary

void	<a href="#">onApplicationCreate</a> ( <a href="#">IApplication</a> application) Triggered when application created
void	<a href="#">onApplicationDestroy</a> ( <a href="#">IApplication</a> application) Triggered when application destroyed

---

### Methods

#### onApplicationCreate

public void **onApplicationCreate**([IApplication](#) application)

Triggered when application created

**Parameters:**

application - application

---

#### onApplicationDestroy

public void **onApplicationDestroy**([IApplication](#) application)

Triggered when application destroyed

**Parameters:**

application - application

---



## com.wowza.wms.application Class WMSProperties

```

java.lang.Object
  |-- java.util.AbstractMap
        |-- java.util.HashMap
              |-- com.wowza.wms.application.WMSProperties

```

### All Implemented Interfaces:

java.util.Map, java.io.Serializable, Cloneable, java.util.Map

```

public class WMSProperties
extends java.util.HashMap

```

WMSProperties: generic properties container used by many class to store extended property information. Acts like a simple Map with some simple utilities for performing type conversion.

## Constructor Summary

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

## Method Summary

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

String	<a href="#">toString()</a>
--------	----------------------------

**Methods inherited from class java.util.HashMap**

clear, clone, containsKey, containsValue, entrySet, get, isEmpty, keySet, put, putAll, remove, size, values

**Methods inherited from class java.util.AbstractMap**

clear, clone, containsKey, containsValue, entrySet, equals, get, hashCode, isEmpty, keySet, put, putAll, remove, size, toString, values

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

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

**Methods inherited from interface java.util.Map**

clear, containsKey, containsValue, entrySet, equals, get, hashCode, isEmpty, keySet, put, putAll, remove, size, values

**Methods inherited from interface java.util.Map**

clear, containsKey, containsValue, entrySet, equals, get, hashCode, isEmpty, keySet, put, putAll, remove, size, values

## Constructors

### WMSProperties

```
public WMSProperties()
```

## Methods

### 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)
```

Set property to generic object.

(continued from last page)

**Parameters:**

name - property name  
value - value

---

**getProperty**

```
public Object getProperty(String name)
```

Get property value as generic object.

**Parameters:**

name - property name

**Returns:**

value, null if does not exist

---

**getPropertyStr**

```
public String getPropertyStr(String name)
```

Get property as String

**Parameters:**

name - property name

**Returns:**

value as String, null if does not exist

---

**getPropertyStr**

```
public String getPropertyStr(String name,  
                               String defaultVal)
```

Get property as String, return default value if does not exist.

**Parameters:**

name - property name  
defaultVal - default value

**Returns:**

value as String, defaultVal if does not exist

---

**getPropertyBoolean**

```
public boolean getPropertyBoolean(String name,  
                                    boolean defaultVal)
```

Get property as boolean, return default value if does not exist.

**Parameters:**

name - property name  
defaultVal - default value

**Returns:**

value as boolean, defaultVal if does not exist

---

(continued from last page)

## 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()
```

---

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](#)

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

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

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

#### Methods inherited from interface

[com.wowza.wms.authentication.IAuthenticateUsernamePasswordProvider](#)

```
getClient, getPassword, getRTPSession, getVHost, setClient, setRTPSession, setVHost,  
userExists
```

## Fields

### vhost

```
protected com.wowza.wms.vhost.IVHost vhost
```

### client

```
protected com.wowza.wms.client.IClient client
```

### rtpSession

```
protected com.wowza.wms.rtp.model.RTPSession rtpSession
```

## Constructors

### AuthenticateUsernamePasswordProviderBase

```
public AuthenticateUsernamePasswordProviderBase()
```

## Methods

### getVHost

```
public IVHost getVHost()
```

### setVHost

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

### getRTPSession

```
public RTPSession getRTPSession()
```

### setRTPSession

```
public void setRTPSession(RTPSession rtpSession)
```

(continued from last page)

---

## **getClient**

```
public IClient getClient()
```

---

## **setClient**

```
public void setClient(IClient client)
```



## com.wowza.wms.authentication Interface IAuthenticate

public interface **IAuthenticate**  
extends

### Field Summary

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

### Method Summary

void	<a href="#">init</a> ( <a href="#">IApplicationInstance</a> appInstance, AuthenticationItem authenticationItem)
void	<a href="#">init</a> ( <a href="#">IVHost</a> vhost, AuthenticationItem authenticationItem)

### 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)

(continued from last page)

**init**

```
public void init(IVHost vhost,  
                AuthenticationItem authenticationItem)
```

---

## com.wowza.wms.authentication Interface IAuthenticateHTTPProvider

---

public interface **IAuthenticateHTTPProvider**  
extends

---

### Method Summary

boolean	<a href="#">authenticateHTTPProvider</a> ( <a href="#">IVHost</a> vhost, <a href="#">IHTTPRequest</a> req, <a href="#">IHTTPResponse</a> resp)
---------	--

---

### Methods

#### **authenticateHTTPProvider**

```
public boolean authenticateHTTPProvider(IVHost vhost,  
    IHTTPRequest req,  
    IHTTPResponse resp)
```

---

## com.wowza.wms.authentication Interface IAuthenticateRTSP

---

public interface **IAuthenticateRTSP**  
extends

---

### Method Summary

boolean	<a href="#">authenticateRTSP</a> ( <a href="#">RTPSession</a> rtspSession, com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp)
---------	---

---

### Methods

#### **authenticateRTSP**

```
public boolean authenticateRTSP(RTPSession rtspSession,  
    com.wowza.wms.rtsp.RTSPRequestMessage req,  
    com.wowza.wms.rtsp.RTSPResponseMessages resp)
```

**com.wowza.wms.authentication****Interface IAuthenticateUsernamePasswordProvider**

All Known Implementing Classes:

[AuthenticateUsernamePasswordProviderBase](#)public interface **IAuthenticateUsernamePasswordProvider**

extends

**Method Summary**

<a href="#">IClient</a>	<a href="#">getClient()</a>
String	<a href="#">getPassword</a> (String username)
<a href="#">RTPSession</a>	<a href="#">getRTPSession()</a>
<a href="#">IVHost</a>	<a href="#">getVHost()</a>
void	<a href="#">setClient</a> ( <a href="#">IClient</a> client)
void	<a href="#">setRTPSession</a> ( <a href="#">RTPSession</a> rtpSession)
void	<a href="#">setVHost</a> ( <a href="#">IVHost</a> vhost)
boolean	<a href="#">userExists</a> (String username)

**Methods****userExists**public boolean **userExists**(String username)**getPassword**public String **getPassword**(String username)**getVHost**public [IVHost](#) **getVHost**()

(continued from last page)

**setVHost**

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

---

**getClient**

```
public IClient getClient()
```

---

**setClient**

```
public void setClient(IClient client)
```

---

**getRTPSession**

```
public RTPSession getRTPSession()
```

---

**setRTPSession**

```
public void setRTPSession(RTPSession rtpSession)
```

---

---

Package

**com.wowza.wms.client**

## com.wowza.wms.client Class ConnectionCounter

java.lang.Object

```

  |
+-com.wowza.wms.client.ConnectionCounterBase
  |
+-com.wowza.wms.client.ConnectionCounter

```

```

public class ConnectionCounter
extends ConnectionCounterBase

```

### Fields inherited from class com.wowza.wms.client.ConnectionCounterBase

DATEFORMATSTR, NEVERSTR, REJECTREASON\_APPLICATION, REJECTREASON\_COUNT, REJECTREASON\_LICENSE

### Constructor Summary

public	<a href="#">ConnectionCounter()</a> Create empty ConnectionCounter
--------	---

### Method Summary

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



String	<a href="#"><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="#">validateNewConnection</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 occurred  
stamp - time stamp connection occurred (milliseconds)  
license - security data

---

## rejectConnection

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

Reject connection.

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

**Parameters:**

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

---

(continued from last page)

## incrementReject

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

Increment reject connection.

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

**Parameters:**

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

---

## disconnect

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

Disconnect connection.

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

**Parameters:**

connectionHolder - connection holder  
license - security data

---

## decrement

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

Decrement connection counters.

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

**Parameters:**

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

---

## getCurrent

```
public long getCurrent()
```

Get total number of client currently connected to this object.

**Returns:**

total number of client currently connected to this object

## getLastConnectAcceptedStamp

```
public long getLastConnectAcceptedStamp()
```

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

**Returns:**

time (milliseconds) of the last conenction to this object

---

## getLastConnectAcceptedStampString

```
public String getLastConnectAcceptedStampString()
```

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

**Returns:**

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

---

## getLastConnectAcceptedTime

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

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

**Returns:**

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

---

## getLastConnectAcceptedTimeString

```
public String getLastConnectAcceptedTimeString()
```

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

**Returns:**

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

---

## getLastConnectRejectedStamp

```
public long getLastConnectRejectedStamp()
```

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

**Returns:**

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

---

## getLastConnectRejectedStampString

```
public String getLastConnectRejectedStampString()
```

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

**Returns:**

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

---

## getLastConnectRejectedTime

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

---

(continued from last page)

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

**Returns:**

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

---

## getLastConnectRejectedTimeString

```
public String getLastConnectRejectedTimeString()
```

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

**Returns:**

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

---

## getLastConnectRejectedByReasonStampString

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

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

**Parameters:**

reason - reason, see REJECTREASON\_\*

**Returns:**

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

---

## getLastConnectRejectedByReasonStamp

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

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

**Parameters:**

reason - reason, see REJECTREASON\_\*

**Returns:**

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

---

## getLastConnectRejectedByReasonTime

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

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

**Parameters:**

reason - reason, see REJECTREASON\_\*

**Returns:**

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

---

## getLastConnectRejectedByReasonTimeString

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

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

**Parameters:**

reason - reason, see REJECTREASON\_\*

(continued from last page)

**Returns:**

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

---

**getLastDisconnectStampString**

```
public String getLastDisconnectStampString()
```

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

**Returns:**

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

---

**getLastDisconnectStamp**

```
public long getLastDisconnectStamp()
```

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

**Returns:**

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

---

**getLastDisconnectTime**

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

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

**Returns:**

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

---

**getLastDisconnectTimeString**

```
public String getLastDisconnectTimeString()
```

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

**Returns:**

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

---

**getTotal**

```
public long getTotal()
```

Get total number of connection attempts to this object.

**Returns:**

total number of connection attempts to this object

---

**getTotalAccepted**

```
public long getTotalAccepted()
```

Get total number of accepted connections to this object.

**Returns:**

total number of accepted connections to this object

## getTotalRejected

```
public long getTotalRejected()
```

Get total number of rejected connections to this object.

**Returns:**

total number of rejected connections to this object



## com.wowza.wms.client Interface IClient

public interface **IClient**  
extends

IClient: public interface to Client object.

### Field Summary

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

### Method Summary

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

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

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

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

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

void	<a href="#"><u>redirectConnection</u></a> (String url, String description, String errorStr) Redirection connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected and info.ex.code of 302)
void	<a href="#"><u>rejectConnection</u></a> ( ) Reject connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected)
void	<a href="#"><u>rejectConnection</u></a> ( <a href="#"><u>AMFData</u></a> errorObj) Reject connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected)
void	<a href="#"><u>rejectConnection</u></a> (String errorStr) Reject connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected)
void	<a href="#"><u>rejectConnection</u></a> (String description, <a href="#"><u>AMFData</u></a> errorObj) Reject connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected)
void	<a href="#"><u>rejectConnection</u></a> (String description, String errorStr) Reject connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected)
void	<a href="#"><u>reparentClient</u></a> ( <a href="#"><u>IVHost</u></a> vhost) Move a client object to a new vhost.
void	<a href="#"><u>setAcceptConnection</u></a> (boolean acceptConnection) Set the default for accept connection
void	<a href="#"><u>setAcceptConnectionDescription</u></a> (String description) Call this method from onConnect to set the info.description property returned in NetConnection onStatus handler
void	<a href="#"><u>setAcceptConnectionExObj</u></a> ( <a href="#"><u>AMFDataObj</u></a> acceptConnectionExObj) Call this method from onConnect to set the info.ex property returned in NetConnection onStatus handler
void	<a href="#"><u>setAcceptConnectionObj</u></a> ( <a href="#"><u>AMFData</u></a> acceptConnectionObj) Call this method from onConnect to add an "application" property to the info object that is passed back to the Flash client in the onStatus handler.
void	<a href="#"><u>setBufferTime</u></a> (int bufferTime) Set default buffer time for newly created mediaStream objects
void	<a href="#"><u>setFastPlaySettings</u></a> ( <a href="#"><u>FastPlaySettings</u></a> fastPlaySettings) Set fastPlay settings
void	<a href="#"><u>setFlashVer</u></a> (String flashVer) Set client flash version
void	<a href="#"><u>setIdleFrequency</u></a> (int idleFrequency) Set client idle frequency (milliseconds)
void	<a href="#"><u>setLastValidateTime</u></a> (long lastValidateTime) Set the last time (milliseconds) the connection was validated with ping
void	<a href="#"><u>setLiveRepeaterCapabilities</u></a> (int liveRepeaterCapabilities) Set the live repeater capabilities of this connection

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

## Fields

### VIDEOSAMPLE\_ACCESS\_ALL

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

Constant value: \*

(continued from last page)

---

## VIDEOSAMPLE\_ACCESS\_NONE

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

Constant value:

---

## AUDIOSAMPLE\_ACCESS\_ALL

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

Constant value: \*

---

## AUDIOSAMPLE\_ACCESS\_NONE

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

Constant value:

---

## READ\_ACCESS\_ALL

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

Constant value: \*

---

## READ\_ACCESS\_NONE

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

Constant value:

---

## WRITE\_ACCESS\_ALL

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

Constant value: \*

---

## WRITE\_ACCESS\_NONE

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

Constant value:

## Methods

### getClientId

```
public int getClientId()
```

Get client id. Assigned by server and connection time.

**Returns:**

client id

---



## getFlashVer

```
public String getFlashVer()
```

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

**Returns:**

client flash version string

---

## setFlashVer

```
public void setFlashVer(String flashVer)
```

Set client flash version

**Parameters:**

flashVer - client flash version string

---

## getTotalIOPerformanceCounter

```
public IOPerformanceCounter getTotalIOPerformanceCounter()
```

Get performance counter for all bytes transferred to this client. This includes function calls.

**Returns:**

performance counter

---

## getMediaIOPerformanceCounter

```
public IOPerformanceCounter getMediaIOPerformanceCounter()
```

Get performance counter for media bytes transferred to this client. Only includes bytes to mediaStream objects

**Returns:**

performance counter

---

## getAppInstance

```
public IApplicationInstance getAppInstance()
```

Get parent applicationInstance. Is null if connection reject or before accepted.

**Returns:**

parent applicationInstance

---

## getApplication

```
public IApplication getApplication()
```

Get parent application. Is null if connection reject or before accepted.

**Returns:**

application

---

## getPlayStreams

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

---

(continued from last page)

Get a collection of all play streams. Play streams are streams that are created due to a call to play.

## Iterate Play Streams

```
IClient client;

List playStreams = client.getPlayStreams();
Iterator iter = playStreams.iterator();
while(iter.hasNext())
{
    IMediaStream stream = (IMediaStream)iter.next();
    WMSLoggerFactory.getLogger(null).debug("stream: "+stream.getName());
}
```

**Returns:**

collection of play streams

---

## getPublishStreams

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

Get a collection of publish streams. Publish streams are streams that are created due to a call to publish.

## Iterate Public Streams

```
IClient client;

List publishStreams = client.getPublishStreams();
Iterator iter = publishStreams.iterator();
while(iter.hasNext())
{
    IMediaStream stream = (IMediaStream)iter.next();
    WMSLoggerFactory.getLogger(null).debug("stream: "+stream.getName());
}
```

**Returns:**

collection of publish stream

---

---

## shutdownClient

```
public void shutdownClient()
```

Gracefully shutdown a client. Only use this method to shutdown a client if you know the client is connected to Wowza Pro. If the client may be disconnected from Wowza Pro, use `IClient.setShutdownClient(true)`

---

## touch

```
public void touch()
```

Update the last touch time for client

---

## isConnected

```
public boolean isConnected()
```

Is this client connected

**Returns:**

is connected

---

## getProperties

```
public WMSProperties getProperties()
```

Get client properties

**Returns:**

collection of client properties

---

## getStreamType

```
public String getStreamType()
```

Get default streamType

**Returns:**

streamType

---

## setStreamType

```
public void setStreamType(String streamType)
```

Set default streamType

**Parameters:**

streamType - streamType

---

## isAcceptConnection

```
public boolean isAcceptConnection()
```

Is auto accept connection

**Returns:**

auto accept connection

---

---

## acceptConnection

```
public void acceptConnection()
```

Accept connection

---

## acceptConnection

```
public void acceptConnection(String successStr)
```

Accept connection

**Parameters:**

successStr - application property value added to NetConnection.Connect.Success event object

---

## acceptConnection

```
public void acceptConnection(AMFData successObj)
```

Accept connection

**Parameters:**

successObj - application property value added to NetConnection.Connect.Success event object

---

## rejectConnection

```
public void rejectConnection()
```

Reject connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected)

---

## 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)

---

(continued from last page)

errorStr - error (returned in onStatus handler in info.application parameter)

---

## rejectConnection

```
public void rejectConnection(String description,  
    AMFData errorObj)
```

Reject connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected)

### Parameters:

description - description (returned in onStatus handler in info.description parameter)  
errorObj - error AMFData (returned in onStatus handler in info.application parameter)

---

## redirectConnection

```
public void redirectConnection(String url)
```

Redirection connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected and info.ex.code of 302)

### Parameters:

url - url (returned in onStatus handler in info.ex.redirect parameter)

---

## redirectConnection

```
public void redirectConnection(String url,  
    String description)
```

Redirection connection (NetConnection.onStatus handler returns info.code of NetConnection.Connect.Rejected and info.ex.code of 302)

### Parameters:

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 Media Server terms is the full url to the Flash movie that connected to the server. Checking referre upon connection is a great way to secure server connections and only allow your Flash movies to connect to the server.

## Only Allow Your .swf Files to Connect to Wowza Pro server

```
public class MyModule extends ModuleBase
{
    static public void onConnect(IClient client, RequestFunction function,
        AMFDataList params)
    {
        String referrer = client.getReferrer();

        if (referrer.indexOf("www.mycompany.com") >= 0 &&
            referrer.indexOf("mycoolflash.swf") >= 0)
            client.acceptConnection("valid referrer");
        else
            client.rejectConnection("invalid referrer: "+referrer);
    }
}
```

**Returns:**referrer

---

## getPageUrl

```
public String getPageUrl()
```

Get the pageUrl for this connection.

**Returns:**pageUrl for this connection

---

## getUri

```
public String getUri()
```

Get the full URI of the connection string

(continued from last page)

**Returns:**

URI of connection string

---

**getProtocol**

```
public int getProtocol()
```

Get connection protocol (1 = RTMP, 3 = RTMPT)

**Returns:**

connection protocol (1 = RTMP, 3 = RTMPT)

---

**getServerHostPort**

```
public HostPort getServerHostPort()
```

Get the hostPort object for the connection that is servicing this client

**Returns:**

vHost hostPort servicing request

---

**isSecure**

```
public boolean isSecure()
```

Is this connection protected by either SSL or encryption (RTMPE, RTMPTE, RTMPS)

**Returns:**

is connection protected by either SSL or encryption (RTMPE, RTMPTE, RTMPS)

---

**isSSL**

```
public boolean isSSL()
```

Is this connection SSL (RTMPS)

**Returns:**

is connection (RTMPS)

---

**isEncrypted**

```
public boolean isEncrypted()
```

Is this connection encrypted (RTMPE or RTMPTE)

**Returns:**

is connection encrypted (RTMPE or RTMPTE)

---

**getIdleFrequency**

```
public int getIdleFrequency()
```

Get client idle frequency (milliseconds)

**Returns:**

client idle frequency (milliseconds)

## setIdleFrequency

```
public void setIdleFrequency(int idleFrequency)
```

Set client idle frequency (milliseconds)

**Parameters:**

idleFrequency - client idle frequency (milliseconds)

---

## getSharedObjectReadAccess

```
public String getSharedObjectReadAccess()
```

Get the shared object read access value. see setSharedObjectReadAccess for more information.

**Returns:**

shared object read access value

---

## setSharedObjectReadAccess

```
public void setSharedObjectReadAccess(String sharedObjectReadAccess)
```

Set the shared object read access value. This list is a semi-colon delimited list of shared objects that this client has access to. By default the value is set to IClient.READ\_ACCESS\_ALL which indicates this client can read all shared objects. If you want to disable read access to all shared objects set this value to IClient.READ\_ACCESS\_NONE. If this value is set to anything other than IClient.READ\_ACCESS\_ALL or IClient.READ\_ACCESS\_NONE then each item in the semi-color delimited list is compared against incoming shared object read requests. If any item in the list completely matches the shared object name or the start of the shared object name, then access is allowed. For example, if the sharedObjectReadAccess list is set to "testa/testb;testc" then the following requests would be granted the following access:

- testc: Granted Access
- testc/test: Granted Access
- testC/test: Denied Access (incorrect case)
- testa/testb: Granted Access
- testa/testb123: Granted Access
- testa/testb/file123: Granted Access
- testa/test: Denied Access (incomplete match)

Values in this list are always case sensitive.

**Parameters:**

sharedObjectReadAccess - shared object read access value

---

## getSharedObjectWriteAccess

```
public String getSharedObjectWriteAccess()
```

Get the shared object write access value. see setSharedObjectWriteAccess for more information.

**Returns:**

shared object write access value

---

## setSharedObjectWriteAccess

```
public void setSharedObjectWriteAccess(String sharedObjectWriteAccess)
```

(continued from last page)

Set the shared object write access value. This list is a semi-colon delimited list of shared objects that this client has access to. By default the value is set to IClient.WRITE\_ACCESS\_ALL which indicates this client can write all shared objects. If you want to disable write access to all shared objects set this value to IClient.WRITE\_ACCESS\_NONE. If this value is set to anything other than IClient.WRITE\_ACCESS\_ALL or IClient.WRITE\_ACCESS\_NONE then each item in the semi-color delimited list is compared against incoming shared object write requests. If any item in the list completely matches the shared object name or the start of the shared object name, then access is allowed. For example, if the sharedObjectWriteAccess list is set to "testa/testb;testc" then the following requests would be granted the following access:

- testc: Granted Access
- testc/test: Granted Access
- testC/test: Denied Access (incorrect case)
- testa/testb: Granted Access
- testa/testb123: Granted Access
- testa/testb/file123: Granted Access
- testa/test: Denied Access (incomplete match)

Values in this list are always case sensitive.

**Parameters:**

sharedObjectWriteAccess

## getStreamVideoSampleAccess

```
public String getStreamVideoSampleAccess()
```

Get the video sample access value. see setVideoStreamAccess for more information.

**Returns:**

video sample access

## setStreamVideoSampleAccess

```
public void setStreamVideoSampleAccess(String videoSampleAccess)
```

Set the stream video sample access value. This list is a semi-colon delimited list of stream names that this client has access to. By default the value is set to IClient.VIDEOSTREAM\_ACCESS\_NONE which indicates this client can access no stream data client side. If you want to enable access to all stream names set this value to IClient.VIDEOSTREAM\_ACCESS\_ALL. If this value is set to anything other than IClient.VIDEOSTREAM\_ACCESS\_ALL or IClient.VIDEOSTREAM\_ACCESS\_NONE then each item in the semi-color delimited list is compared against incoming stream play requests. If any item in the list completely matches the stream name or the start of the stream name, then access is allowed. For example, if the streamVideoSampleAccess list is set to "testa/testb;testc" then the following play requests would be granted the following access:

- testc: Granted Access
- testc/test: Granted Access
- testC/test: Denied Access (incorrect case)
- testa/testb: Granted Access
- testa/testb123: Granted Access
- testa/testb/file123: Granted Access
- testa/test: Denied Access (incomplete match)

Values in this list are always case sensitive.

**Parameters:**

videoSampleAccess

## getStreamAudioSampleAccess

```
public String getStreamAudioSampleAccess()
```

Get the audio sample access value. see setVideoStreamAccess for more information.

**Returns:**



(continued from last page)

audio sample access

---

## setStreamAudioSampleAccess

```
public void setStreamAudioSampleAccess(String audioSampleAccess)
```

Set the stream audio sample access value. This list is a semi-colon delimited list of stream names that this client has access to. By default the value is set to IClient.AUDIOSTREAM\_ACCESS\_NONE which indicates this client can access no stream data client side. If you want to enable access to all stream names set this value to IClient.AUDIOSTREAM\_ACCESS\_ALL. If this value is set to anything other than IClient.AUDIOSTREAM\_ACCESS\_ALL or IClient.AUDIOSTREAM\_ACCESS\_NONE then each item in the semi-color delimited list is compared against incoming stream play requests. If any item in the list completely matches the stream name or the start of the stream name, then access is allowed. For example, if the streamAudioSampleAccess list is set to "testa/testb;testc" then the following play requests would be granted the following access:

- testc: Granted Access
- testc/test: Granted Access
- testC/test: Denied Access (incorrect case)
- testa/testb: Granted Access
- testa/testb123: Granted Access
- testa/testb/file123: Granted Access
- testa/test: Denied Access (incomplete match)

Values in this list are always case sensitive.

### Parameters:

audioSampleAccess

---

## getStreamReadAccess

```
public String getStreamReadAccess()
```

Get the stream read access value. see setStreamReadAccess for more information.

### Returns:

stream read access value

---

## setStreamReadAccess

```
public void setStreamReadAccess(String streamReadAccess)
```

Set the stream object read access value. This list is a semi-colon delimited list of stream names that this client has access to. By default the value is set to IClient.READ\_ACCESS\_ALL which indicates this client can play all streams. If you want to disable read access to all stream names set this value to IClient.READ\_ACCESS\_NONE. If this value is set to anything other than IClient.READ\_ACCESS\_ALL or IClient.READ\_ACCESS\_NONE then each item in the semi-color delimited list is compared against incoming stream play requests. If any item in the list completely matches the stream name or the start of the stream name, then access is allowed. For example, if the streamReadAccess list is set to "testa/testb;testc" then the following play requests would be granted the following access:

- testc: Granted Access
- testc/test: Granted Access
- testC/test: Denied Access (incorrect case)
- testa/testb: Granted Access
- testa/testb123: Granted Access
- testa/testb/file123: Granted Access
- testa/test: Denied Access (incomplete match)

Values in this list are always case sensitive.

### Parameters:

streamReadAccess - shared object read access value

## getStreamWriteAccess

```
public String getStreamWriteAccess()
```

Get the stream write access value. see setStreamWriteAccess for more information.

**Returns:**

stream write access value

---

## setStreamWriteAccess

```
public void setStreamWriteAccess(String streamWriteAccess)
```

Set the stream object write access value. This list is a semi-colon delimited list of stream names that this client has access to. By default the value is set to IClient.WRITE\_ACCESS\_ALL which indicates this client can play all streams. If you want to disable write access to all stream names set this value to IClient.WRITE\_ACCESS\_NONE. If this value is set to anything other than IClient.WRITE\_ACCESS\_ALL or IClient.WRITE\_ACCESS\_NONE then each item in the semi-color delimited list is compared against incoming stream play requests. If any item in the list completely matches the stream name or the start of the stream name, then access is allowed. For example, if the streamWriteAccess list is set to "testa/testb;testc" then the following play requests would be granted the following access:

- testc: Granted Access
- testc/test: Granted Access
- testC/test: Denied Access (incorrect case)
- testa/testb: Granted Access
- testa/testb123: Granted Access
- testa/testb/file123: Granted Access
- testa/test: Denied Access (incomplete match)

Values in this list are always case sensitive.

**Parameters:**

streamWriteAccess - shared object read access value

---

## getWriteListener

```
public ClientWriteListener getWriteListener()
```

Object that tracks write operations

**Returns:**

Object that tracks write operations

---

## addAcceptConnectionAttribute

```
public void addAcceptConnectionAttribute(String key,  
    AMFDataObj item)
```

Add and attribute to the resultObj that gets passed back to the client on successful connection

**Parameters:**

key - key  
item - item to add

---

## addAcceptConnectionAttribute

```
public void addAcceptConnectionAttribute(String key,  
    String item)
```

Add and attribute to the resultObj that gets passed back to the client on successful connection

---

(continued from last page)

**Parameters:**

key - key  
item - item to add

---

**getRepeaterOriginUrl**

```
public String getRepeaterOriginUrl()
```

Get the origin URL used by the Live Stream Repeater

**Returns:**

URL used by the Live Stream Repeater

---

**setRepeaterOriginUrl**

```
public void setRepeaterOriginUrl(String repeaterOriginUrl)
```

Set the origin URL used by the Live Stream Repeater

**Parameters:**

repeaterOriginUrl - URL used by the Live Stream Repeater

---

**getLastValidateTime**

```
public long getLastValidateTime()
```

Get last time (millisecond) the connection was validated with a ping

**Returns:**

last time (millisecond) the connection was validated with a ping

---

**setLastValidateTime**

```
public void setLastValidateTime(long lastValidateTime)
```

Set the last time (milliseconds) the connection was validated with ping

**Parameters:**

lastValidateTime - last time (milliseconds) the connection was validated with ping

---

**getPingTimeout**

```
public int getPingTimeout()
```

Get the ping timeout (milliseconds)

**Returns:**

ping timeout (milliseconds)

---

**isLiveRepeater**

```
public boolean isLiveRepeater()
```

Returns true if this connection is from the live stream repeater

**Returns:**

true if this connection is from the live stream repeater

## isFlashVersionH264Capable

```
public boolean isFlashVersionH264Capable()
```

Returns true if the connected client is capable of playing H.264 video (Flash player 9.0.45.x or greater)

**Returns:**

returns true if the connected client is capable of playing H.264 video (Flash player 9.0.45.x or greater)

---

## isFlashVersion90115

```
public boolean isFlashVersion90115()
```

Returns true if the Flash version is equal or greater than 9.0.115.x

**Returns:**

returns true if the Flash version is equal or greater than 9.0.115.x

---

## isFlashVersion10

```
public boolean isFlashVersion10()
```

Returns true if the Flash version is equal or greater than 10.x.x.x

**Returns:**

returns true if the Flash version is equal or greater than 10.x.x.x

---

## isFlashMediaLiveEncoder

```
public boolean isFlashMediaLiveEncoder()
```

Returns true if this connection is the Flash Media Live Encoder

**Returns:**

true if this connection is the Flash Media Live Encoder

---

## testFlashVersion

```
public int testFlashVersion(int[] version)
```

Test to see if the connected client flash version is equal to or greater than a given value. Example: int isGood = testFlashVersion( { 9, 0, 5, 12} );

**Parameters:**

version - array of version values

**Returns:**

1 if greater, 0 if equal, -1 if less than

---

## isObjectEncodingAMF3

```
public boolean isObjectEncodingAMF3()
```

Is the object encoding for this client AMF3

**Returns:**

true of the object encoding for this client is AMF3

---

## isObjectEncodingAMF0

```
public boolean isObjectEncodingAMF0()
```

Is the object encoding for this client AMF0

**Returns:**

true of the object encoding for this client is AMF0

---

## setObjectEncoding

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

Set the object encoding level

**Parameters:**

objectEncoding - object encoding level

---

## getObjectEncoding

```
public int getObjectEncoding()
```

Get the object encoding level

**Returns:**

object encoding level

---

## setAcceptConnectionObj

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

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

**Parameters:**

acceptConnectionObj - connection object

---

## setAcceptConnectionDescription

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

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

**Parameters:**

description - description

---

## setAcceptConnectionExObj

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

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

**Parameters:**

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

---

(continued from last page)

## getRTPStream

```
public RTPStream getRTPStream( )
```

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

**Returns:**

underlying RTPStream object

---

## setAcceptConnection

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

Set the default for accept connection

**Parameters:**

acceptConnection - default for accept connection

---

## setShutdownClient

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

Gracefully and forcefully shutdown a client.

**Parameters:**

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

---

## reparentClient

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

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

**Parameters:**

vhost - new vhost

---

## getMaximumSetBufferTime

```
public int getMaximumSetBufferTime( )
```

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

**Returns:**

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

---

## getMaximumPendingWriteBytes

```
public int getMaximumPendingWriteBytes( )
```

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

**Returns:**

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

(continued from last page)

---

## getLiveStreamPacketizerList

```
public String getLiveStreamPacketizerList()
```

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

**Returns:**

comma separated list of LiveStreamPacketizers names

---

## setLiveStreamPacketizerList

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

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

**Parameters:**

liveStreamPacketizerList - comma separated list of LiveStreamPacketizers names

---

## 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)
```

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)
```

---

(continued from last page)

Subscribe to a live stream (use a particular mediaCasterType)

**Parameters:**

streamName - stream name

mediaCasterType - media caster type name

---

## fcUnSubscribeAll

```
public void fcUnSubscribeAll()
```

UnSubscribe to all streams that this client is current subscribed to

---

## fcUnSubscribe

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

UnSubscribe from a stream

**Parameters:**

streamName - stream name

---

## isValidateFMLEConnections

```
public boolean isValidateFMLEConnections()
```

Returns true if validating FMLE connection (default is false)

**Returns:**

true if validating FMLE connection

---

## setValidateFMLEConnections

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

Returns true if validating FMLE connection (default is false)

**Parameters:**

validateFMLEConnections - true if validating FMLE connection

---



## com.wowza.wms.client Interface IClientNotify

public interface **IClientNotify**  
extends

IClientNotify: listener interface used by [IApplicationInstance.addClientListener\(IClientNotify\)](#)

See Also:

[IApplicationInstance.addClientListener\(IClientNotify\)](#)

### Method Summary

void	<a href="#">onClientAccept</a> ( <a href="#">IClient</a> client) Triggered when client connection accepted
void	<a href="#">onClientConnect</a> ( <a href="#">IClient</a> client) Triggered when client attempt connection
void	<a href="#">onClientDisconnect</a> ( <a href="#">IClient</a> client) Triggered when client disconnected
void	<a href="#">onClientReject</a> ( <a href="#">IClient</a> client) Triggered when client connection rejected

### Methods

#### onClientConnect

public void **onClientConnect**([IClient](#) client)

Triggered when client attempt connection

**Parameters:**

client - client

#### onClientDisconnect

public void **onClientDisconnect**([IClient](#) client)

Triggered when client disconnected

**Parameters:**

client - client

#### onClientAccept

public void **onClientAccept**([IClient](#) client)

Triggered when client connection accepted

**Parameters:**

client - client

## onClientReject

```
public void onClientReject(IClient client)
```

Triggered when client connection rejected

**Parameters:**

client - client

## com.wowza.wms.client Interface IConnectionNotify

public interface **IConnectionNotify**  
extends

IConnectionNotify: internal class to manage connection counting

### Method Summary

void	<a href="#">onAcceptConnection</a> ( <a href="#">ConnectionCounter</a> connectionCounter, <a href="#">ConnectionHolder</a> connectionHolder, java.util.Date date, long stamp) Triggered when client connection accepted
void	<a href="#">onDisconnect</a> ( <a href="#">ConnectionCounter</a> connectionCounter, <a href="#">ConnectionHolder</a> connectionHolder, boolean isValid, java.util.Date date, long stamp) Triggered when client disconnected
void	<a href="#">onRejectConnection</a> ( <a href="#">ConnectionCounter</a> connectionCounter, <a href="#">ConnectionHolder</a> connectionHolder, int reason, java.util.Date date, long stamp) Triggered when client connection rejected

### Methods

#### onAcceptConnection

```
public void onAcceptConnection(ConnectionCounter connectionCounter,  
    ConnectionHolder connectionHolder,  
    java.util.Date date,  
    long stamp)
```

Triggered when client connection accepted

**Parameters:**

connectionCounter  
connectionHolder  
date  
stamp

#### onRejectConnection

```
public void onRejectConnection(ConnectionCounter connectionCounter,  
    ConnectionHolder connectionHolder,  
    int reason,  
    java.util.Date date,  
    long stamp)
```

Triggered when client connection rejected

**Parameters:**

connectionCounter  
connectionHolder  
reason

(continued from last page)

date  
stamp

---

## onDisconnect

```
public void onDisconnect(ConnectionCounter connectionCounter,  
    ConnectionHolder connectionHolder,  
    boolean isValid,  
    java.util.Date date,  
    long stamp)
```

Triggered when client disconnected

### Parameters:

connectionCounter  
connectionHolder  
isValid  
date  
stamp

---

## com.wowza.wms.client Interface IConnectionValidator

---

public interface **IConnectionValidator**  
extends

IConnectionValidator: callback for validating connections

---

### Method Summary

boolean	<a href="#">validateConnection</a> ( <a href="#">ConnectionCounter</a> connectionCounter, <a href="#">ConnectionHolder</a> connectionHolder, byte[] license) Triggered on connection
---------	--

---

### Methods

#### **validateConnection**

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

Triggered on connection

#### **Parameters:**

connectionCounter - connection counter  
connectionHolder - connection holder

#### **Returns:**

isValid

---

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](#)

### Field Summary

protected	<a href="#">authenticateHandler</a>
protected	<a href="#">authenticateHTTPProviderHandler</a>
protected	<a href="#">authenticationMethod</a>
protected	<a href="#">filters</a>
protected	<a href="#">properties</a>
protected	<a href="#">requestFilters</a>

### Constructor Summary

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

### Method Summary

boolean	<a href="#">canHandle</a> (String path)
boolean	<a href="#">doHTTPAuthentication</a> ( <a href="#">IVHost</a> vhost, <a href="#">IHTTPRequest</a> req, <a href="#">IHTTPResponse</a> resp)
String	<a href="#">getAuthenticationMethod</a> ()
String	<a href="#">getPath</a> ( <a href="#">IHTTPRequest</a> req, boolean removeFilter)
String	<a href="#">getRequestFilters</a> ()
void	<a href="#">init</a> ()

void	<a href="#">onBind</a> ( <a href="#">IVHost</a> vhost, <a href="#">HostPort</a> hostPort)
void	<a href="#">onUnbind</a> ( <a href="#">IVHost</a> vhost, <a href="#">HostPort</a> hostPort)
void	<a href="#">setAuthenticationMethod</a> (String authenticationMethod)
void	<a href="#">setProperties</a> ( <a href="#">WMSProperties</a> properties)
void	<a href="#">setRequestFilters</a> (String requestFilters)

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

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

#### Methods inherited from interface [com.wowza.wms.http.IHTTPProvider2](#)

[canHandle](#), [getAuthenticationMethod](#), [getRequestFilters](#), [init](#), [setAuthenticationMethod](#), [setRequestFilters](#)

#### Methods inherited from interface [com.wowza.wms.http.IHTTPProvider](#)

[onBind](#), [onHTTPRequest](#), [onUnbind](#), [setProperties](#)

## Fields

### properties

protected [com.wowza.wms.application.WMSProperties](#) **properties**

### filters

protected [java.util.List](#) **filters**

### requestFilters

protected [java.lang.String](#) **requestFilters**

### authenticateHandler

protected [com.wowza.wms.authentication.IAuthenticate](#) **authenticateHandler**



(continued from last page)

## authenticateHTTPProviderHandler

```
protected com.wowza.wms.authentication.IAuthenticateHTTPProvider  
authenticateHTTPProviderHandler
```

## authenticationMethod

```
protected java.lang.String authenticationMethod
```

## Constructors

### HTTPProvider2Base

```
public HTTPProvider2Base()
```

## Methods

### init

```
public void init()
```

### canHandle

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

### getPath

```
public String getPath(IHTTPRequest req,  
    boolean removeFilter)
```

### setRequestFilters

```
public void setRequestFilters(String requestFilters)
```

### getRequestFilters

```
public String getRequestFilters()
```

### setProperties

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

(continued from last page)

---

## onBind

```
public void onBind(IVHost vhost,  
                  HostPort hostPort)
```

---

## onUnbind

```
public void onUnbind(IVHost vhost,  
                   HostPort hostPort)
```

---

## getAuthenticationMethod

```
public String getAuthenticationMethod()
```

---

## setAuthenticationMethod

```
public void setAuthenticationMethod(String authenticationMethod)
```

---

## doHTTPAuthentication

```
public boolean doHTTPAuthentication(IVHost vhost,  
                                     IHTTPRequest req,  
                                     IHTTPResponse resp)
```

## com.wowza.wms.http Interface IHTTPProvider

All Subinterfaces:

[IHTTPProvider2](#)

public interface **IHTTPProvider**  
extends

IHTTPProvider: HTTP provider class for a given HostPort definition. Receive all HTTP requests that are not RTMPT requests. See IHTTPProvider2 for the extended version of this interface.

### Method Summary

void	<a href="#">onBind</a> ( <a href="#">IVHost</a> vhost, <a href="#">HostPort</a> hostPort) Triggered after hostPort binds to socket
void	<a href="#">onHTTPRequest</a> ( <a href="#">IVHost</a> vhost, <a href="#">IHTTPRequest</a> req, <a href="#">IHTTPResponse</a> resp) Triggered for each HTTP request to the given hostPort that is not an RTMPT request.
void	<a href="#">onUnbind</a> ( <a href="#">IVHost</a> vhost, <a href="#">HostPort</a> hostPort) Triggered after hostPort unbinds
void	<a href="#">setProperties</a> ( <a href="#">WMSProperties</a> properties) Called to provide properties set in configuration files

### Methods

#### onBind

```
public void onBind(IVHost vhost,  
                   HostPort hostPort)
```

Triggered after hostPort binds to socket

**Parameters:**

vhost - parent vhost  
hostPort - host port definition

#### onHTTPRequest

```
public void onHTTPRequest(IVHost vhost,  
                           IHTTPRequest req,  
                           IHTTPResponse resp)
```

Triggered for each HTTP request to the given hostPort that is not an RTMPT request.

**Parameters:**

vhost - parent vhost  
req - HTML request object  
resp - HTML response object

---

## onUnbind

```
public void onUnbind(IVHost vhost,  
    HostPort hostPort)
```

Triggered after hostPort unbinds

### Parameters:

vhost - parent vhost

hostPort - host port definition

---

## setProperties

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

Called to provide properties set in configuration files

## com.wowza.wms.http Interface IHTTPProvider2

All Superinterfaces:

[IHTTPProvider](#)

All Known Implementing Classes:

[HTTPProvider2Base](#)

---

public interface **IHTTPProvider2**  
extends [IHTTPProvider](#)

IHTTPProvider2: Extension of IHTTPProvider that adds support for multiple HTTPProviders attached to a single HostPort along with authentication. Multiple HTTPProviders can be added to a HostPort definition. They are configured as follows:

```
<HTTPProvider>
  <BaseClass>com.wowza.wms.http.HTTPServerVersion</BaseClass>
  <RequestFilters>*serverversion</RequestFilters>
  <AuthenticationMethod>none</AuthenticationMethod>
</HTTPProvider>
```

Below is an sample implementation of the HTTPServerVersion provider:

```

import java.io.*;

import com.wowza.wms.server.*;
import com.wowza.wms.stream.*;
import com.wowza.wms.vhost.*;
import com.wowza.wms.logging.*;

public class HTTPServerVersion extends HTTPProvider2Base
{
    public void onHTTPRequest(IVHost vhost, IHTTPRequest req, IHTTPResponse resp)
    {
        if (!doHTTPAuthentication(vhost, req, resp))
            return;

        String version = MediaStreamBase.p+" "+ReleaseInfo.getVersion()+"
build"+ReleaseInfo.getBuildNumber();
        String retStr =
"<html><head><title>"+version+"</title></head><body>"+version+"</body></html>";
        try
        {
            OutputStream out = resp.getOutputStream();
            byte[] outBytes = retStr.getBytes();
            out.write(outBytes);
        }
        catch (Exception e)
        {
            WMSLoggerFactory.getLogger(HTTPServerVersion.class).error("HTMLServerVersion:
"+e.toString());
        }
    }
}

```

## Method Summary

boolean	<a href="#">canHandle</a> (String path)
String	<a href="#">getAuthenticationMethod</a> ()
String	<a href="#">getRequestFilters</a> ()
void	<a href="#">init</a> ()
void	<a href="#">setAuthenticationMethod</a> (String authenticationMethod)
void	<a href="#">setRequestFilters</a> (String requestFilters)

---

Methods inherited from interface [com.wowza.wms.http.IHTTPProvider](#)

[onBind](#), [onHTTPRequest](#), [onUnbind](#), [setProperties](#)

---

## Methods

### canHandle

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

---

### setRequestFilters

```
public void setRequestFilters(String requestFilters)
```

---

### getRequestFilters

```
public String getRequestFilters()
```

---

### init

```
public void init()
```

---

### getAuthenticationMethod

```
public String getAuthenticationMethod()
```

---

### setAuthenticationMethod

```
public void setAuthenticationMethod(String authenticationMethod)
```

---

## com.wowza.wms.http Interface IHttpRequest

public interface **IHttpRequest**  
extends

### Method Summary

int	<a href="#"><code>getContentLength()</code></a> Get the content length of the body of the message
String	<a href="#"><code>getContentType()</code></a> Get the request content type
String	<a href="#"><code>getHeader(String name)</code></a> Get a HTTP header value such as 'Content-Length'
java.util.Set	<a href="#"><code>getHeaderNames()</code></a> Get a Set of the header names
java.io.InputStream	<a href="#"><code>getInputStream()</code></a> Get the body of the message as an input stream
int	<a href="#"><code>getIntHeader(String name)</code></a> Get a HTTP header value such as 'Content-Length' and return as int
java.util.Locale	<a href="#"><code>getLocale()</code></a> Get locale of request (Example: en-us)
String	<a href="#"><code>getMethod()</code></a> Get the method invocation method: GET, POST, HEAD
String	<a href="#"><code>getParameter(String name)</code></a> Get a parameter value
java.util.Map	<a href="#"><code>getParameterMap()</code></a> Get the entire parameter Map
java.util.Set	<a href="#"><code>getParameterNames()</code></a> Get a Set of parameter names
String[]	<a href="#"><code>getParameterValues(String name)</code></a> Get a multi-value parameter as an array of String
String	<a href="#"><code>getProtocol()</code></a> Get the request protocol (example: HTTP/1.1)
String	<a href="#"><code>getQueryString()</code></a> Get the query string part of the url (everything after the ?)
String	<a href="#"><code>getRemoteAddr()</code></a> Get the remote ip address of the request
String	<a href="#"><code>getRemoteHost()</code></a> Get the remote host name (if known) if not return ip address



String	<a href="#"><code>getRequestURI()</code></a> Get the full request URI
String	<a href="#"><code>getRequestURL()</code></a> Get the request url (same as URI minus the query string)
String	<a href="#"><code>getScheme()</code></a> Get the request scheme (Example "http")
String	<a href="#"><code>getServerName()</code></a> Get the name of the server (Example: "Wowza Media Server Pro")
int	<a href="#"><code>getServerPort()</code></a> Get the port this request was received on
boolean	<a href="#"><code>isSecure()</code></a> Returns true if the request is protected by SSL
void	<a href="#"><code>parseBodyForParams()</code></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="#"><code>parseBodyForParams(boolean doDecode)</code></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

### getHeader

```
public String getHeader(String name)
```

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

**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

(continued from last page)

**Returns:**

Set of header names

---

**getMethod**

```
public String getMethod()
```

Get the method invocation method: GET, POST, HEAD

**Returns:**

method

---

**getQueryString**

```
public String getQueryString()
```

Get the query string part of the url (everything after the ?)

**Returns:**

query string

---

**getContentLength**

```
public int getContentLength()
```

Get the content length of the body of the message

**Returns:**

content length of the body of the message

---

**getRequestURI**

```
public String getRequestURI()
```

Get the full request URI

**Returns:**

full request URI

---

**getRequestURL**

```
public String getRequestURL()
```

Get the request url (same as URI minus the query string)

**Returns:**

request url

---

**getContentType**

```
public String getContentType()
```

Get the request content type

**Returns:**

request content type

---

## parseBodyForParams

```
public void parseBodyForParams(boolean doDecode)
```

If the body of the message contains parameter data (data in name value pairs separated by & character) call this routine to decode those parameters and add them to the parameter map.

**Parameters:**

doDecode - true to decode the params as though they are url params

---

## parseBodyForParams

```
public void parseBodyForParams()
```

If the body of the message contains parameter data (data in name value pairs separated by & character) call this routine to decode those parameters and add them to the parameter map.

---

## getParameter

```
public String getParameter(String name)
```

Get a parameter value

**Parameters:**

name - parameter name

**Returns:**

parameter value

---

## getParameterNames

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

Get a Set of parameter names

**Returns:**

Set of parameter names

---

## getParameterValues

```
public String[] getParameterValues(String name)
```

Get a multi-value parameter as an array of String

**Parameters:**

name - parameter name

**Returns:**

multi-value parameter as an array of String

---

## getParameterMap

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

Get the entire parameter Map

**Returns:**

parameter Map

---

---

## getInputStream

```
public java.io.InputStream getInputStream()
```

Get the body of the message as an input stream

**Returns:**

body of the message as an input stream

---

## getProtocol

```
public String getProtocol()
```

Get the request protocol (example: HTTP/1.1)

**Returns:**

request protocol

---

## getScheme

```
public String getScheme()
```

Get the request scheme (Example "http")

**Returns:**

request scheme

---

## getServerName

```
public String getServerName()
```

Get the name of the server (Example: "Wowza Media Server Pro")

**Returns:**

name of the server

---

## getServerPort

```
public int getServerPort()
```

Get the port this request was received on

**Returns:**

the port this request was received on

---

## getRemoteAddr

```
public String getRemoteAddr()
```

Get the remote ip address of the request

**Returns:**

remote ip address of the request

---

## getRemoteHost

```
public String getRemoteHost()
```

---

(continued from last page)

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 if the request is protected by SSL

**Returns:**

true if the request is protected by SSL

## com.wowza.wms.http Interface IHTTPResponse

public interface **IHTTPResponse**  
extends

### Method Summary

boolean	<a href="#"><code>containsHeader</code></a> (String name) Returns true if reponse header contains parameter name
String	<a href="#"><code>getHeader</code></a> (String name) Get header value
java.util.Map	<a href="#"><code>getHeaders</code></a> () Get the current response headers as a map
int	<a href="#"><code>getHeaderSize</code></a> () Get the size in bytes of the HTTP header
int	<a href="#"><code>getIntHeader</code></a> (String name) Get header value as int
java.io.OutputStream	<a href="#"><code>getOutputStream</code></a> () Get the output stream for the reponse.
void	<a href="#"><code>setHeader</code></a> (String name, String value) Set header value
void	<a href="#"><code>setIntHeader</code></a> (String name, int value) Set header value as int
void	<a href="#"><code>setResponseCode</code></a> (int responseCode) Set the HTTP response code

### Methods

#### `getOutputStream`

public java.io.OutputStream **getOutputStream**()

Get the output stream for the reponse. You can then write directly into the output stream.

**Returns:**

output stream for the reponse

#### `containsHeader`

public boolean **containsHeader**(String name)

Returns true if reponse header contains parameter name

(continued from last page)

**Parameters:**

name - header parameter name

**Returns:**true if header contains value

---

**setHeader**

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

Set header value

**Parameters:**

name - header parameter name

value - parameter value

---

**setIntHeader**

```
public void setIntHeader(String name,  
                          int value)
```

Set header value as int

**Parameters:**

name - header parameter name

value - parameter value

---

**getHeaders**

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

Get the current response headers as a map

**Returns:**current response headers as a map

---

**getHeader**

```
public String getHeader(String name)
```

Get header value

**Parameters:**

name - header parameter name

**Returns:**parameter value

---

**getIntHeader**

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

Get header value as int

**Parameters:**

name - header parameter name

**Returns:**

(continued from last page)

parameter value

---

## setResponseCode

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

Set the HTTP response code

**Parameters:**

responseCode - HTTP response code

---

## getHeaderSize

```
public int getHeaderSize()
```

Get the size in bytes of the HTTP header

**Returns:**

header size in bytes



---

Package

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

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

public interface IHTTPStreamerCupertinoVODActionNotify  
extends

### Method Summary

void	<a href="#">onCreate</a> (IHTTPStreamerCupertinoIndex fileIndex, <a href="#">IHTTPStreamerApplicationContext</a> appContext, <a href="#">IHTTPStreamerSession</a> httpStreamerSession, String rawStreamName, String streamExt, String streamName)
void	<a href="#">onDestroy</a> (IHTTPStreamerCupertinoIndex fileIndex)
void	<a href="#">onFillChunkEnd</a> (IHTTPStreamerCupertinoIndex fileIndex, <a href="#">IHTTPStreamerCupertinoIndexItem</a> item, <a href="#">LiveStreamPacketizerCupertinoChunk</a> chunk, boolean audioOnly)
void	<a href="#">onFillChunkStart</a> (IHTTPStreamerCupertinoIndex fileIndex, <a href="#">IHTTPStreamerCupertinoIndexItem</a> item, <a href="#">LiveStreamPacketizerCupertinoChunk</a> chunk, boolean audioOnly)
void	<a href="#">onIndex</a> (IHTTPStreamerCupertinoIndex fileIndex, <a href="#">IHTTPStreamerApplicationContext</a> appContext, <a href="#">IHTTPStreamerSession</a> httpStreamerSession, String rawStreamName, String streamExt, String streamName)
void	<a href="#">onInit</a> (IHTTPStreamerCupertinoIndex fileIndex, <a href="#">IHTTPStreamerApplicationContext</a> appContext, <a href="#">IHTTPStreamerSession</a> httpStreamerSession, String rawStreamName, String streamExt, String streamName)
void	<a href="#">onOpen</a> (IHTTPStreamerCupertinoIndex fileIndex, <a href="#">IHTTPStreamerApplicationContext</a> appContext, <a href="#">IHTTPStreamerSession</a> httpStreamerSession, String rawStreamName, String streamExt, String streamName)

### Methods

#### onCreate

```
public void onCreate(IHTTPStreamerCupertinoIndex fileIndex,
    IHTTPStreamerApplicationContext appContext,
    IHTTPStreamerSession httpStreamerSession,
    String rawStreamName,
    String streamExt,
    String streamName)
```

(continued from last page)

---

## onInit

```
public void onInit(IHTTPStreamerCupertinoIndex fileIndex,  
    IHTTPStreamerApplicationContext appContext,  
    IHTTPStreamerSession httpStreamerSession,  
    String rawStreamName,  
    String streamExt,  
    String streamName)
```

---

## onOpen

```
public void onOpen(IHTTPStreamerCupertinoIndex fileIndex,  
    IHTTPStreamerApplicationContext appContext,  
    IHTTPStreamerSession httpStreamerSession,  
    String rawStreamName,  
    String streamExt,  
    String streamName)
```

---

## onIndex

```
public void onIndex(IHTTPStreamerCupertinoIndex fileIndex,  
    IHTTPStreamerApplicationContext appContext,  
    IHTTPStreamerSession httpStreamerSession,  
    String rawStreamName,  
    String streamExt,  
    String streamName)
```

---

## onFillChunkStart

```
public void onFillChunkStart(IHTTPStreamerCupertinoIndex fileIndex,  
    IHTTPStreamerCupertinoIndexItem item,  
    LiveStreamPacketizerCupertinoChunk chunk,  
    boolean audioOnly)
```

---

## onFillChunkEnd

```
public void onFillChunkEnd(IHTTPStreamerCupertinoIndex fileIndex,  
    IHTTPStreamerCupertinoIndexItem item,  
    LiveStreamPacketizerCupertinoChunk chunk,  
    boolean audioOnly)
```

---

## onDestroy

```
public void onDestroy(IHTTPStreamerCupertinoIndex fileIndex)
```

---

Package

**com.wowza.wms.httpstreamer.model**

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

public interface **IHTTPStreamerAdapter**  
extends

### Method Summary

boolean	<a href="#"><u>canHandle</u></a> (String path)
String	<a href="#"><u>getAdapterName</u></a> ()
HTTPStreamerItem	<a href="#"><u>getHTTPStreamerItem</u></a> ()
String	<a href="#"><u>getID</u></a> ()
int	<a href="#"><u>getIdleFrequency</u></a> ()
<a href="#"><u>WMSProperties</u></a>	<a href="#"><u>getProperties</u></a> ()
<a href="#"><u>IVHost</u></a>	<a href="#"><u>getVHost</u></a> ()
void	<a href="#"><u>init</u></a> ()
void	<a href="#"><u>service</u></a> (org.apache.mina.common.Session session, RtmpRequestMessage req, RtmpResponseMessage resp)
void	<a href="#"><u>setHTTPStreamerItem</u></a> (HTTPStreamerItem httpStreamerItem)
void	<a href="#"><u>setID</u></a> (String id)
void	<a href="#"><u>setProperties</u></a> ( <a href="#"><u>WMSProperties</u></a> properties)
void	<a href="#"><u>setVHost</u></a> ( <a href="#"><u>IVHost</u></a> vhost)
void	<a href="#"><u>shutdownSession</u></a> ( <a href="#"><u>IHTTPStreamerSession</u></a> session)

### Methods

#### canHandle

public boolean **canHandle**(String path)

(continued from last page)

## service

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

---

## getProperties

```
public WMSProperties getProperties()
```

---

## setProperties

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

---

## getHTTPStreamerItem

```
public HTTPStreamerItem getHTTPStreamerItem()
```

---

## setHTTPStreamerItem

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

---

## getVHost

```
public IVHost getVHost()
```

---

## setVHost

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

---

## init

```
public void init()
```

---

## shutdownSession

```
public void shutdownSession(IHTTPStreamerSession session)
```

---

(continued from last page)

**getIdleFrequency**

```
public int getIdleFrequency()
```

---

**getAdapterName**

```
public String getAdapterName()
```

---

**getID**

```
public String getID()
```

---

**setID**

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

---

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

public interface **IHTTPStreamerApplicationContext**  
extends

### Method Summary

<a href="#">IApplicationInstance</a>	<a href="#">getAppInstance()</a>
<a href="#">WMSProperties</a>	<a href="#">getProperties()</a>
<a href="#">IVHost</a>	<a href="#">getVHost()</a>
void	<a href="#">init(IApplicationInstance</a> appInstance, HTTPStreamerItem httpStreamerItem)

### Methods

#### **init**

public void **init**([IApplicationInstance](#) appInstance,  
HTTPStreamerItem httpStreamerItem)

#### **getProperties**

public [WMSProperties](#) **getProperties()**

#### **getAppInstance**

public [IApplicationInstance](#) **getAppInstance()**

#### **getVHost**

public [IVHost](#) **getVHost()**



# com.wowza.wms.httpstreamer.model

## Interface IHTTPStreamerRepeater

public interface IHTTPStreamerRepeater  
extends

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

### Field Summary

public static final	<a href="#"><u>SESSIONPROTOCOL_COUNT</u></a> Value: <b>4</b>
public static final	<a href="#"><u>SESSIONPROTOCOL_CUPERTINOSTREAMING</u></a> Value: <b>1</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_UNKNOWN</u></a> Value: <b>0</b>
public static final	<a href="#"><u>SESSIONTYPE_VOD</u></a> Value: <b>2</b>

### Method Summary

void	<a href="#"><u>acceptSession()</u></a>
void	<a href="#"><u>addIOPerformance()</u></a> ( <a href="#"><u>IOPerformanceCounter</u></a> totalIOPerformanceResult)
void	<a href="#"><u>addIOPerformance2()</u></a> ( <a href="#"><u>IOPerformanceCounter</u></a> totalIOPerformanceResult)
void	<a href="#"><u>addStreamDomainStr()</u></a> (String streamDomainStr)

void	<a href="#"><u>addStreamDomainStrs</u></a> (java.util.List streamNames)
boolean	<a href="#"><u>checkAndSetPlayLogged</u></a> ()
void	<a href="#"><u>clearLoggingValues</u></a> ()
boolean	<a href="#"><u>containsStreamDomainStr</u></a> (String streamDomainStr)
boolean	<a href="#"><u>containsStreamNameParts</u></a> (String streamName)
<a href="#"><u>IApplicationInstance</u></a>	<a href="#"><u>getAppInstance</u></a> ()
ConnectionHolder	<a href="#"><u>getConnectionHolder</u></a> ()
String	<a href="#"><u>getCookieStr</u></a> ()
<a href="#"><u>ElapsedTimer</u></a>	<a href="#"><u>getElapsedTime</u></a> ()
HTTPStreamerFileInfo	<a href="#"><u>getFileInfo</u></a> (String streamName)
<a href="#"><u>IHTTPStreamerAdapter</u></a>	<a href="#"><u>getHTTPStreamerAdapter</u></a> ()
<a href="#"><u>IOPerformanceCounter</u></a>	<a href="#"><u>getIOPerformanceCounter</u></a> ()
String	<a href="#"><u>getIpAddress</u></a> ()
String	<a href="#"><u>getLiveStreamingPacketizer</u></a> ()
Object	<a href="#"><u>getLock</u></a> ()
String	<a href="#"><u>getQueryStr</u></a> ()
String	<a href="#"><u>getReferrer</u></a> ()
String	<a href="#"><u>getServerIp</u></a> ()
int	<a href="#"><u>getServerPort</u></a> ()
String	<a href="#"><u>getSessionId</u></a> ()
int	<a href="#"><u>getSessionProtocol</u></a> ()
int	<a href="#"><u>getSessionTimeout</u></a> ()
int	<a href="#"><u>getSessionType</u></a> ()
<a href="#"><u>IMediaStream</u></a>	<a href="#"><u>getStream</u></a> ()

String	<a href="#"><code>getStreamExt()</code></a>
String	<a href="#"><code>getStreamName()</code></a>
HTTPStreamerStreamNameParts	<a href="#"><code>getStreamNameParts(String streamName)</code></a>
long	<a href="#"><code>getStreamPosition()</code></a>
String	<a href="#"><code>getTimeRunning()</code></a>
double	<a href="#"><code>getTimeRunningSeconds()</code></a>
String	<a href="#"><code>getUri()</code></a>
String	<a href="#"><code>getUserAgent()</code></a>
<a href="#"><code>IVHost</code></a>	<a href="#"><code>getVHost()</code></a>
boolean	<a href="#"><code>isAcceptSession()</code></a>
boolean	<a href="#"><code>isActive()</code></a>
boolean	<a href="#"><code>isFileInfo(String streamName)</code></a>
boolean	<a href="#"><code>isPlayLogged()</code></a>
boolean	<a href="#"><code>isTimeout(long timecode)</code></a>
boolean	<a href="#"><code>isTimeoutSession()</code></a>
boolean	<a href="#"><code>isValidated()</code></a>
boolean	<a href="#"><code>isValidStreamDomainStr(String streamDomainStr)</code></a>
void	<a href="#"><code>lockRepeaterStreams(java.util.List streamNames, String liveStreamPacketizer, String liveStreamRepeater, String streamTypeStr)</code></a>
void	<a href="#"><code>putFileInfo(String streamName, HTTPStreamerFileInfo fileInfo)</code></a>
void	<a href="#"><code>putStreamNameParts(String streamName, HTTPStreamerStreamNameParts streamNameParts)</code></a>
void	<a href="#"><code>rejectSession()</code></a>
void	<a href="#"><code>removeStreamDomainStr(String streamDomainStr)</code></a>
void	<a href="#"><code>setAcceptSession(boolean acceptSession)</code></a>

void	<a href="#"><u>setActive</u></a> (boolean isActive)
void	<a href="#"><u>setAppInstance</u></a> ( <a href="#"><u>IApplicationInstance</u></a> appInstance)
void	<a href="#"><u>setCookieStr</u></a> (String cookieStr)
void	<a href="#"><u>setHTTPStreamerAdapter</u></a> ( <a href="#"><u>IHTTPStreamerAdapter</u></a> httpStreamerAdapter)
void	<a href="#"><u>setIpAddress</u></a> (String ipAddress)
void	<a href="#"><u>setLiveStreamingPacketizer</u></a> (String liveStreamingPacketizer)
void	<a href="#"><u>setPlayLogged</u></a> (boolean isPlayLogged)
void	<a href="#"><u>setQueryStr</u></a> (String queryStr)
void	<a href="#"><u>setReferrer</u></a> (String referrer)
void	<a href="#"><u>setServerIp</u></a> (String serverIp)
void	<a href="#"><u>setServerPort</u></a> (int serverPort)
void	<a href="#"><u>setSessionId</u></a> (String sessionId)
void	<a href="#"><u>setSessionProtocol</u></a> (int sessionProtocol)
void	<a href="#"><u>setSessionTimeout</u></a> (int sessionTimeout)
void	<a href="#"><u>setSessionType</u></a> (int sessionType)
void	<a href="#"><u>setStream</u></a> ( <a href="#"><u>IMediaStream</u></a> stream)
void	<a href="#"><u>setStreamExt</u></a> (String streamExt)
void	<a href="#"><u>setStreamName</u></a> (String streamName)
void	<a href="#"><u>setStreamPosition</u></a> (long streamPosition)
void	<a href="#"><u>setTimeoutSession</u></a> (boolean timeoutSession)
void	<a href="#"><u>setUri</u></a> (String uri)
void	<a href="#"><u>setUserAgent</u></a> (String userAgent)
void	<a href="#"><u>setVHost</u></a> ( <a href="#"><u>IVHost</u></a> vhost)
void	<a href="#"><u>shutdown</u></a> ()

void	<a href="#">touch</a> (long timecode)
void	<a href="#">updateLoggingValues</a> ()

## 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**

### 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**

(continued from last page)

## SESSIONPROTOCOL\_WEBMSTREAMING

```
public static final int SESSIONPROTOCOL_WEBMSTREAMING
```

Constant value: **3**

## SESSIONPROTOCOL\_COUNT

```
public static final int SESSIONPROTOCOL_COUNT
```

Constant value: **4**

## Methods

### getSessionId

```
public String getSessionId()
```

### setSessionId

```
public void setSessionId(String sessionId)
```

### getVHost

```
public IVHost getVHost()
```

### setVHost

```
public void setVHost(IVHost vhost)
```

### touch

```
public void touch(long timecode)
```

### isTimeout

```
public boolean isTimeout(long timecode)
```

### getLock

```
public Object getLock()
```

---

## shutdown

```
public void shutdown()
```

---

---

## isActive

```
public boolean isActive()
```

---

---

## setActive

```
public void setActive(boolean isActive)
```

---

---

## getStream

```
public IMediaStream getStream()
```

---

---

## setStream

```
public void setStream(IMediaStream stream)
```

---

---

## isTimeoutSession

```
public boolean isTimeoutSession()
```

---

---

## setTimeoutSession

```
public void setTimeoutSession(boolean timeoutSession)
```

---

---

## getSessionTimeout

```
public int getSessionTimeout()
```

---

---

## setSessionTimeout

```
public void setSessionTimeout(int sessionTimeout)
```

---

---

## isValidated

```
public boolean isValidated()
```

---



(continued from last page)

---

## checkAndSetPlayLogged

```
public boolean checkAndSetPlayLogged()
```

---

## isPlayLogged

```
public boolean isPlayLogged()
```

---

## setPlayLogged

```
public void setPlayLogged(boolean isPlayLogged)
```

---

## addIOPerformance

```
public void addIOPerformance(IOPerformanceCounter totalIOPerformanceResult)
```

---

## addIOPerformance2

```
public void addIOPerformance2(IOPerformanceCounter totalIOPerformanceResult)
```

---

## getConnectionHolder

```
public ConnectionHolder getConnectionHolder()
```

---

## getHTTPStreamerAdapter

```
public IHTTPStreamerAdapter getHTTPStreamerAdapter()
```

---

## setHTTPStreamerAdapter

```
public void setHTTPStreamerAdapter(IHTTPStreamerAdapter httpStreamerAdapter)
```

---

## getAppInstance

```
public IApplicationInstance getAppInstance()
```

---

(continued from last page)

## setAppInstance

```
public void setAppInstance(IApplicationInstance appInstance)
```

---

## getSessionType

```
public int getSessionType()
```

---

## setSessionType

```
public void setSessionType(int sessionType)
```

---

## getLiveStreamingPacketizer

```
public String getLiveStreamingPacketizer()
```

---

## setLiveStreamingPacketizer

```
public void setLiveStreamingPacketizer(String liveStreamingPacketizer)
```

---

## getIpAddress

```
public String getIpAddress()
```

---

## setIpAddress

```
public void setIpAddress(String ipAddress)
```

---

## updateLoggingValues

```
public void updateLoggingValues()
```

---

## clearLoggingValues

```
public void clearLoggingValues()
```

---

## getSessionProtocol

```
public int getSessionProtocol()
```

---

(continued from last page)

---

**setSessionProtocol**

```
public void setSessionProtocol(int sessionProtocol)
```

---

**getServerIp**

```
public String getServerIp()
```

---

**setServerIp**

```
public void setServerIp(String serverIp)
```

---

**getServerPort**

```
public int getServerPort()
```

---

**setServerPort**

```
public void setServerPort(int serverPort)
```

---

**getUserAgent**

```
public String getUserAgent()
```

---

**setUserAgent**

```
public void setUserAgent(String userAgent)
```

---

**getUri**

```
public String getUri()
```

---

**setUri**

```
public void setUri(String uri)
```

---

(continued from last page)

**getReferrer**

```
public String getReferrer()
```

---

**setReferrer**

```
public void setReferrer(String referrer)
```

---

**getQueryStr**

```
public String getQueryStr()
```

---

**setQueryStr**

```
public void setQueryStr(String queryStr)
```

---

**lockRepeaterStreams**

```
public void lockRepeaterStreams(java.util.List streamNames,  
    String liveStreamPacketizer,  
    String liveStreamRepeater,  
    String streamTypeStr)
```

---

**rejectSession**

```
public void rejectSession()
```

---

**acceptSession**

```
public void acceptSession()
```

---

**isAcceptSession**

```
public boolean isAcceptSession()
```

---

**setAcceptSession**

```
public void setAcceptSession(boolean acceptSession)
```

---

(continued from last page)

**getCookieStr**

```
public String getCookieStr()
```

---

**setCookieStr**

```
public void setCookieStr(String cookieStr)
```

---

**getStreamName**

```
public String getStreamName()
```

---

**setStreamName**

```
public void setStreamName(String streamName)
```

---

**getStreamExt**

```
public String getStreamExt()
```

---

**setStreamExt**

```
public void setStreamExt(String streamExt)
```

---

**getStreamNameParts**

```
public HTTPStreamerStreamNameParts getStreamNameParts(String streamName)
```

---

**containsStreamNameParts**

```
public boolean containsStreamNameParts(String streamName)
```

---

**putStreamNameParts**

```
public void putStreamNameParts(String streamName,  
    HTTPStreamerStreamNameParts streamNameParts)
```

---

**getStreamPosition**

```
public long getStreamPosition()
```

---

(continued from last page)

---

### setStreamPosition

```
public void setStreamPosition(long streamPosition)
```

---

### getIOPerformanceCounter

```
public IOPerformanceCounter getIOPerformanceCounter()
```

---

### getFileInfo

```
public HTTPStreamerFileInfo getFileInfo(String streamName)
```

---

### putFileInfo

```
public void putFileInfo(String streamName,  
    HTTPStreamerFileInfo fileInfo)
```

---

### isFileInfo

```
public boolean isFileInfo(String streamName)
```

---

### isValidStreamDomainStr

```
public boolean isValidStreamDomainStr(String streamDomainStr)
```

---

### containsStreamDomainStr

```
public boolean containsStreamDomainStr(String streamDomainStr)
```

---

### removeStreamDomainStr

```
public void removeStreamDomainStr(String streamDomainStr)
```

---

### addStreamDomainStr

```
public void addStreamDomainStr(String streamDomainStr)
```

---

(continued from last page)

## addStreamDomainStrs

```
public void addStreamDomainStrs(java.util.List streamNames)
```

---

## getElapsedTime

```
public ElapsedTimer getElapsedTime()
```

---

## getTimeRunning

```
public String getTimeRunning()
```

---

## getTimeRunningSeconds

```
public double getTimeRunningSeconds()
```

---

---

## com.wowza.wms.httpstreamer.model Interface IHTTPStreamerSessionNotify

---

public interface IHTTPStreamerSessionNotify  
extends

---

### Method Summary

void	<a href="#">onHTTPStreamerSessionCreate</a> ( <a href="#">IHTTPStreamerSession</a> httpStreamerSession)
void	<a href="#">onHTTPStreamerSessionDestroy</a> ( <a href="#">IHTTPStreamerSession</a> httpStreamerSession)

---

### Methods

#### onHTTPStreamerSessionCreate

public void **onHTTPStreamerSessionCreate**([IHTTPStreamerSession](#) httpStreamerSession)

---

#### onHTTPStreamerSessionDestroy

public void **onHTTPStreamerSessionDestroy**([IHTTPStreamerSession](#) httpStreamerSession)



---

Package

**com.wowza.wms.httpstreamer.smoothstreaming.model**

## com.wowza.wms.httpstreamer.smoothstreaming.model Class SmoothStreamingDataTrack

java.lang.Object

└─com.wowza.wms.httpstreamer.smoothstreaming.model.SmoothStreamingDataTrack

public class **SmoothStreamingDataTrack**  
extends Object

### Constructor Summary

public	<a href="#">SmoothStreamingDataTrack</a> (Object lock)
--------	--

### Method Summary

void	<a href="#">addData</a> ( <a href="#">SmoothStreamingDataTrackData</a> data)
void	<a href="#">addQualityLevel</a> ( <a href="#">SmoothStreamingDataTrackQualityLevel</a> qualityLevel)
java.util.Map	<a href="#">getAttributes</a> ()
<a href="#">SmoothStreamingDataTrackData</a>	<a href="#">getDdatas</a> ()
int	<a href="#">getDataSize</a> ()
String	<a href="#">getName</a> ()
String	<a href="#">getParentStreamIndex</a> ()
java.util.List	<a href="#">getQualityLevels</a> ()
String	<a href="#">getSubType</a> ()
long	<a href="#">getTimeScale</a> ()
String	<a href="#">getType</a> ()
String	<a href="#">getUrlStr</a> ()
boolean	<a href="#">isManifestOutput</a> ()
void	<a href="#">removeAttribute</a> (String key)
boolean	<a href="#">removeQualityLevel</a> ( <a href="#">SmoothStreamingDataTrackQualityLevel</a> qualityLevel)

void	<a href="#"><u>setAttribute</u></a> (String key, String value)
void	<a href="#"><u>setManifestOutput</u></a> (boolean manifestOutput)
void	<a href="#"><u>setName</u></a> (String name)
void	<a href="#"><u>setParentStreamIndex</u></a> (String parentStreamIndex)
void	<a href="#"><u>setQualityLevels</u></a> (java.util.List qualityLevels)
void	<a href="#"><u>setSubType</u></a> (String subType)
void	<a href="#"><u>setTimeScale</u></a> (long timeScale)
void	<a href="#"><u>setType</u></a> (String type)
void	<a href="#"><u>setUrlStr</u></a> (String urlStr)
int	<a href="#"><u>trimData</u></a> ()
int	<a href="#"><u>trimData</u></a> (int count)

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructors

### SmoothStreamingDataTrack

```
public SmoothStreamingDataTrack(Object lock)
```

## Methods

### getDatas

```
public SmoothStreamingDataTrackDatas getDdatas()
```

### addData

```
public void addData(SmoothStreamingDataTrackData data)
```

(continued from last page)

## trimData

```
public int trimData()
```

---

## trimData

```
public int trimData(int count)
```

---

## getDataSize

```
public int getDataSize()
```

---

## addQualityLevel

```
public void addQualityLevel(SmoothStreamingDataTrackQualityLevel qualityLevel)
```

---

## removeQualityLevel

```
public boolean removeQualityLevel(SmoothStreamingDataTrackQualityLevel qualityLevel)
```

---

## getQualityLevels

```
public java.util.List getQualityLevels()
```

---

## getType

```
public String getType()
```

---

## setType

```
public void setType(String type)
```

---

## getName

```
public String getName()
```

---

## setName

```
public void setName(String name)
```

---

(continued from last page)

---

### getSubType

```
public String getSubType()
```

---

### setSubType

```
public void setSubType(String subType)
```

---

### getTimeScale

```
public long getTimeScale()
```

---

### setTimeScale

```
public void setTimeScale(long timeScale)
```

---

### getParentStreamIndex

```
public String getParentStreamIndex()
```

---

### setParentStreamIndex

```
public void setParentStreamIndex(String parentStreamIndex)
```

---

### isManifestOutput

```
public boolean isManifestOutput()
```

---

### setManifestOutput

```
public void setManifestOutput(boolean manifestOutput)
```

---

### getUrlStr

```
public String getUrlStr()
```

---

(continued from last page)

## setUrlStr

```
public void setUrlStr(String urlStr)
```

---

## setQualityLevels

```
public void setQualityLevels(java.util.List qualityLevels)
```

---

## setAttribute

```
public void setAttribute(String key,  
                          String value)
```

---

## removeAttribute

```
public void removeAttribute(String key)
```

---

## getAttributes

```
public java.util.Map getAttributes()
```

---

## com.wowza.wms.httpstreamer.smoothstreaming.model Class SmoothStreamingDataTrackData

java.lang.Object

└-com.wowza.wms.httpstreamer.smoothstreaming.model.SmoothStreamingDataTrackData

public class **SmoothStreamingDataTrackData**  
extends Object

### Constructor Summary

public	<a href="#">SmoothStreamingDataTrackData()</a>
--------	--

### Method Summary

String	<a href="#">getDataStr()</a>
long	<a href="#">getTimecode()</a>
boolean	<a href="#">isCDATA()</a>
void	<a href="#">setCDATA(boolean isCDATA)</a>
void	<a href="#">setDataStr(String dataStr)</a>
void	<a href="#">setTimecode(long timecode)</a>

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Constructors

#### SmoothStreamingDataTrackData

public **SmoothStreamingDataTrackData()**

### Methods

#### getTimecode

public long **getTimecode()**

---

### **setTimecode**

```
public void setTimecode(long timecode)
```

---

### **getDataStr**

```
public String getDataStr()
```

---

### **setDataStr**

```
public void setDataStr(String dataStr)
```

---

### **isCDATA**

```
public boolean isCDATA()
```

---

### **setCDATA**

```
public void setCDATA(boolean isCDATA)
```

---



## com.wowza.wms.httpstreamer.smoothstreaming.model Class SmoothStreamingDataTrackDatas

java.lang.Object

└─com.wowza.wms.httpstreamer.smoothstreaming.model.SmoothStreamingDataTrackDatas

public class **SmoothStreamingDataTrackDatas**  
extends Object

### Field Summary

public	<a href="#">datas</a>
public	<a href="#">indexOffset</a>

### Constructor Summary

public	<a href="#">SmoothStreamingDataTrackDatas()</a>
--------	---

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Fields

#### **datas**

public java.util.List **datas**

#### **indexOffset**

public long **indexOffset**

### Constructors

#### **SmoothStreamingDataTrackDatas**

public **SmoothStreamingDataTrackDatas()**

## com.wowza.wms.httpstreamer.smoothstreaming.model Class SmoothStreamingDataTrackQualityLevel

java.lang.Object



com.wowza.wms.httpstreamer.smoothstreaming.model.SmoothStreamingDataTrackQualityLevel

public class **SmoothStreamingDataTrackQualityLevel**  
extends Object

### Constructor Summary

public	<a href="#">SmoothStreamingDataTrackQualityLevel()</a>
--------	--

### Method Summary

java.util.Map	<a href="#">getAttributes()</a>
String	<a href="#">getBitrate()</a>
String	<a href="#">getCodecPrivateData()</a>
String	<a href="#">getFourCC()</a>
void	<a href="#">removeAttribute</a> (String key)
void	<a href="#">setAttribute</a> (String key, String value)
void	<a href="#">setBitrate</a> (String bitrate)
void	<a href="#">setCodecPrivateData</a> (String codecPrivateData)
void	<a href="#">setFourCC</a> (String fourCC)

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Constructors

#### SmoothStreamingDataTrackQualityLevel

public **SmoothStreamingDataTrackQualityLevel**()

## Methods

### **getBitrate**

```
public String getBitrate()
```

### **setBitrate**

```
public void setBitrate(String bitrate)
```

### **getCodecPrivateData**

```
public String getCodecPrivateData()
```

### **setCodecPrivateData**

```
public void setCodecPrivateData(String codecPrivateData)
```

### **getFourCC**

```
public String getFourCC()
```

### **setFourCC**

```
public void setFourCC(String fourCC)
```

### **setAttribute**

```
public void setAttribute(String key,  
    String value)
```

### **removeAttribute**

```
public void removeAttribute(String key)
```

### **getAttributes**

```
public java.util.Map getAttributes()
```

# com.wowza.wms.httpstreamer.smoothstreaming.model

## Class SmoothStreamingDataTracks

java.lang.Object

└─com.wowza.wms.httpstreamer.smoothstreaming.model.SmoothStreamingDataTracks

public class **SmoothStreamingDataTracks**  
extends Object

### Constructor Summary

public	<a href="#">SmoothStreamingDataTracks</a> (Object lock)
--------	---

### Method Summary

<a href="#">SmoothStreamingDataTrack</a>	<a href="#">getDataTrack</a> (String trackId)
int	<a href="#">getDataTrackCount</a> ()
java.util.List	<a href="#">getDataTracks</a> ()
void	<a href="#">putDataTrack</a> (String trackId, <a href="#">SmoothStreamingDataTrack</a> dataTrack)
<a href="#">SmoothStreamingDataTrack</a>	<a href="#">removeDataTrack</a> (String trackId)

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Constructors

**SmoothStreamingDataTracks**

public **SmoothStreamingDataTracks**(Object lock)

### Methods

**putDataTrack**

public void **putDataTrack**(String trackId, [SmoothStreamingDataTrack](#) dataTrack)

## getDataTrack

```
public SmoothStreamingDataTrack getDataTrack(String trackId)
```

---

## getDataTracks

```
public java.util.List getDataTracks()
```

---

## removeDataTrack

```
public SmoothStreamingDataTrack removeDataTrack(String trackId)
```

---

## getDataTrackCount

```
public int getDataTrackCount()
```

---

Package

**com.wowza.wms.httpstreamer.util**

## com.wowza.wms.httpstreamer.util Class HTTPStreamerUtils

java.lang.Object

└─com.wowza.wms.httpstreamer.util.HTTPStreamerUtils

public class **HTTPStreamerUtils**  
extends Object

### Field Summary

public static final	<a href="#">KEYFILEEXTENSION</a> Value: <b>.key</b>
---------------------	--

### Constructor Summary

public	<a href="#">HTTPStreamerUtils()</a>
--------	-------------------------------------

### Method Summary

static byte[]	<a href="#">keyFileStrToBytes</a> (String inStr)
---------------	--

static java.util.Map	<a href="#">parseEncryptionKeyFile</a> (java.io.File encFile)
----------------------	---

static <a href="#">IHTTPStreamerAdapter</a>	<a href="#">pathToAdapter</a> ( <a href="#">IVHost</a> vhost, <a href="#">HostPort</a> hostPort, String path)
--	---

static String	<a href="#">urlDecode</a> (String url)
---------------	--

static String	<a href="#">urlEncode</a> (String url)
---------------	--

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Fields

#### KEYFILEEXTENSION

public static final java.lang.String **KEYFILEEXTENSION**

Constant value: **.key**

### Constructors

(continued from last page)

## HTTPStreamerUtils

```
public HTTPStreamerUtils()
```

## Methods

### keyFileStrToBytes

```
public static byte[] keyFileStrToBytes(String inStr)
```

---

### parseEncryptionKeyFile

```
public static java.util.Map parseEncryptionKeyFile(java.io.File encFile)
```

---

### pathToAdapter

```
public static IHTTPStreamerAdapter pathToAdapter(IVHost vhost,  
    HostPort hostPort,  
    String path)
```

---

### urlDecode

```
public static String urlDecode(String url)
```

---

### urlEncode

```
public static String urlEncode(String url)
```



---

Package

**com.wowza.wms.logging**

## com.wowza.wms.logging Interface Constants

public interface **Constants**  
extends

Constants used internally throughout log4j.

### Field Summary

public static final	<a href="#"><u>ABSOLUTE_FORMAT</u></a> ABSOLUTE string literal. Value: <b>ABSOLUTE</b>
public static final	<a href="#"><u>ABSOLUTE_TIME_PATTERN</u></a> SimpleTimePattern for ABSOLUTE. Value: <b>HH:mm:ss,SSS</b>
public static final	<a href="#"><u>APPLICATION_KEY</u></a> application string literal. Value: <b>application</b>
public static final	<a href="#"><u>CODES_HREF</u></a> Codes URL string literal. Value: <b>http://logging.apache.org/log4j/docs/codes.html</b>
public static final	<a href="#"><u>CONFIGURATOR_CLASS_KEY</u></a> log4j.configuratorClass string literal. Value: <b>log4j.configuratorClass</b>
public static final	<a href="#"><u>DATE_AND_TIME_FORMAT</u></a> DATE string literal. Value: <b>DATE</b>
public static final	<a href="#"><u>DATE_AND_TIME_PATTERN</u></a> SimpleTimePattern for DATE. Value: <b>dd MMM yyyy HH:mm:ss,SSS</b>
public static final	<a href="#"><u>DEFAULT_CONFIGURATION_FILE</u></a> The default property file name for automatic configuration. Value: <b>log4j.properties</b>
public static final	<a href="#"><u>DEFAULT_CONFIGURATION_KEY</u></a> log4j.configuration string literal. Value: <b>log4j.configuration</b>
public static final	<a href="#"><u>DEFAULT_REPOSITORY_NAME</u></a> The name of the default repository is "default" (without the quotes). Value: <b>default</b>
public static final	<a href="#"><u>DEFAULT_XML_CONFIGURATION_FILE</u></a> The default XML configuration file name for automatic configuration. Value: <b>log4j.xml</b>

public static final	<a href="#">HOSTNAME_KEY</a> hostname string literal. Value: <b>hostname</b>
public static final	<a href="#">ISO8601_FORMAT</a> ISO8601 string literal. Value: <b>ISO8601</b>
public static final	<a href="#">ISO8601_PATTERN</a> SimpleTimePattern for ISO8601. Value: <b>yyyy-MM-dd HH:mm:ss,SSS</b>
public static final	<a href="#">JNDI_CONTEXT_NAME</a> JNDI context name string literal. Value: <b>java:comp/env/log4j/context-name</b>
public static final	<a href="#">LOG4J_ID_KEY</a> log4jid string literal. Value: <b>log4jid</b>
public static final	<a href="#">LOG4J_PACKAGE_NAME</a> log4j package name string literal. Value: <b>org.apache.log4j</b>
public static final	<a href="#">RECEIVER_NAME_KEY</a> receiver string literal. Value: <b>receiver</b>
public static final	<a href="#">TEMP_CONSOLE_APPENDER_NAME</a> TEMP_CONSOLE_APPENDER string literal. Value: <b>TEMP_CONSOLE_APPENDER</b>
public static final	<a href="#">TEMP_LIST_APPENDER_NAME</a> TEMP_LIST_APPENDER string literal. Value: <b>TEMP_LIST_APPENDER</b>
public static final	<a href="#">TIMESTAMP_RULE_FORMAT</a> time stamp pattern string literal. Value: <b>yyyy/MM/dd HH:mm:ss</b>

## Fields

### LOG4J\_PACKAGE\_NAME

public static final java.lang.String **LOG4J\_PACKAGE\_NAME**

log4j package name string literal.  
Constant value: **org.apache.log4j**

### DEFAULT\_REPOSITORY\_NAME

public static final java.lang.String **DEFAULT\_REPOSITORY\_NAME**

The name of the default repository is "default" (without the quotes).  
Constant value: **default**

## APPLICATION\_KEY

```
public static final java.lang.String APPLICATION_KEY
```

application string literal.

Constant value: **application**

---

## HOSTNAME\_KEY

```
public static final java.lang.String HOSTNAME_KEY
```

hostname string literal.

Constant value: **hostname**

---

## RECEIVER\_NAME\_KEY

```
public static final java.lang.String RECEIVER_NAME_KEY
```

receiver string literal.

Constant value: **receiver**

---

## LOG4J\_ID\_KEY

```
public static final java.lang.String LOG4J_ID_KEY
```

log4jid string literal.

Constant value: **log4jid**

---

## TIMESTAMP\_RULE\_FORMAT

```
public static final java.lang.String TIMESTAMP_RULE_FORMAT
```

time stamp pattern string literal.

Constant value: **yyyy/MM/dd HH:mm:ss**

---

## DEFAULT\_CONFIGURATION\_FILE

```
public static final java.lang.String DEFAULT_CONFIGURATION_FILE
```

The default property file name for automatic configuration.

Constant value: **log4j.properties**

---

## DEFAULT\_XML\_CONFIGURATION\_FILE

```
public static final java.lang.String DEFAULT_XML_CONFIGURATION_FILE
```

The default XML configuration file name for automatic configuration.

Constant value: **log4j.xml**

---

## DEFAULT\_CONFIGURATION\_KEY

```
public static final java.lang.String DEFAULT_CONFIGURATION_KEY
```

log4j.configuration string literal.

Constant value: **log4j.configuration**

---

(continued from last page)

---

## CONFIGURATOR\_CLASS\_KEY

```
public static final java.lang.String CONFIGURATOR_CLASS_KEY
```

log4j.configuratorClass string literal.

Constant value: **log4j.configuratorClass**

---

## JNDI\_CONTEXT\_NAME

```
public static final java.lang.String JNDI_CONTEXT_NAME
```

JNDI context name string literal.

Constant value: **java:comp/env/log4j/context-name**

---

## TEMP\_LIST\_APPENDER\_NAME

```
public static final java.lang.String TEMP_LIST_APPENDER_NAME
```

TEMP\_LIST\_APPENDER string literal.

Constant value: **TEMP\_LIST\_APPENDER**

---

## TEMP\_CONSOLE\_APPENDER\_NAME

```
public static final java.lang.String TEMP_CONSOLE_APPENDER_NAME
```

TEMP\_CONSOLE\_APPENDER string literal.

Constant value: **TEMP\_CONSOLE\_APPENDER**

---

## CODES\_HREF

```
public static final java.lang.String CODES_HREF
```

Codes URL string literal.

Constant value: **http://logging.apache.org/log4j/docs/codes.html**

---

## ABSOLUTE\_FORMAT

```
public static final java.lang.String ABSOLUTE_FORMAT
```

ABSOLUTE string literal.

Constant value: **ABSOLUTE**

---

## ABSOLUTE\_TIME\_PATTERN

```
public static final java.lang.String ABSOLUTE_TIME_PATTERN
```

SimpleTimePattern for ABSOLUTE.

Constant value: **HH:mm:ss,SSS**

---

## DATE\_AND\_TIME\_FORMAT

```
public static final java.lang.String DATE_AND_TIME_FORMAT
```

DATE string literal.

Constant value: **DATE**

---

## DATE\_AND\_TIME\_PATTERN

```
public static final java.lang.String DATE_AND_TIME_PATTERN
```

---

(continued from last page)

SimpleTimePattern for DATE.

Constant value: **dd MMM yyyy HH:mm:ss,SSS**

---

## ISO8601\_FORMAT

```
public static final java.lang.String ISO8601_FORMAT
```

ISO8601 string literal.

Constant value: **ISO8601**

---

## ISO8601\_PATTERN

```
public static final java.lang.String ISO8601_PATTERN
```

SimpleTimePattern for ISO8601.

Constant value: **yyyy-MM-dd HH:mm:ss,SSS**

## com.wowza.wms.logging Interface ILogNotify

All Known Implementing Classes:

[LogNotifyCalculateIncremental](#)

---

public interface **ILogNotify**  
extends

ILogNotify: Interface to add custom fields to the Wowza Pro log files. To add your own custom log fields, define a class that implements this interface. The onLog method will be called each time the Wowza Pro server logs a message. Here is an example of a simple ILogNotify class that logs the current system time in milliseconds as a Long (systime-long) and as a String (systime-string).

```
package com.wowza.wms.plugin.newlogfields;

import com.wowza.wms.logging.*;

public class NewLogFields 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", new String(sc_bytes));
    }
}
```

*Note: To get any of the values currently being logged use the logging API  
WMSLoggerFactory.getGlobalLogValue(WMSLoggerIDs.FD\_\*)*

To add your class to Wowza Pro, compile your class into a .class file, bind the class into a .jar file and copy the .jar file into the Wowza Pro server /lib folder. Next, edit:

- [install-dir]/bin/setenv.bat (Windows)
- [install-dir]/bin/setenv.sh (Linux, OSX, Solaris)

Add **-Dcom.wowza.wms.logging.LogNotify=[full-path-to-your-ILogNotify-class]** to the **JAVA\_OPTS**. For example for the class above the JAVA\_OPTS would look like:

```
Linux/OSX
JAVA_OPTS="-Xmx768M -
Dcom.wowza.wms.logging.LogNotify=com.wowza.wms.plugin.newlogfields.NewLogFields"

Windows
JAVA_OPTS=-Xmx768M -
Dcom.wowza.wms.logging.LogNotify=com.wowza.wms.plugin.newlogfields.NewLogFields
```

If on Windows, also edit `[install-dir]/bin/WowzaMediaServerPro-Service.conf` and add `wrapper.java.additional.[n]=-Dcom.wowza.wms.logging.LogNotify=[full-path-to-your-ILogNotify-class]` to the **Java Additional Parameters** section where `[n]` is the next number in the list of active parameters. For example for the class above if the last active additional parameter is 6, the entry would look like this:

```
wrapper.java.additional.7=-
Dcom.wowza.wms.logging.LogNotify=com.wowza.wms.plugin.newlogfields.NewLogFields
```

Next, edit `[install-dir]/conf/log4j.properties` and add the new field names to any `log4j.appender.[appender-name].layout.Fields` fields lists to which you want to log these values. For example:

```
log4j.appender.stdout.layout.Fields=x-severity,x-category,x-event,x-ctx,x-comment,sysime-
long,sysime-string
```

## Method Summary

void	<code>onLog(org.apache.log4j.Level level, String comment, <a href="#">IMediaStream</a> stream, String category, String event, int status, String context)</code> Called each time the server logs a message.
------	---

## Methods

### onLog

```
public void onLog(org.apache.log4j.Level level,
    String comment,
    IMediaStream stream,
    String category,
    String event,
    int status,
    String context)
```

Called each time the server logs a message.

#### Parameters:

- level - log level as defined by (org.apache.log4j.Level)
- comment - comment part of the log statement
- stream - if stream category log message it's the source stream
- category - log category as defined by WMSLoggerIDs.CAT\_\*
- event - log event as defined by WMSLoggerIDs.EVT\_\*
- status - log status (same as HTTP status field) as defined by WMSLoggerIDs.STAT\_\*
- context - log context value like stream name, vhost name, application name



## com.wowza.wms.logging Class LogNotifyCalculateIncremental

java.lang.Object

└─com.wowza.wms.logging.LogNotifyCalculateIncremental

All Implemented Interfaces:

[ILogNotify](#)

public class **LogNotifyCalculateIncremental**  
extends Object  
implements [ILogNotify](#)

LogNotifyCalculateIncremental: pre-built implementation of ILogNotify that adds 5 new log fields:

- cs-bytes-inc (client to server bytes streamed since last log entry)
- sc-bytes-inc (server to client bytesstreamed since last log entry)
- cs-stream-bytes-inc (client to server stream bytesstreamed since last log entry)
- sc-stream-bytes-inc (server to client stream bytesstreamed since last log entry)
- x-duration-inc (time in seconds that have passed since the last log event)

These are incremental versions of the regular log values. To add this to Wowza Pro edit:

- [install-dir]/bin/setenv.bat (Windows)
- [install-dir]/bin/setenv.sh (Linux, OSX)

Add `-Dcom.wowza.wms.logging.LogNotify=com.wowza.wms.logging.LogNotifyCalculateIncremental` to the `JAVA_OPTS`. Also, edit `[install-dir]/conf/log4j.properties` and add these field names to any `log4j.appender.[appender-name].layout.Fields` fields lists to which you want to log these values.

*NOTE: These new log values are only accurate if all the events are included for the **session** and **stream** log categories.*

### Field Summary

public static final	<a href="#">FD_cs_bytes_inc</a> Value: <b>cs-bytes-inc</b>
public static final	<a href="#">FD_cs_stream_bytes_inc</a> Value: <b>cs-stream-bytes-inc</b>
public static final	<a href="#">FD_sc_bytes_inc</a> Value: <b>sc-bytes-inc</b>
public static final	<a href="#">FD_sc_stream_bytes_inc</a> Value: <b>sc-stream-bytes-inc</b>
public static final	<a href="#">FD_x_duration_inc</a> Value: <b>x-duration-inc</b>

### Constructor Summary

public	<a href="#">LogNotifyCalculateIncremental()</a>
--------	---

## Method Summary

void	<a href="#">onLog</a> (org.apache.log4j.Level level, String comment, <a href="#">IMediaStream</a> stream, String category, String event, int status, String context)
------	--

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Methods inherited from interface [com.wowza.wms.logging.ILogNotify](#)

[onLog](#)

## Fields

### FD\_cs\_bytes\_inc

public static final java.lang.String **FD\_cs\_bytes\_inc**

Constant value: **cs-bytes-inc**

### FD\_sc\_bytes\_inc

public static final java.lang.String **FD\_sc\_bytes\_inc**

Constant value: **sc-bytes-inc**

### FD\_cs\_stream\_bytes\_inc

public static final java.lang.String **FD\_cs\_stream\_bytes\_inc**

Constant value: **cs-stream-bytes-inc**

### FD\_sc\_stream\_bytes\_inc

public static final java.lang.String **FD\_sc\_stream\_bytes\_inc**

Constant value: **sc-stream-bytes-inc**

### FD\_x\_duration\_inc

public static final java.lang.String **FD\_x\_duration\_inc**

Constant value: **x-duration-inc**

## Constructors

(continued from last page)

## LogNotifyCalculateIncremental

```
public LogNotifyCalculateIncremental()
```

## Methods

### onLog

```
public void onLog(org.apache.log4j.Level level,  
    String comment,  
    IMediaStream stream,  
    String category,  
    String event,  
    int status,  
    String context)
```

## com.wowza.wms.logging Interface NetworkBased

All Subinterfaces:  
[PortBased](#)

public interface **NetworkBased**  
extends

The parent of all the Network based interfaces.

### Method Summary

String	<a href="#">getName()</a> Get name.
boolean	<a href="#">isActive()</a> Get if item is active.

### Methods

#### getName

public String **getName()**

Get name.

**Returns:**  
name.

#### isActive

public boolean **isActive()**

Get if item is active.

**Returns:**  
if true, item is active.

## com.wowza.wms.logging Interface PortBased

All Superinterfaces:

[NetworkBased](#)

All Known Implementing Classes:

[UDPAppender](#)

---

public interface **PortBased**  
extends [NetworkBased](#)

Net based entities that 'work with' a Port should consider implementing this interface so that they can be treated generically.

---

### Method Summary

int	<a href="#">getPort()</a> Returns the Port # that this net based thing is using.
-----	---

Methods inherited from interface [com.wowza.wms.logging.NetworkBased](#)

[getName](#), [isActive](#)

---

### Methods

#### getPort

public int **getPort()**

Returns the Port # that this net based thing is using.

**Returns:**

int port number

## com.wowza.wms.logging Class UDPAppender

```
java.lang.Object
|
|--org.apache.log4j.AppenderSkeleton
|   |
|   |--com.wowza.wms.logging.UDPAppender
```

### All Implemented Interfaces:

[PortBased](#), org.apache.log4j.spi.OptionHandler, org.apache.log4j.Appender

```
public class UDPAppender
extends org.apache.log4j.AppenderSkeleton
implements org.apache.log4j.Appender, org.apache.log4j.spi.OptionHandler, PortBased
```

Sends log information as a UDP datagrams.

The UDPAppender is meant to be used as a diagnostic logging tool so that logging can be monitored by a simple UDP client.

Messages are not sent as LoggingEvent objects but as text after applying the designated Layout.

The port and remoteHost properties can be set in configuration properties. By setting the remoteHost to a broadcast address any number of clients can listen for log messages.

This was inspired and really extended/copied from SocketAppender. Please see the docs for the proper credit to the authors of that class.

## Field Summary

public static final	<a href="#">DEFAULT_PORT</a> The default port number for the UDP packets, 9991. Value: <b>9991</b>
---------------------	--

## Fields inherited from class org.apache.log4j.AppenderSkeleton

closed, errorHandler, headFilter, layout, name, tailFilter, threshold

## Constructor Summary

public	<a href="#">UDPAppender</a> ()
public	<a href="#">UDPAppender</a> (java.net.InetAddress address, int port) Sends UDP packets to the address and port.
public	<a href="#">UDPAppender</a> (String host, int port) Sends UDP packets to the address and port.

## Method Summary

void	<a href="#">activateOptions</a> () Open the UDP sender for the <b>RemoteHost</b> and <b>Port</b> .
void	<a href="#">append</a> (org.apache.log4j.spi.LoggingEvent event)

void	<a href="#"><code>cleanup()</code></a> Close the UDP Socket and release the underlying connector thread if it has been created
void	<a href="#"><code>close()</code></a> Close this appender.
String	<a href="#"><code>getApplication()</code></a> Returns value of the <b>App</b> option.
String	<a href="#"><code>getEncoding()</code></a> Returns value of the <b>Encoding</b> option.
int	<a href="#"><code>getPort()</code></a> Returns value of the <b>Port</b> option.
String	<a href="#"><code>getRemoteHost()</code></a> Returns value of the <b>RemoteHost</b> option.
boolean	<a href="#"><code>isActive()</code></a>
boolean	<a href="#"><code>requiresLayout()</code></a> The UDPAppender uses layouts.
void	<a href="#"><code>setApplication(String app)</code></a> The <b>App</b> option takes a string value which should be the name of the application getting logged.
void	<a href="#"><code>setEncoding(String encoding)</code></a> The <b>Encoding</b> option specifies how the bytes are encoded.
void	<a href="#"><code>setPort(int port)</code></a> The <b>Port</b> option takes a positive integer representing the port where UDP packets will be sent.
void	<a href="#"><code>setRemoteHost(String host)</code></a> The <b>RemoteHost</b> option takes a string value which should be the host name or ipaddress to send the UDP packets.

**Methods inherited from class org.apache.log4j.AppenderSkeleton**

activateOptions, addFilter, append, clearFilters, doAppend, finalize, getErrorHandler, getFilter, getFirstFilter, getLayout, getName, getThreshold, isAsSevereAsThreshold, setErrorHandler, setLayout, setName, setThreshold

**Methods inherited from class java.lang.Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

**Methods inherited from interface org.apache.log4j.Appender**

addFilter, clearFilters, close, doAppend, getErrorHandler, getFilter, getLayout, getName, requiresLayout, setErrorHandler, setLayout, setName

**Methods inherited from interface org.apache.log4j.spi.OptionHandler**

activateOptions

**Methods inherited from interface [com.wowza.wms.logging.PortBased](#)**

[getPort](#)

Methods inherited from interface [com.wowza.wms.logging.NetworkBased](#)

[getName](#), [isActive](#)

## Fields

### DEFAULT\_PORT

```
public static final int DEFAULT_PORT
```

The default port number for the UDP packets, 9991.  
Constant value: **9991**

## Constructors

### UDPAppender

```
public UDPAppender()
```

### UDPAppender

```
public UDPAppender(java.net.InetAddress address,  
                    int port)
```

Sends UDP packets to the address and port.

### UDPAppender

```
public UDPAppender(String host,  
                    int port)
```

Sends UDP packets to the address and port.

## Methods

### activateOptions

```
public void activateOptions()
```

Open the UDP sender for the **RemoteHost** and **Port**.

### close

```
public void close()
```

Close this appender.

This will mark the appender as closed and call then [cleanup\(\)](#) method.



(continued from last page)

## cleanUp

```
public void cleanUp()
```

Close the UDP Socket and release the underlying connector thread if it has been created

---

## append

```
public void append(org.apache.log4j.spi.LoggingEvent event)
```

---

## isActive

```
public boolean isActive()
```

---

## requiresLayout

```
public boolean requiresLayout()
```

The **UDPAppender** uses layouts. Hence, this method returns true.

---

## setRemoteHost

```
public void setRemoteHost(String host)
```

The **RemoteHost** option takes a string value which should be the host name or ipaddress to send the UDP packets.

---

## getRemoteHost

```
public String getRemoteHost()
```

Returns value of the **RemoteHost** option.

---

## setApplication

```
public void setApplication(String app)
```

The **App** option takes a string value which should be the name of the application getting logged. If property was already set (via system property), don't set here.

---

## getApplication

```
public String getApplication()
```

Returns value of the **App** option.

---

## setEncoding

```
public void setEncoding(String encoding)
```

The **Encoding** option specifies how the bytes are encoded. If this option is not specified, the System encoding is used.

---

## getEncoding

```
public String getEncoding()
```

---

(continued from last page)

Returns value of the **Encoding** option.

---

## setPort

```
public void setPort(int port)
```

The **Port** option takes a positive integer representing the port where UDP packets will be sent.

---

## getPort

```
public int getPort()
```

Returns value of the **Port** option.

## com.wowza.wms.logging Class WMSLogger

```

java.lang.Object
  |
  +-org.apache.log4j.Category
    |
    +-org.apache.log4j.Logger
      |
      +-com.wowza.wms.logging.WMSLogger
  
```

### All Implemented Interfaces:

org.apache.log4j.spi.AppenderAttachable

```

public class WMSLogger
extends org.apache.log4j.Logger
  
```

### Fields inherited from class org.apache.log4j.Category

additive, level, name, parent, repository, resourceBundle

## Constructor Summary

public	<a href="#">WMSLogger</a> (String name, org.apache.log4j.Logger tmpLogger)
public	<a href="#">WMSLogger</a> (String name)

## Method Summary

void	<a href="#">debug</a> (String comment)
void	<a href="#">debug</a> (String comment, <a href="#">IMediaStream</a> stream)
void	<a href="#">debug</a> (String comment, <a href="#">IMediaStream</a> stream, String category, String event, int status, String context)
void	<a href="#">debug</a> (String comment, String category, String event)
void	<a href="#">debug</a> (String comment, String category, String event, int status, String context)
void	<a href="#">error</a> (String comment)
void	<a href="#">error</a> (String comment, <a href="#">IMediaStream</a> stream)
void	<a href="#">error</a> (String comment, <a href="#">IMediaStream</a> stream, String category, String event, int status, String context)
void	<a href="#">error</a> (String comment, String category, String event)

void	<a href="#">error</a> (String comment, String category, String event, int status, String context)
void	<a href="#">fatal</a> (String comment)
void	<a href="#">fatal</a> (String comment, <a href="#">IMediaStream</a> stream)
void	<a href="#">fatal</a> (String comment, <a href="#">IMediaStream</a> stream, String category, String event, int status, String context)
void	<a href="#">fatal</a> (String comment, String category, String event)
void	<a href="#">fatal</a> (String comment, String category, String event, int status, String context)
static <a href="#">WMSLogger</a>	<a href="#">getLogger</a> (String name)
void	<a href="#">info</a> (String comment)
void	<a href="#">info</a> (String comment, <a href="#">IMediaStream</a> stream)
void	<a href="#">info</a> (String comment, <a href="#">IMediaStream</a> stream, String category, String event, int status, String context)
void	<a href="#">info</a> (String comment, String category, String event)
void	<a href="#">info</a> (String comment, String category, String event, int status, String context)
boolean	<a href="#">isDebugEnabled</a> ()
boolean	<a href="#">isEnabledFor</a> (org.apache.log4j.Priority level)
boolean	<a href="#">isErrorEnabled</a> ()
boolean	<a href="#">isInfoEnabled</a> ()
boolean	<a href="#">isTraceEnabled</a> ()
boolean	<a href="#">isWarnEnabled</a> ()
void	<a href="#">log</a> (org.apache.log4j.Level level, String comment)
void	<a href="#">log</a> (org.apache.log4j.Level level, String comment, <a href="#">IMediaStream</a> stream)
void	<a href="#">log</a> (org.apache.log4j.Level level, String comment, <a href="#">IMediaStream</a> stream, String category, String event)
void	<a href="#">log</a> (org.apache.log4j.Level level, String comment, <a href="#">IMediaStream</a> stream, String category, String event, int status, String context)

void	<a href="#">log</a> (org.apache.log4j.Level level, String comment, String category, String event)
void	<a href="#">warn</a> (String comment)
void	<a href="#">warn</a> (String comment, <a href="#">IMediaStream</a> stream)
void	<a href="#">warn</a> (String comment, <a href="#">IMediaStream</a> stream, String category, String event, int status, String context)
void	<a href="#">warn</a> (String comment, String category, String event)
void	<a href="#">warn</a> (String comment, String category, String event, int status, String context)

#### Methods inherited from class org.apache.log4j.Logger

getLogger, getLogger, getLogger, getRootLogger, isDebugEnabled, trace, trace

#### Methods inherited from class org.apache.log4j.Category

addAppender, assertLog, callAppenders, debug, debug, error, error, exists, fatal, fatal, forcedLog, getAdditivity, getAllAppenders, getAppender, getChainedPriority, getCurrentCategories, getDefaultHierarchy, getEffectiveLevel, getHierarchy, getInstance, getInstance, getLevel, getLoggerRepository, getName, getParent, getPriority, getResourceBundle, getResourceBundleString, getRoot, info, info, isAttached, isDebugEnabled, isEnabledFor, isInfoEnabled, l7dlog, l7dlog, log, log, log, removeAllAppenders, removeAppender, removeAppender, setAdditivity, setLevel, setPriority, setResourceBundle, shutdown, warn, warn

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

#### Methods inherited from interface org.apache.log4j.spi.AppenderAttachable

addAppender, getAllAppenders, getAppender, isAttached, removeAllAppenders, removeAppender, removeAppender

## Constructors

### WMSLogger

```
public WMSLogger(String name,
                 org.apache.log4j.Logger tmpLogger)
```

### WMSLogger

```
public WMSLogger(String name)
```

## Methods

### getLogger

```
public static WMSLogger getLogger(String name)
```

---

### isWarnEnabled

```
public boolean isWarnEnabled()
```

---

### isErrorEnabled

```
public boolean isErrorEnabled()
```

---

### isTraceEnabled

```
public boolean isTraceEnabled()
```

---

### isDebugEnabled

```
public boolean isDebugEnabled()
```

---

### isInfoEnabled

```
public boolean isInfoEnabled()
```

---

### isEnabledFor

```
public boolean isEnabledFor(org.apache.log4j.Priority level)
```

---

### log

```
public void log(org.apache.log4j.Level level,  
               String comment,  
               IMediaStream stream,  
               String category,  
               String event,  
               int status,  
               String context)
```

---

(continued from last page)

## log

```
public void log(org.apache.log4j.Level level,
               String comment,
               IMediaStream stream,
               String category,
               String event)
```

---

## log

```
public void log(org.apache.log4j.Level level,
               String comment,
               String category,
               String event)
```

---

## log

```
public void log(org.apache.log4j.Level level,
               String comment)
```

---

## log

```
public void log(org.apache.log4j.Level level,
               String comment,
               IMediaStream stream)
```

---

## info

```
public void info(String comment)
```

---

## info

```
public void info(String comment,
               IMediaStream stream)
```

---

## info

```
public void info(String comment,
               String category,
               String event)
```

---

(continued from last page)

**info**

```
public void info(String comment,
                String category,
                String event,
                int status,
                String context)
```

---

**info**

```
public void info(String comment,
                IMediaStream stream,
                String category,
                String event,
                int status,
                String context)
```

---

**error**

```
public void error(String comment)
```

---

**error**

```
public void error(String comment,
                IMediaStream stream)
```

---

**error**

```
public void error(String comment,
                String category,
                String event)
```

---

**error**

```
public void error(String comment,
                String category,
                String event,
                int status,
                String context)
```

---

**error**

```
public void error(String comment,
                IMediaStream stream,
                String category,
                String event,
                int status,
                String context)
```

---



**fatal**

```
public void fatal(String comment)
```

---

**fatal**

```
public void fatal(String comment,  
    IMediaStream stream)
```

---

**fatal**

```
public void fatal(String comment,  
    String category,  
    String event)
```

---

**fatal**

```
public void fatal(String comment,  
    String category,  
    String event,  
    int status,  
    String context)
```

---

**fatal**

```
public void fatal(String comment,  
    IMediaStream stream,  
    String category,  
    String event,  
    int status,  
    String context)
```

---

**debug**

```
public void debug(String comment)
```

---

**debug**

```
public void debug(String comment,  
    IMediaStream stream)
```

---

(continued from last page)

## debug

```
public void debug(String comment,  
                  String category,  
                  String event)
```

---

## debug

```
public void debug(String comment,  
                  String category,  
                  String event,  
                  int status,  
                  String context)
```

---

## debug

```
public void debug(String comment,  
                  IMediaStream stream,  
                  String category,  
                  String event,  
                  int status,  
                  String context)
```

---

## warn

```
public void warn(String comment)
```

---

## warn

```
public void warn(String comment,  
                  IMediaStream stream)
```

---

## warn

```
public void warn(String comment,  
                  String category,  
                  String event)
```

---

## warn

```
public void warn(String comment,  
                  String category,  
                  String event,  
                  int status,  
                  String context)
```

---

(continued from last page)

**warn**

```
public void warn(String comment,  
    IMediaStream stream,  
    String category,  
    String event,  
    int status,  
    String context)
```

## com.wowza.wms.logging Class WMSLoggerFactory

java.lang.Object

└-com.wowza.wms.logging.WMSLoggerFactory

### All Implemented Interfaces:

org.apache.log4j.spi.LoggerFactory

```
public class WMSLoggerFactory
  extends Object
  implements org.apache.log4j.spi.LoggerFactory
```

### Field Summary

public static final	<a href="#">LOGGERNAME_SERVER</a>
	Value: <b>_Server_</b>

### Constructor Summary

public	<a href="#">WMSLoggerFactory()</a>
--------	------------------------------------

### Method Summary

<a href="#">WMSLogger</a>	<a href="#">forceNewLoggerInstance</a> (String name, org.apache.log4j.Logger tmpLogger)
---------------------------	---

static Object	<a href="#">getGlobalLogValue</a> (String key)
---------------	--

static <a href="#">WMSLoggerFactory</a>	<a href="#">getInstance</a> ()
--	--------------------------------

static <a href="#">WMSLogger</a>	<a href="#">getLogger</a> (Class classObj)
----------------------------------	--

static <a href="#">WMSLogger</a>	<a href="#">getLoggerObj</a> ( <a href="#">IApplication</a> application)
----------------------------------	--

static <a href="#">WMSLogger</a>	<a href="#">getLoggerObj</a> ( <a href="#">IApplicationInstance</a> appInstance)
----------------------------------	--

static <a href="#">WMSLogger</a>	<a href="#">getLoggerObj</a> ( <a href="#">IVHost</a> vhost)
----------------------------------	--

<a href="#">WMSLogger</a>	<a href="#">getLoggerObj</a> (String name)
---------------------------	--

static void	<a href="#">initContextLogging</a> ( <a href="#">IApplication</a> application)
-------------	--

static void	<a href="#">initContextLogging</a> ( <a href="#">IApplicationInstance</a> appInstance)
-------------	--

static void	<a href="#">initContextLogging</a> ( <a href="#">IVHost</a> vhost)
-------------	--

static <a href="#">WMSLogger</a>	<a href="#">initializeLogging</a> (String loggingConfigURL)
static <a href="#">WMSLogger</a>	<a href="#">initializeLogging</a> (String loggingConfigURL, <a href="#">IVHost</a> vhost)
static boolean	<a href="#">isGlobalLogValueSet</a> (String key)
org.apache.log4j.Logger	<a href="#">makeNewLoggerInstance</a> (String name)
static void	<a href="#">putGlobalLogValue</a> (String key, Object obj)
static void	<a href="#">removeGlobalLogValue</a> (String key)
static void	<a href="#">removeGlobalLogValues</a> ()
static void	<a href="#">resetMDC</a> ()

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

#### Methods inherited from interface org.apache.log4j.spi.LoggerFactory

makeNewLoggerInstance

## Fields

### LOGGERNAME\_SERVER

```
public static final java.lang.String LOGGERNAME_SERVER
```

Constant value: **\_Server\_**

## Constructors

### WMSLoggerFactory

```
public WMSLoggerFactory()
```

## Methods

### getLoggerObj

```
public WMSLogger getLoggerObj(String name)
```

(continued from last page)

## forceNewLoggerInstance

```
public WMSLogger forceNewLoggerInstance(String name,  
    org.apache.log4j.Logger tmpLogger)
```

---

## getInstance

```
public static WMSLoggerFactory getInstance()
```

---

## initContextLogging

```
public static void initContextLogging(IVHost vhost)
```

---

## initContextLogging

```
public static void initContextLogging(IApplication application)
```

---

## initContextLogging

```
public static void initContextLogging(IApplicationInstance appInstance)
```

---

## initializeLogging

```
public static WMSLogger initializeLogging(String loggingConfigURL)
```

---

## initializeLogging

```
public static WMSLogger initializeLogging(String loggingConfigURL,  
    IVHost vhost)
```

---

## makeNewLoggerInstance

```
public org.apache.log4j.Logger makeNewLoggerInstance(String name)
```

---

## resetMDC

```
public static void resetMDC()
```

---

(continued from last page)

## removeGlobalLogValues

```
public static void removeGlobalLogValues()
```

---

## removeGlobalLogValue

```
public static void removeGlobalLogValue(String key)
```

---

## isGlobalLogValueSet

```
public static boolean isGlobalLogValueSet(String key)
```

---

## putGlobalLogValue

```
public static void putGlobalLogValue(String key,  
                                     Object obj)
```

---

## getGlobalLogValue

```
public static Object getGlobalLogValue(String key)
```

---

## getLoggerObj

```
public static WMSLogger getLoggerObj(IVHost vhost)
```

---

## getLoggerObj

```
public static WMSLogger getLoggerObj(IApplication application)
```

---

## getLoggerObj

```
public static WMSLogger getLoggerObj(IApplicationInstance appInstance)
```

---

## getLogger

```
public static WMSLogger getLogger(Class classObj)
```

---

## com.wowza.wms.logging Class WMSLoggerIDs

java.lang.Object

└-com.wowza.wms.logging.WMSLoggerIDs

public class **WMSLoggerIDs**  
extends Object

### Field Summary

public static final	<a href="#">CAT_ALL</a>
public static final	<a href="#">CAT_application</a> Value: <b>application</b>
public static final	<a href="#">CAT_cupertino</a> Value: <b>cupertino</b>
public static final	<a href="#">CAT_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_vhost</a> Value: <b>vhost</b>
public static final	<a href="#">CAT_webm</a> Value: <b>webm</b>
public static final	<a href="#">CTRL_playlist_node</a> Value: <b>CTRL-playlist-node</b>



public static final	<a href="#">EVT_ALL</a>
public static final	<a href="#">EVT_announce</a> Value: <b>announce</b>
public static final	<a href="#">EVT_app_start</a> Value: <b>app-start</b>
public static final	<a href="#">EVT_app_stop</a> Value: <b>app-stop</b>
public static final	<a href="#">EVT_comment</a> Value: <b>comment</b>
public static final	<a href="#">EVT_connect</a> Value: <b>connect</b>
public static final	<a href="#">EVT_connect_burst</a> Value: <b>connect-burst</b>
public static final	<a href="#">EVT_connect_pending</a> Value: <b>connect-pending</b>
public static final	<a href="#">EVT_create</a> Value: <b>create</b>
public static final	<a href="#">EVT_describe</a> Value: <b>describe</b>
public static final	<a href="#">EVT_destroy</a> Value: <b>destroy</b>
public static final	<a href="#">EVT_disconnect</a> Value: <b>disconnect</b>
public static final	<a href="#">EVT_pause</a> Value: <b>pause</b>
public static final	<a href="#">EVT_play</a> Value: <b>play</b>
public static final	<a href="#">EVT_publish</a> Value: <b>publish</b>
public static final	<a href="#">EVT_record</a> Value: <b>record</b>

public static final	<a href="#">EVT_recordstop</a> Value: <b>recordstop</b>
public static final	<a href="#">EVT_seek</a> Value: <b>seek</b>
public static final	<a href="#">EVT_server_start</a> Value: <b>server-start</b>
public static final	<a href="#">EVT_server_stop</a> Value: <b>server-stop</b>
public static final	<a href="#">EVT_setbuffertime</a> Value: <b>setbuffertime</b>
public static final	<a href="#">EVT_setstreamtype</a> Value: <b>setstreamtype</b>
public static final	<a href="#">EVT_stop</a> Value: <b>stop</b>
public static final	<a href="#">EVT_unpause</a> Value: <b>unpause</b>
public static final	<a href="#">EVT_unpublish</a> Value: <b>unpublish</b>
public static final	<a href="#">EVT_vhost_start</a> Value: <b>vhost-start</b>
public static final	<a href="#">EVT_vhost_stop</a> Value: <b>vhost-stop</b>
public static final	<a href="#">FD_ALL</a>
public static final	<a href="#">FD_c_client_id</a> Value: <b>c-client-id</b>
public static final	<a href="#">FD_c_ip</a> Value: <b>c-ip</b>
public static final	<a href="#">FD_c_proto</a> Value: <b>c-proto</b>
public static final	<a href="#">FD_c_referrer</a> Value: <b>c-referrer</b>

public static final	<a href="#">FD_c_user_agent</a> Value: <b>c-user-agent</b>
public static final	<a href="#">FD_cs_bytes</a> Value: <b>cs-bytes</b>
public static final	<a href="#">FD_cs_stream_bytes</a> Value: <b>cs-stream-bytes</b>
public static final	<a href="#">FD_cs_uri_query</a> Value: <b>cs-uri-query</b>
public static final	<a href="#">FD_cs_uri_stem</a> Value: <b>cs-uri-stem</b>
public static final	<a href="#">FD_date</a> Value: <b>date</b>
public static final	<a href="#">FD_s_ip</a> Value: <b>s-ip</b>
public static final	<a href="#">FD_s_port</a> Value: <b>s-port</b>
public static final	<a href="#">FD_s_uri</a> Value: <b>s-uri</b>
public static final	<a href="#">FD_sc_bytes</a> Value: <b>sc-bytes</b>
public static final	<a href="#">FD_sc_stream_bytes</a> Value: <b>sc-stream-bytes</b>
public static final	<a href="#">FD_time</a> Value: <b>time</b>
public static final	<a href="#">FD_tz</a> Value: <b>tz</b>
public static final	<a href="#">FD_x_app</a> Value: <b>x-app</b>
public static final	<a href="#">FD_x_appinst</a> Value: <b>x-appinst</b>
public static final	<a href="#">FD_x_category</a> Value: <b>x-category</b>

public static final	<a href="#">FD_x_comment</a> Value: <b>x-comment</b>
public static final	<a href="#">FD_x_ctx</a> Value: <b>x-ctx</b>
public static final	<a href="#">FD_x_ctx_override</a> Value: <b>x-ctx-override</b>
public static final	<a href="#">FD_x_duration</a> Value: <b>x-duration</b>
public static final	<a href="#">FD_x_event</a> Value: <b>x-event</b>
public static final	<a href="#">FD_x_file_ext</a> Value: <b>x-file-ext</b>
public static final	<a href="#">FD_x_file_length</a> Value: <b>x-file-length</b>
public static final	<a href="#">FD_x_file_name</a> Value: <b>x-file-name</b>
public static final	<a href="#">FD_x_file_size</a> Value: <b>x-file-size</b>
public static final	<a href="#">FD_x_severity</a> Value: <b>x-severity</b>
public static final	<a href="#">FD_x_sname</a> Value: <b>x-sname</b>
public static final	<a href="#">FD_x_sname_query</a> Value: <b>x-sname-query</b>
public static final	<a href="#">FD_x_spos</a> Value: <b>x-spos</b>
public static final	<a href="#">FD_x_status</a> Value: <b>x-status</b>
public static final	<a href="#">FD_x_stream_id</a> Value: <b>x-stream-id</b>
public static final	<a href="#">FD_x_suri</a> Value: <b>x-suri</b>

public static final	<a href="#">FD_x_suri_query</a> Value: <b>x-suri-query</b>
public static final	<a href="#">FD_x_suri_stem</a> Value: <b>x-suri-stem</b>
public static final	<a href="#">FD_x_vhost</a> Value: <b>x-vhost</b>
public static final	<a href="#">PROTO_HTTPCUPERTINO</a> Value: <b>http (cupertino)</b>
public static final	<a href="#">PROTO_HTTPSCUPERTINO</a> Value: <b>https (cupertino)</b>
public static final	<a href="#">PROTO_HTTPSMOOTH</a> Value: <b>http (smooth)</b>
public static final	<a href="#">PROTO_HTTPSSMOOTH</a> Value: <b>https (smooth)</b>
public static final	<a href="#">PROTO_HTTPSSTREAMER</a> Value: <b>https (streamer)</b>
public static final	<a href="#">PROTO_HTTPSTREAMER</a> Value: <b>http (streamer)</b>
public static final	<a href="#">PROTO_RTMP</a> Value: <b>rtmp</b>
public static final	<a href="#">PROTO_RTMPE</a> Value: <b>rtmpe</b>
public static final	<a href="#">PROTO_RTMPs</a> Value: <b>rtmps</b>
public static final	<a href="#">PROTO_RTMPt</a> Value: <b>rtmpt (HTTP-1.1)</b>
public static final	<a href="#">PROTO_RTMPTe</a> Value: <b>rtmpte (HTTP-1.1)</b>
public static final	<a href="#">PROTO_RTMPts</a> Value: <b>rtmpts (HTTP-1.1)</b>
public static final	<a href="#">PROTO_RTSP</a> Value: <b>rtsp</b>

public static final	<a href="#"><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, 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**

---



(continued from last page)

---

## 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**

---

## 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
```

(continued from last page)

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
```

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**

---

(continued from last page)

---

## 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**

---

## CAT\_server

```
public static final java.lang.String CAT_server
```

Constant value: **server**

---

## CAT\_vhost

```
public static final java.lang.String CAT_vhost
```

---

(continued from last page)

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
```

Constant value: **smoothstreaming**

---

## CAT\_webm

```
public static final java.lang.String CAT_webm
```

Constant value: **webm**

## 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**

---

(continued from last page)

---

## EVT\_setbuffertime

```
public static final java.lang.String EVT_setbuffertime
```

Constant value: **setbuffertime**

---

## EVT\_create

```
public static final java.lang.String EVT_create
```

Constant value: **create**

---

## EVT\_destroy

```
public static final java.lang.String EVT_destroy
```

Constant value: **destroy**

---

## EVT\_setstreamtype

```
public static final java.lang.String EVT_setstreamtype
```

Constant value: **setstreamtype**

---

## EVT\_unpause

```
public static final java.lang.String EVT_unpause
```

Constant value: **unpause**

---

## EVT\_seek

```
public static final java.lang.String EVT_seek
```

Constant value: **seek**

---

## EVT\_stop

```
public static final java.lang.String EVT_stop
```

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
```

(continued from last page)

---

Constant value: **recordstop**

---

**EVT\_server\_start**

```
public static final java.lang.String EVT_server_start
```

Constant value: **server-start**

---

**EVT\_server\_stop**

```
public static final java.lang.String EVT_server_stop
```

Constant value: **server-stop**

---

**EVT\_vhost\_start**

```
public static final java.lang.String EVT_vhost_start
```

Constant value: **vhost-start**

---

**EVT\_vhost\_stop**

```
public static final java.lang.String EVT_vhost_stop
```

Constant value: **vhost-stop**

---

**EVT\_app\_start**

```
public static final java.lang.String EVT_app_start
```

Constant value: **app-start**

---

**EVT\_app\_stop**

```
public static final java.lang.String EVT_app_stop
```

Constant value: **app-stop**

---

**EVT\_comment**

```
public static final java.lang.String EVT_comment
```

Constant value: **comment**

---

**EVT\_announce**

```
public static final java.lang.String EVT_announce
```

Constant value: **announce**



---

## EVT\_describe

```
public static final java.lang.String EVT_describe
```

Constant value: **describe**

---

## CTRL\_playlist\_node

```
public static final java.lang.String CTRL_playlist_node
```

Constant value: **CTRL-playlist-node**

---

## STAT\_connect\_pending\_wating

```
public static final int STAT_connect_pending_wating
```

Constant value: **100**

---

## STAT\_connect\_successful

```
public static final int STAT_connect_successful
```

Constant value: **200**

---

## STAT\_connect\_application\_not\_available

```
public static final int STAT_connect_application_not_available
```

Constant value: **302**

---

## STAT\_connect\_unknown\_protocol

```
public static final int STAT_connect_unknown_protocol
```

Constant value: **400**

---

## STAT\_connect\_rejected\_by\_application

```
public static final int STAT_connect_rejected_by_application
```

Constant value: **401**

---

## STAT\_connect\_rejected\_by\_module

```
public static final int STAT_connect_rejected_by_module
```

Constant value: **403**

---

(continued from last page)

---

**STAT\_connect\_application\_not\_found**

```
public static final int STAT_connect_application_not_found
```

Constant value: **404**

---

**STAT\_connect\_resource\_limit**

```
public static final int STAT_connect_resource_limit
```

Constant value: **409**

---

**STAT\_connect\_license\_limit**

```
public static final int STAT_connect_license_limit
```

Constant value: **413**

---

**STAT\_connect\_redirect**

```
public static final int STAT_connect_redirect
```

Constant value: **302**

---

**STAT\_connect\_internal\_error**

```
public static final int STAT_connect_internal_error
```

Constant value: **500**

---

**STAT\_connect\_bad\_gateway**

```
public static final int STAT_connect_bad_gateway
```

Constant value: **502**

---

**STAT\_connect\_service\_unavailable**

```
public static final int STAT_connect_service_unavailable
```

Constant value: **503**

---

**STAT\_play\_successful**

```
public static final int STAT_play_successful
```

Constant value: **200**

---

**STAT\_play\_bad\_request**

```
public static final int STAT_play_bad_request
```

---

(continued from last page)

---

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**

---

## 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**

---

(continued from last page)

---

## PROTO\_RTMP

```
public static final java.lang.String PROTO_RTMP
```

Constant value: **rtmp**

---

## PROTO\_RTMP

```
public static final java.lang.String PROTO_RTMP
```

Constant value: **rtmp (HTTP-1.1)**

---

## PROTO\_RTMP

```
public static final java.lang.String PROTO_RTMP
```

Constant value: **rtmps (HTTP-1.1)**

---

## PROTO\_RTMP

```
public static final java.lang.String PROTO_RTMP
```

Constant value: **rtmpe**

---

## PROTO\_RTMP

```
public static final java.lang.String PROTO_RTMP
```

Constant value: **rtmpte (HTTP-1.1)**

---

## PROTO\_RTSP

```
public static final java.lang.String PROTO_RTSP
```

Constant value: **rtsp**

---

## PROTO\_HTTPSTREAMER

```
public static final java.lang.String PROTO_HTTPSTREAMER
```

Constant value: **http (streamer)**

---

## PROTO\_HTTPCUPERTINO

```
public static final java.lang.String PROTO_HTTPCUPERTINO
```

Constant value: **http (cupertino)**

---

## PROTO\_HTTPSMOOTH

```
public static final java.lang.String PROTO_HTTPSMOOTH
```

(continued from last page)

Constant value: **http (smooth)**

---

## 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)**

---

## FD\_ALL

```
public static final java.lang.String FD_ALL
```

---

## CAT\_ALL

```
public static final java.lang.String CAT_ALL
```

---

## EVT\_ALL

```
public static final java.lang.String EVT_ALL
```

---

## Constructors

### WMSLoggerIDs

```
public WMSLoggerIDs()
```

---

Package

**com.wowza.wms.mediacaster**

## com.wowza.wms.mediacaster Interface IMediaCaster

public interface **IMediaCaster**  
extends

### Field Summary

public static final	<a href="#"><u>MEDIACASTERTYPE_LIVEREPEATER</u></a> Value: <b>1</b>
public static final	<a href="#"><u>MEDIACASTERTYPE_RTPLIVE</u></a> Value: <b>3</b>
public static final	<a href="#"><u>MEDIACASTERTYPE_SHOUTCAST</u></a> Value: <b>2</b>
public static final	<a href="#"><u>MEDIACASTERTYPE_UNKNOWN</u></a> Value: <b>0</b>
public static final	<a href="#"><u>STREAMTIMEOUTREASON_GOOD</u></a> Value: <b>100</b>
public static final	<a href="#"><u>STREAMTIMEOUTREASON_MISSING</u></a> Value: <b>101</b>
public static final	<a href="#"><u>STREAMTIMEOUTREASON_NORTSPSESSION</u></a> Value: <b>6</b>
public static final	<a href="#"><u>STREAMTIMEOUTREASON_NOSESSION</u></a> Value: <b>2</b>
public static final	<a href="#"><u>STREAMTIMEOUTREASON_NOSTREAM</u></a> Value: <b>4</b>
public static final	<a href="#"><u>STREAMTIMEOUTREASON_NOTIMEOUT</u></a> Value: <b>1</b>
public static final	<a href="#"><u>STREAMTIMEOUTREASON_NOURL</u></a> Value: <b>3</b>
public static final	<a href="#"><u>STREAMTIMEOUTREASON_RECONNECTRUNNING</u></a> Value: <b>5</b>
public static final	<a href="#"><u>STREAMTIMEOUTREASON_UNKNOWN</u></a> Value: <b>0</b>



## Method Summary

boolean	<a href="#"><u>doWatchdog</u></a> ( ) Idle processor
void	<a href="#"><u>forceReset</u></a> ( ) Force a reset/reconnect of this media caster
<a href="#"><u>IApplicationInstance</u></a>	<a href="#"><u>getAppInstance</u></a> ( ) Get the application instance this media caster is associated with
long	<a href="#"><u>getConnectLastAttempt</u></a> ( ) Get system time in milliseconds of last connection attempt
long	<a href="#"><u>getConnectLastForceReset</u></a> ( ) Get system time in milliseconds of last time forceReset was called
long	<a href="#"><u>getConnectLastSuccess</u></a> ( ) Get system time in milliseconds of last connection success
int	<a href="#"><u>getIdleTimeout</u></a> ( ) Get the idle timeout for this media caster (milliseconds)
<a href="#"><u>MediaCasterItem</u></a>	<a href="#"><u>getMediaCasterDef</u></a> ( ) Get the media caster definition
String	<a href="#"><u>getMediaCasterId</u></a> ( ) Get the media caster id
<a href="#"><u>MediaCasterStreamItem</u></a>	<a href="#"><u>getMediaCasterStreamItem</u></a> ( ) Get the media caster item associated with this media caster
int	<a href="#"><u>getMediaCasterType</u></a> ( ) Get the media caster type.
int	<a href="#"><u>getReconnectWaitTime</u></a> ( ) Get the minimum time between reconnect attempts (milliseconds)
<a href="#"><u>IMediaStream</u></a>	<a href="#"><u>getStream</u></a> ( ) Get the underlying stream being used by this media caster
Object	<a href="#"><u>getStreamIsRunningLock</u></a> ( ) Get stream running lock
long	<a href="#"><u>getStreamLastSeq</u></a> ( ) Get the AMFPacket sequence number of last watchdog processed packet
long	<a href="#"><u>getStreamMissingTime</u></a> ( ) Get the time in milliseconds the stream has been missing
int	<a href="#"><u>getStreamTimeout</u></a> ( ) Get the watchdog stream timeout (milliseconds)
long	<a href="#"><u>getStreamTimeoutLastReset</u></a> ( ) Get system time in milliseconds of last time stream was reset due to stream timeout (debug)
long	<a href="#"><u>getStreamTimeoutLastTime</u></a> ( ) Get system time in milliseconds of last time stream was considered in missing state (debug)

int	<a href="#"><u>getStreamTimeoutReason()</u></a> Get the reason the stream is in timeout condition (debug)
<a href="#"><u>IVHost</u></a>	<a href="#"><u>getVHost()</u></a> Get the virtual host associated with this media caster
void	<a href="#"><u>init(MediaCasterStreamItem mediaCasterStreamItem, MediaCasterItem mediaCasterDef, IApplicationInstance appInstance, String mediaCasterId, String streamExt)</u></a> Initialize the media caster
boolean	<a href="#"><u>isSession()</u></a> Is there current a session attached to this MediaCaster
boolean	<a href="#"><u>isStream()</u></a> Is there a stream associated with this MediaCaster
boolean	<a href="#"><u>isStreamIsRunning()</u></a> Return true if stream is currently running
void	<a href="#"><u>registerPlayer(IMediaStreamPlay player)</u></a> Register a player with this media caster
void	<a href="#"><u>sessionClosed(org.apache.mina.common.Session session)</u></a> sessionClosed callback
void	<a href="#"><u>sessionOpened(org.apache.mina.common.Session session)</u></a> sessionOpened callback
void	<a href="#"><u>setAppInstance(IApplicationInstance appInstance)</u></a> Set the application instance this media caster is associated with
void	<a href="#"><u>setMediaCasterDef(MediaCasterItem mediaCasterDef)</u></a> Set the media caster definition
void	<a href="#"><u>setMediaCasterId(String mediaCasterId)</u></a> Get the media caster id
void	<a href="#"><u>setMediaCasterType(int mediaCasterType)</u></a> Set the media caster type.
void	<a href="#"><u>setReconnectWaitTime(int reconnectWaitTime)</u></a> Set the minimum time between reconnect attempts (milliseconds)
void	<a href="#"><u>setStream(IMediaStream stream)</u></a> Set the underlying stream being used by this media caster
void	<a href="#"><u>setStreamTimeout(int streamTimeout)</u></a> Set the watchdog stream timeout (milliseconds)
void	<a href="#"><u>shutdown(boolean isAppInstanceShutdown)</u></a> Shutdown media caster
void	<a href="#"><u>unregisterPlayer(IMediaStreamPlay player)</u></a> Unregister a player with this media caster

## Fields

(continued from last page)

---

## STREAMTIMEOUTREASON\_UNKNOWN

```
public static final int STREAMTIMEOUTREASON_UNKNOWN
```

Constant value: **0**

---

## STREAMTIMEOUTREASON\_NOTIMEOUT

```
public static final int STREAMTIMEOUTREASON_NOTIMEOUT
```

Constant value: **1**

---

## STREAMTIMEOUTREASON\_NOSESSION

```
public static final int STREAMTIMEOUTREASON_NOSESSION
```

Constant value: **2**

---

## STREAMTIMEOUTREASON\_NOURL

```
public static final int STREAMTIMEOUTREASON_NOURL
```

Constant value: **3**

---

## STREAMTIMEOUTREASON\_NOSTREAM

```
public static final int STREAMTIMEOUTREASON_NOSTREAM
```

Constant value: **4**

---

## STREAMTIMEOUTREASON\_RECONNECTRUNNING

```
public static final int STREAMTIMEOUTREASON_RECONNECTRUNNING
```

Constant value: **5**

---

## STREAMTIMEOUTREASON\_NORTSPSESSION

```
public static final int STREAMTIMEOUTREASON_NORTSPSESSION
```

Constant value: **6**

---

## STREAMTIMEOUTREASON\_GOOD

```
public static final int STREAMTIMEOUTREASON_GOOD
```

Constant value: **100**

---

(continued from last page)

## STREAMTIMEOUTREASON\_MISSING

```
public static final int STREAMTIMEOUTREASON_MISSING
```

Constant value: **101**

## MEDIACASTERTYPE\_UNKNOWN

```
public static final int MEDIACASTERTYPE_UNKNOWN
```

Constant value: **0**

## MEDIACASTERTYPE\_LIVEREPEATER

```
public static final int MEDIACASTERTYPE_LIVEREPEATER
```

Constant value: **1**

## MEDIACASTERTYPE\_SHOUTCAST

```
public static final int MEDIACASTERTYPE_SHOUTCAST
```

Constant value: **2**

## MEDIACASTERTYPE\_RTPLIVE

```
public static final int MEDIACASTERTYPE_RTPLIVE
```

Constant value: **3**

## Methods

### init

```
public void init(MediaCasterStreamItem mediaCasterStreamItem,  
    MediaCasterItem mediaCasterDef,  
    IApplicationInstance appInstance,  
    String mediaCasterId,  
    String streamExt)
```

Initialize the media caster

#### Parameters:

`mediaCasterStreamItem` - media caster item  
`mediaCasterDef` - media caster definition  
`appInstance` - application instance  
`mediaCasterId` - media caster id  
`streamExt` - stream ext or prefix

### getVHost

```
public IVHost getVHost()
```

Get the virtual host associated with this media caster

(continued from last page)

**Returns:**

virtual host associated with this media caster

---

**getMediaCasterId**

```
public String getMediaCasterId()
```

Get the media caster id

**Returns:**

media caster id

---

**setMediaCasterId**

```
public void setMediaCasterId(String mediaCasterId)
```

Get the media caster id

**Parameters:**

mediaCasterId - media caster id

---

**getStream**

```
public IMediaStream getStream()
```

Get the underlying stream being used by this media caster

**Returns:**

underlying stream being used by this media caster

---

**setStream**

```
public void setStream(IMediaStream stream)
```

Set the underlying stream being used by this media caster

**Parameters:**

stream - underlying stream being used by this media caster

---

**getAppInstance**

```
public IApplicationInstance getAppInstance()
```

Get the application instance this media caster is associated with

**Returns:**

application instance this media caster is associated with

---

**setAppInstance**

```
public void setAppInstance(IApplicationInstance appInstance)
```

Set the application instance this media caster is associated with

**Parameters:**

appInstance - application instance this media caster is associated with

(continued from last page)

## getMediaCasterDef

```
public MediaCasterItem getMediaCasterDef( )
```

Get the media caster definition

**Returns:**

media caster definition

---

## setMediaCasterDef

```
public void setMediaCasterDef(MediaCasterItem mediaCasterDef)
```

Set the media caster definition

**Parameters:**

mediaCasterDef - media caster definition

---

## shutdown

```
public void shutdown(boolean isAppInstanceShutdown)
```

Shutdown media caster

**Parameters:**

isAppInstanceShutdown - is server shutdown

---

## registerPlayer

```
public void registerPlayer(IMediaStreamPlay player)
```

Register a player with this media caster

**Parameters:**

player - player to register

---

## unregisterPlayer

```
public void unregisterPlayer(IMediaStreamPlay player)
```

Unregister a player with this media caster

**Parameters:**

player - player to unregister

---

## getIdleTimeout

```
public int getIdleTimeout( )
```

Get the idle timeout for this media caster (milliseconds)

**Returns:**

idle timeout for this media caster (milliseconds)

---

## sessionOpened

```
public void sessionOpened(org.apache.mina.common.Session session)
```

(continued from last page)

sessionOpened callback

**Parameters:**

session - IO Session

---

## sessionClosed

```
public void sessionClosed(org.apache.mina.common.Session session)
```

sessionClosed callback

**Parameters:**

session - IO Session

---

## getMediaCasterStreamItem

```
public MediaCasterStreamItem getMediaCasterStreamItem()
```

Get the media caster item associated with this media caster

**Returns:**

media caster item associated with this media caster

---

## forceReset

```
public void forceReset()
```

Force a reset/reconnect of this media caster

---

## doWatchdog

```
public boolean doWatchdog()
```

Idle processor

**Returns:**

return true if media caster unloaded due to idle event

---

## getStreamTimeout

```
public int getStreamTimeout()
```

Get the watchdog stream timeout (milliseconds)

**Returns:**

stream timeout

---

## setStreamTimeout

```
public void setStreamTimeout(int streamTimeout)
```

Set the watchdog stream timeout (milliseconds)

**Parameters:**

streamTimeout - stream timeout

(continued from last page)

---

## getStreamMissingTime

```
public long getStreamMissingTime()
```

Get the time in milliseconds the stream has been missing

**Returns:**

time in milliseconds the stream has been missing

---

## getStreamLastSeq

```
public long getStreamLastSeq()
```

Get the AMFPacket sequence number of last watchdog processed packet

**Returns:**

AMFPacket sequence number

---

## getStreamTimeoutReason

```
public int getStreamTimeoutReason()
```

Get the reason the stream is in timeout condition (debug)

**Returns:**

reason the stream is in timeout condition (debug)

---

## getStreamTimeoutLastTime

```
public long getStreamTimeoutLastTime()
```

Get system time in milliseconds of last time stream was considered in missing state (debug)

**Returns:**

time in milliseconds of last time stream was considered in missing

---

## getStreamTimeoutLastReset

```
public long getStreamTimeoutLastReset()
```

Get system time in milliseconds of last time stream was reset due to stream timeout (debug)

**Returns:**

time in milliseconds of last time stream was reset

---

## isSession

```
public boolean isSession()
```

Is there current a session attached to this MediaCaster

**Returns:**

true is MediaCaster has session

---

## isStream

```
public boolean isStream()
```

---



(continued from last page)

Is there a stream associated with this MediaCaster

**Returns:**

true if stream associated with this MediaCaster

---

## getConnectLastAttempt

```
public long getConnectLastAttempt( )
```

Get system time in milliseconds of last connection attempt

**Returns:**

system time in milliseconds of last connection attempt

---

## getConnectLastSuccess

```
public long getConnectLastSuccess( )
```

Get system time in milliseconds of last connection success

**Returns:**

system time in milliseconds of last connection success

---

## getConnectLastForceReset

```
public long getConnectLastForceReset( )
```

Get system time in milliseconds of last time forceReset was called

**Returns:**

system time in milliseconds of last time forceReset was called

---

## isStreamIsRunning

```
public boolean isStreamIsRunning( )
```

Return true if stream is currently running

**Returns:**

true if stream is currently running

---

## getStreamIsRunningLock

```
public Object getStreamIsRunningLock( )
```

Get stream running lock

**Returns:**

stream running lock

---

## getReconnectWaitTime

```
public int getReconnectWaitTime( )
```

Get the minimum time between reconnect attempts (milliseconds)

**Returns:**

minimum time between reconnect attempts (milliseconds)

## setReconnectWaitTime

```
public void setReconnectWaitTime(int reconnectWaitTime)
```

Set the minimum time between reconnect attempts (milliseconds)

**Parameters:**

reconnectWaitTime - minimum time between reconnect attempts (milliseconds)

---

## getMediaCasterType

```
public int getMediaCasterType()
```

Get the media caster type. See IMediaCaster.MEDIACASTERTYPE\_\*

**Returns:**

media caster type

---

## setMediaCasterType

```
public void setMediaCasterType(int mediaCasterType)
```

Set the media caster type. See IMediaCaster.MEDIACASTERTYPE\_\*

**Parameters:**

mediaCasterType - media caster type

---

## com.wowza.wms.mediacaster Interface IMediaCasterDataReceiver

---

public interface **IMediaCasterDataReceiver**  
extends

---

### Method Summary

void	<a href="#"><code>onData</code></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

---

### Method Summary

com.wowza.wms.netconnection.NetConnection	<a href="#">getNetConnection()</a>
---	------------------------------------

---

### Methods

#### **getNetConnection**

public com.wowza.wms.netconnection.NetConnection **getNetConnection()**

## com.wowza.wms.mediacaster Interface IMediaCasterNotify

All Subinterfaces:

[IMediaCasterNotify2](#)

public interface **IMediaCasterNotify**  
extends

IMediaCasterNotify: listener interface to MediaCaster system. See IApplicationInstance.addMediaCasterListener().

### Method Summary

void	<a href="#">onMediaCasterCreate</a> ( <a href="#">IMediaCaster</a> mediaCaster) Invoked when mediaCaster created
void	<a href="#">onMediaCasterDestroy</a> ( <a href="#">IMediaCaster</a> mediaCaster) Invoked when MediaCaster destroyed
void	<a href="#">onRegisterPlayer</a> ( <a href="#">IMediaCaster</a> mediaCaster, <a href="#">IMediaStreamPlay</a> player) Invoked when a player is added to this mediaCaster
void	<a href="#">onSetSourceStream</a> ( <a href="#">IMediaCaster</a> mediaCaster, <a href="#">IMediaStream</a> stream) Invoked when soure stream is set (can be called with stream of null)
void	<a href="#">onUnRegisterPlayer</a> ( <a href="#">IMediaCaster</a> mediaCaster, <a href="#">IMediaStreamPlay</a> player) Invoked when a player is removed from this mediaCaster

### Methods

#### onMediaCasterCreate

```
public void onMediaCasterCreate(IMediaCaster mediaCaster)
```

Invoked when mediaCaster created

**Parameters:**

mediaCaster

#### onMediaCasterDestroy

```
public void onMediaCasterDestroy(IMediaCaster mediaCaster)
```

Invoked when MediaCaster destroyed

**Parameters:**

mediaCaster

#### onRegisterPlayer

```
public void onRegisterPlayer(IMediaCaster mediaCaster,  
    IMediaStreamPlay player)
```

(continued from last page)

Invoked when a player is added to this mediaCaster

**Parameters:**

mediaCaster  
player

---

## onUnRegisterPlayer

```
public void onUnRegisterPlayer(IMediaCaster mediaCaster,  
    IMediaStreamPlay player)
```

Invoked when a player is removed from this mediaCaster

**Parameters:**

mediaCaster  
player

---

## onSetSourceStream

```
public void onSetSourceStream(IMediaCaster mediaCaster,  
    IMediaStream stream)
```

Invoked when source stream is set (can be called with stream of null)

**Parameters:**

mediaCaster  
stream

## com.wowza.wms.mediacaster Interface IMediaCasterNotify2

All Superinterfaces:

[IMediaCasterNotify](#)

public interface **IMediaCasterNotify2**

extends [IMediaCasterNotify](#)

### Method Summary

void	<a href="#">onConnectFailure</a> ( <a href="#">IMediaCaster</a> mediaCaster) Invoked when a connection or reconnection attempt fails
void	<a href="#">onConnectStart</a> ( <a href="#">IMediaCaster</a> mediaCaster) Invoked when a connection or reconnection attempt is invoked
void	<a href="#">onConnectSuccess</a> ( <a href="#">IMediaCaster</a> mediaCaster) Invoked when a connection or reconnection attempt is successful
void	<a href="#">onStreamStart</a> ( <a href="#">IMediaCaster</a> mediaCaster) Invoked when the stream starts receiving media data from the media source.
void	<a href="#">onStreamStop</a> ( <a href="#">IMediaCaster</a> mediaCaster) Invoked when the stream stops receiving media data from the media source after the streamTimeout value has passed.

Methods inherited from interface [com.wowza.wms.mediacaster.IMediaCasterNotify](#)

[onMediaCasterCreate](#), [onMediaCasterDestroy](#), [onRegisterPlayer](#), [onSetSourceStream](#), [onUnRegisterPlayer](#)

### Methods

#### onConnectStart

public void **onConnectStart**([IMediaCaster](#) mediaCaster)

Invoked when a connection or reconnection attempt is invoked

**Parameters:**

mediaCaster

#### onConnectSuccess

public void **onConnectSuccess**([IMediaCaster](#) mediaCaster)

Invoked when a connection or reconnection attempt is successful

**Parameters:**

mediaCaster

## onConnectFailure

```
public void onConnectFailure(IMediaCaster mediaCaster)
```

Invoked when a connection or reconnection attempt fails

**Parameters:**

mediaCaster

---

## onStreamStart

```
public void onStreamStart(IMediaCaster mediaCaster)
```

Invoked when the stream starts receiving media data from the media source. This event will only be thrown if the MediaCaster property streamTimeout is set to a non-zero value. NOTE: This is not implemented yet (coming soon)

**Parameters:**

mediaCaster

---

## onStreamStop

```
public void onStreamStop(IMediaCaster mediaCaster)
```

Invoked when the stream stops receiving media data from the media source after the streamTimeout value has passed. This event will only be thrown if the MediaCaster property streamTimeout is set to a non-zero value. NOTE: This is not implemented yet (coming soon)

**Parameters:**

mediaCaster

---



## com.wowza.wms.mediacaster Interface IMediaCasterValidateMediaCaster

All Known Implementing Classes:

[ModuleMediaCasterStreamMonitorAdvanced](#)

public interface **IMediaCasterValidateMediaCaster**  
extends

### Method Summary

boolean	<a href="#">onResetMediaCaster</a> ( <a href="#">IApplicationInstance</a> appInstance, <a href="#">IMediaCaster</a> mediaCaster)
boolean	<a href="#">onValidateMediaCaster</a> ( <a href="#">IApplicationInstance</a> appInstance, <a href="#">IMediaCaster</a> mediaCaster)
void	<a href="#">onValidateMediaCastersStart</a> ( <a href="#">IApplicationInstance</a> appInstance)
void	<a href="#">onValidateMediaCastersStop</a> ( <a href="#">IApplicationInstance</a> appInstance)

### Methods

#### onValidateMediaCastersStart

public void **onValidateMediaCastersStart**([IApplicationInstance](#) appInstance)

#### onValidateMediaCastersStop

public void **onValidateMediaCastersStop**([IApplicationInstance](#) appInstance)

#### onValidateMediaCaster

public boolean **onValidateMediaCaster**([IApplicationInstance](#) appInstance, [IMediaCaster](#) mediaCaster)

#### onResetMediaCaster

public boolean **onResetMediaCaster**([IApplicationInstance](#) appInstance, [IMediaCaster](#) mediaCaster)

## com.wowza.wms.mediacaster

### Class MediaCasterItem

java.lang.Object

└─com.wowza.wms.mediacaster.MediaCasterItem

public class **MediaCasterItem**  
extends Object

#### Constructor Summary

public	<a href="#"><u>MediaCasterItem</u></a> (String name, String streamType, String baseClass) Media caster item constructor
--------	--

#### Method Summary

void	<a href="#"><u>clearProperty</u></a> (String name) Clear property
String	<a href="#"><u>getBaseClass</u></a> () Get base class
int	<a href="#"><u>getConnectionTimeout</u></a> () Get connection timeout (milliseconds)
String	<a href="#"><u>getDescription</u></a> () Get description
static String	<a href="#"><u>getIdString</u></a> (String name, String liveStreamPacketizer, String liveStreamRepeater) Get id string for this media caster item (not used - returns name unchanged)
int	<a href="#"><u>getKeepAliveTime</u></a> () Get keep alive time (milliseconds)
String	<a href="#"><u>getName</u></a> () Get name
<a href="#"><u>WMSProperties</u></a>	<a href="#"><u>getProperties</u></a> () Get properties collection
String	<a href="#"><u>getProperty</u></a> (String name) Get property value
String	<a href="#"><u>getStreamType</u></a> () Get the stream type
String	<a href="#"><u>idStringToName</u></a> (String idString) Convert name to id string (not used - returns id string unchanged)
static MediaCasterStreamId	<a href="#"><u>parseIdString</u></a> (String idString)

void	<a href="#"><code>setBaseClass</code></a> (String baseClass) Set base class
void	<a href="#"><code>setConnectionTimeout</code></a> (int connectionTimeout) Set connection timeout (milliseconds)
void	<a href="#"><code>setDescription</code></a> (String description) Set description
void	<a href="#"><code>setKeepAliveTime</code></a> (int keepAliveTime) Set keep alive time (milliseconds)
void	<a href="#"><code>setName</code></a> (String name) Set name
void	<a href="#"><code>setProperty</code></a> (String name, String value) Set a property
void	<a href="#"><code>setStreamType</code></a> (String streamType) Set stream type
String	<a href="#"><code>toString</code></a> () toString

#### Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## Constructors

### MediaCasterItem

```
public MediaCasterItem(String name,
                      String streamType,
                      String baseClass)
```

Media caster item constructor

#### Parameters:

name - media caster name  
streamType - stream type  
baseClass - base class

## Methods

### getStreamType

```
public String getStreamType()
```

Get the stream type

#### Returns:

stream type

(continued from last page)

## setStreamType

```
public void setStreamType(String streamType)
```

Set stream type

**Parameters:**

streamType - stream type

---

## getName

```
public String getName()
```

Get name

**Returns:**

name

---

## setName

```
public void setName(String name)
```

Set name

**Parameters:**

name - name

---

## setProperty

```
public void setProperty(String name,  
                        String value)
```

Set a property

**Parameters:**

name - name

value - value

---

## clearProperty

```
public void clearProperty(String name)
```

Clear property

**Parameters:**

name - name

---

## getProperty

```
public String getProperty(String name)
```

Get property value

**Parameters:**

name - name

**Returns:**

property value

---

## getProperties

```
public WMSProperties getProperties()
```

Get properties collection

**Returns:**  
properties collection

---

## getDescription

```
public String getDescription()
```

Get description

**Returns:**  
description

---

## setDescription

```
public void setDescription(String description)
```

Set description

**Parameters:**  
description - description

---

## getBaseClass

```
public String getBaseClass()
```

Get base class

**Returns:**  
base class

---

## setBaseClass

```
public void setBaseClass(String baseClass)
```

Set base class

**Parameters:**  
baseClass - base class

---

## getConnectionTimeout

```
public int getConnectionTimeout()
```

Get connection timeout (milliseconds)

**Returns:**  
connection timeout (milliseconds)

---

## setConnectionTimeout

```
public void setConnectionTimeout(int connectionTimeout)
```

---

(continued from last page)

Set connection timeout (milliseconds)

**Parameters:**

connectionTimeout - connection timeout (milliseconds)

---

## getKeepAliveTime

```
public int getKeepAliveTime()
```

Get keep alive time (milliseconds)

**Returns:**

keep alive time (milliseconds)

---

## setKeepAliveTime

```
public void setKeepAliveTime(int keepAliveTime)
```

Set keep alive time (milliseconds)

**Parameters:**

keepAliveTime - keep alive time (milliseconds)

---

## getIdString

```
public static String getIdString(String name,  
    String liveStreamPacketizer,  
    String liveStreamRepeater)
```

Get id string for this media caster item (not used - returns name unchanged)

**Parameters:**

name - name

**Returns:**

id string for this media caster item

---

## parseIdString

```
public static MediaCasterStreamId parseIdString(String idString)
```

---

## idStringToName

```
public String idStringToName(String idString)
```

Convert name to id string (not used - returns id string unchanged)

**Parameters:**

idString - id string

**Returns:**

name

---

## toString

```
public String toString()
```

(continued from last page)

toString

## com.wowza.wms.mediacaster Class MediaCasterList

java.lang.Object

└─com.wowza.wms.mediacaster.MediaCasterList

public class **MediaCasterList**  
extends Object

### Constructor Summary

public	<a href="#">MediaCasterList()</a> Constructor
--------	--

### Method Summary

<a href="#">MediaCasterItem</a>	<a href="#">getMediaCasterDef</a> (String name) 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, wait

### Constructors

#### MediaCasterList

public **MediaCasterList**()

Constructor

### Methods

#### getMediaCasterDefs

public java.util.Map **getMediaCasterDefs**()

Get map of media caster items

##### Returns:

map of media caster items



(continued from last page)

## getMediaCasterNames

```
public java.util.List getMediaCasterNames()
```

Get list of media caster names

**Returns:**

list of media caster names

---

## getMediaCasterDef

```
public MediaCasterItem getMediaCasterDef(String name)
```

Get media caster definition by name

**Parameters:**

name - name

**Returns:**

media caster definition

---

## com.wowza.wms.mediacaster Class MediaCasterSettings

java.lang.Object

└─com.wowza.wms.mediacaster.MediaCasterSettings

public class **MediaCasterSettings**  
extends Object

### Constructor Summary

public	<a href="#">MediaCasterSettings()</a>
--------	---------------------------------------

### Method Summary

HostPortConfig	<a href="#">getMediaCasterHostPortConfig()</a> Get media caster host port config
int	<a href="#">getMediaCasterProcessorCount()</a> Get the thread count use for this host port
void	<a href="#">setMediaCasterHostPortConfig</a> (HostPortConfig mediaCasterHostPortConfig) Set media caster host port config
void	<a href="#">setMediaCasterProcessorCount</a> (int mediaCasterProcessorCount) Set the thread count for this processor

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Constructors

#### MediaCasterSettings

public **MediaCasterSettings**()

### Methods

#### getMediaCasterHostPortConfig

public HostPortConfig **getMediaCasterHostPortConfig**()

Get media caster host port config

**Returns:**

host port config

---

## setMediaCasterHostPortConfig

```
public void setMediaCasterHostPortConfig(HostPortConfig mediaCasterHostPortConfig)
```

Set media caster host port config

**Parameters:**

mediaCasterHostPortConfig

---

## getMediaCasterProcessorCount

```
public int getMediaCasterProcessorCount()
```

Get the thread count use for this host port

**Returns:**

thread count use for this host port

---

## setMediaCasterProcessorCount

```
public void setMediaCasterProcessorCount(int mediaCasterProcessorCount)
```

Set the thread count for this processor

**Parameters:**

mediaCasterProcessorCount - thread count use for this host port

## com.wowza.wms.mediacaster Class MediaCasterStreamItem

java.lang.Object

└─com.wowza.wms.mediacaster.MediaCasterStreamItem

public class **MediaCasterStreamItem**  
extends Object

### Constructor Summary

public	<a href="#">MediaCasterStreamItem</a> (long uniqueId) Media caster item constructor
--------	--

### Method Summary

void	<a href="#">acquire</a> () Increment 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="#">init</a> (String mediaCasterId, String streamExt, <a href="#">MediaCasterItem</a> mediaCasterDef, <a href="#">MediaCasterStreamMap</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`, `wait`

## Constructors

### MediaCasterStreamItem

```
public MediaCasterStreamItem(long uniqueId)
```

Media caster item constructor

## Methods

### getLock

```
public Object getLock()
```

(continued from last page)

---

## 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)

---

(continued from last page)

**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

**Returns:**

current number of acquire locks on this media caster item

---

**acquire**

```
public void acquire()
```

(continued from last page)

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:**

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

---



(continued from last page)

## getLiveStreamRepeater

```
public String getLiveStreamRepeater()
```

Get the live stream repeater for this media caster stream item

**Returns:**

live stream repeater

---

## setLiveStreamRepeater

```
public void setLiveStreamRepeater(String liveStreamRepeater)
```

Set the live stream repeater for this media caster stream item

**Parameters:**

liveStreamRepeater - live stream repeater

## com.wowza.wms.mediacaster Class MediaCasterStreamManager

java.lang.Object

└─com.wowza.wms.mediacaster.MediaCasterStreamManager

public class **MediaCasterStreamManager**  
extends Object

### Constructor Summary

public	<a href="#">MediaCasterStreamManager</a> ( <a href="#">MediaCasterStreamMap</a> mediaCasterStreamMap)
--------	---

### Method Summary

String[]	<a href="#">getStreamArray</a> () Get a list of active streams
java.util.List	<a href="#">getStreamList</a> () Get a list of active streams
boolean	<a href="#">startStream</a> (String streamName, String mediaCasterType) Start a stream by name
boolean	<a href="#">stopStream</a> (String streamName) Stop a stream by name
boolean	<a href="#">streamExists</a> (String streamName) Returns true if stream exists

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Constructors

#### MediaCasterStreamManager

public **MediaCasterStreamManager**([MediaCasterStreamMap](#) mediaCasterStreamMap)

### Methods

#### getStreamArray

public String[] **getStreamArray**()

Get a list of active streams

---

(continued from last page)

**Returns:**

list of active streams

---

## getStreamList

```
public java.util.List getStreamList()
```

Get a list of active streams

**Returns:**

list of active streams

---

## streamExists

```
public boolean streamExists(String streamName)
```

Returns true if stream exists

**Parameters:**

streamName - stream name

**Returns:**

true if stream exists

---

## stopStream

```
public boolean stopStream(String streamName)
```

Stop a stream by name

**Parameters:**

streamName - stream name

**Returns:**

true is successful

---

## startStream

```
public boolean startStream(String streamName,  
    String mediaCasterType)
```

Start a stream by name

**Parameters:**

streamName - stream name

mediaCasterType - MediaCaster type as defined in the name field of conf/MediaCasters.xml

**Returns:**

true is successful

---

## com.wowza.wms.mediacaster

### Class MediaCasterStreamMap

java.lang.Object

└─com.wowza.wms.mediacaster.MediaCasterStreamMap

public class **MediaCasterStreamMap**  
extends Object

#### Constructor Summary

public	<a href="#">MediaCasterStreamMap</a> ( <a href="#">IApplicationInstance</a> appInstance) Create a new mediacaster map
--------	--

#### Method Summary

<a href="#">MediaCasterStreamItem</a>	<a href="#">acquire</a> (String streamName) Increment the lock count for a media caster item (so that it remains loaded even if all listeners go away).
<a href="#">MediaCasterStreamItem</a>	<a href="#">acquire</a> (String inStreamName, <a href="#">MediaCasterItem</a> mediaCasterDef) Increment the lock count for a media caster item (so that it remains loaded even if all listeners go away).
<a href="#">MediaCasterStreamItem</a>	<a href="#">acquire</a> (String inStreamName, String streamType) Increment the lock count for a media caster item (so that it remains loaded even if all listeners go away).
<a href="#">MediaCasterStreamItem</a>	<a href="#">acquire</a> (String streamName, String liveStreamPacketizer, String liveStreamRepeater) Increment the lock count for a media caster item (so that it remains loaded even if all listeners go away).
<a href="#">MediaCasterStreamItem</a>	<a href="#">acquire</a> (String inStreamName, String liveStreamPacketizer, String liveStreamRepeater, <a href="#">MediaCasterItem</a> mediaCasterDef) Increment the lock count for a media caster item (so that it remains loaded even if all listeners go away).
<a href="#">MediaCasterStreamItem</a>	<a href="#">acquire</a> (String inStreamName, String liveStreamPacketizer, String liveStreamRepeater, String streamType) Increment the lock count for a media caster item (so that it remains loaded even if all listeners go away).
void	<a href="#">addStreamSrcToMediaCaster</a> (long streamSrc, String mediaCasterId)
void	<a href="#">clearStreamSrcToMediaCaster</a> (long streamSrc)
void	<a href="#">doWatchdog</a> () Do periodic idle time processing
<a href="#">IApplicationInstance</a>	<a href="#">getApplicationInstance</a> () Get the parent application instance for this map

edu.emory.mathcs.backport.java.util.concurrent.locks.WMSReadWriteLock	<a href="#">getLock()</a> Get the read/write lock for this interface
<a href="#">MediaCasterStreamItem</a>	<a href="#">getMediaCaster(String streamName)</a> Get media caster item based on given stream name
<a href="#">MediaCasterStreamItem</a>	<a href="#">getMediaCaster(String streamName, String liveStreamPacketizer, String liveStreamRepeater)</a> Get media caster item based on given stream name
int	<a href="#">getMediaCasterCount()</a> Get the number of mediacasters current running
java.util.List	<a href="#">getMediaCasterNames()</a> Get a list of all the currently running media caster names
<a href="#">MediaCasterStreamManager</a>	<a href="#">getStreamManager()</a> Get the stream manager interface for managing the starting and stopping of streams
void	<a href="#">registerPlayer(IMediaStreamPlay player, MediaCasterItem mediaCasterDef)</a> Register a player to a media caster item (internal use)
void	<a href="#">release(MediaCasterStreamItem mediaCasterStreamItem)</a> Decrement lock count on media caster item
void	<a href="#">release(MediaCasterStreamItem mediaCasterStreamItem, boolean removeIfZero)</a>
void	<a href="#">remove(MediaCasterStreamItem mediaCasterStreamItem)</a>
void	<a href="#">shutdown(boolean isAppInstanceShutdown)</a> Shutdown this media caster and close all running media casters
String	<a href="#">streamSrcToMediaCaster(long streamSrc)</a>
void	<a href="#">unregisterPlayer(IMediaStreamPlay player, MediaCasterItem mediaCasterDef)</a> Unregister a player to a media caster item (internal use)

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructors

### MediaCasterStreamMap

public **MediaCasterStreamMap**([IApplicationInstance](#) appInstance)

Create a new mediacaster map

#### Parameters:

appInstance - application instance

(continued from last page)

## Methods

### getLock

```
public edu.emory.mathcs.backport.java.util.concurrent.locks.WMSReadWriteLock getLock()
```

Get the read/write lock for this interface

**Returns:**

read/write lock

---

### getStreamManager

```
public MediaCasterStreamManager getStreamManager()
```

Get the stream manager interface for managing the starting and stopping of streams

**Returns:**

stream manager interface

---

### getApplicationInstance

```
public IApplicationInstance getApplicationInstance()
```

Get the parent application instance for this map

**Returns:**

parent application instance for this map

---

### getMediaCasterCount

```
public int getMediaCasterCount()
```

Get the number of mediacasters current running

**Returns:**

number of mediacasters current running

---

### doWatchdog

```
public void doWatchdog()
```

Do periodic idle time processing

---

### shutdown

```
public void shutdown(boolean isAppInstanceShutdown)
```

Shutdown this media caster and close all running media casters

**Parameters:**

isAppInstanceShutdown - is this due to application shutdown

---

### getMediaCasterNames

```
public java.util.List getMediaCasterNames()
```

Get a list of all the currently running media caster names

(continued from last page)

**Returns:**

list of all the currently running media caster names

---

**getMediaCaster**

```
public MediaCasterStreamItem getMediaCaster(String streamName)
```

Get media caster item based on given stream name

**Parameters:**

streamName - stream name

**Returns:**

media caster item

---

**getMediaCaster**

```
public MediaCasterStreamItem getMediaCaster(String streamName,  
String liveStreamPacketizer,  
String liveStreamRepeater)
```

Get media caster item based on given stream name

**Parameters:**

streamName - stream name

liveStreamPacketizer - live stream packetizer name

liveStreamRepeater - live stream repeater name

**Returns:**

media caster item

---

**acquire**

```
public MediaCasterStreamItem acquire(String streamName,  
String liveStreamPacketizer,  
String liveStreamRepeater)
```

Increment the lock count for a media caster item (so that it remains loaded even if all listeners go away). Will load the media caster if not already loaded. Uses default application instance stream type.

**Parameters:**

streamName - stream name

liveStreamPacketizer - live stream packetizer name

liveStreamRepeater - live stream repeater name

**Returns:**

media caster item

---

**acquire**

```
public MediaCasterStreamItem acquire(String streamName)
```

Increment the lock count for a media caster item (so that it remains loaded even if all listeners go away). Will load the media caster if not already loaded. Uses default application instance stream type.

**Parameters:**

streamName - stream name

**Returns:**

(continued from last page)

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

---

## 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)
```



(continued from last page)

Increment the lock count for a media caster item (so that it remains loaded even if all listeners go away). Will load the media caster if not already loaded.

**Parameters:**

inStreamName - stream name  
liveStreamPacketizer - live stream packetizer name  
liveStreamRepeater - live stream repeater name  
streamType - stream type to use (null will use default application streamType)

**Returns:**

media caster item

---

**release**

```
public void release(MediaCasterStreamItem mediaCasterStreamItem)
```

Decrement lock count on media caster item

**Parameters:**

mediaCasterStreamItem - media caster item to decrement

---

**release**

```
public void release(MediaCasterStreamItem mediaCasterStreamItem,  
                    boolean removeIfZero)
```

**remove**

```
public void remove(MediaCasterStreamItem mediaCasterStreamItem)
```

---

**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)

---

(continued from last page)

**Parameters:**

player - player to register  
mediaCasterDef - media caster definition

---

**unregisterPlayer**

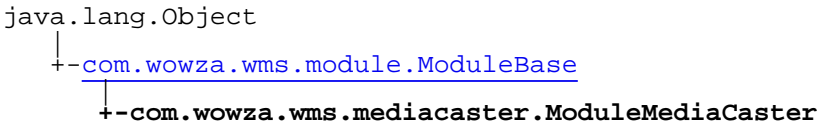
```
public void unregisterPlayer(IMediaStreamPlay player,  
    MediaCasterItem mediaCasterDef)
```

Unregister a player to a media caster item (internal use)

**Parameters:**

player - player to unregister  
mediaCasterDef - media caster definition

com.wowza.wms.mediacaster  
Class ModuleMediaCaster



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 <a href="#">com.wowza.wms.module.ModuleBase</a>
<a href="#">CALLBACK_PARAM1</a> , <a href="#">CALLBACK_PARAM10</a> , <a href="#">CALLBACK_PARAM2</a> , <a href="#">CALLBACK_PARAM3</a> , <a href="#">CALLBACK_PARAM4</a> , <a href="#">CALLBACK_PARAM5</a> , <a href="#">CALLBACK_PARAM6</a> , <a href="#">CALLBACK_PARAM7</a> , <a href="#">CALLBACK_PARAM8</a> , <a href="#">CALLBACK_PARAM9</a> , <a href="#">PARAM1</a> , <a href="#">PARAM10</a> , <a href="#">PARAM2</a> , <a href="#">PARAM3</a> , <a href="#">PARAM4</a> , <a href="#">PARAM5</a> , <a href="#">PARAM6</a> , <a href="#">PARAM7</a> , <a href="#">PARAM8</a> , <a href="#">PARAM9</a> , <a href="#">PARAMMETHODNAME</a> , <a href="#">PLAYTRANSITION_APPEND</a> , <a href="#">PLAYTRANSITION_APPEND_IMMEDIATE</a> , <a href="#">PLAYTRANSITION_RESET</a> , <a href="#">PLAYTRANSITION_RESET_IMMEDIATE</a> , <a href="#">PLAYTRANSITION_STOP</a> , <a href="#">PLAYTRANSITION_SWAP</a> , <a href="#">PLAYTRANSITION_SWITCH</a> , <a href="#">PLAYTRANSITION_UNKNOWN</a> , <a href="#">PLAYTRANSITIONSTR_APPEND</a> , <a href="#">PLAYTRANSITIONSTR_RESET</a> , <a href="#">PLAYTRANSITIONSTR_STOP</a> , <a href="#">PLAYTRANSITIONSTR_SWAP</a> , <a href="#">PLAYTRANSITIONSTR_SWITCH</a> , <a href="#">PLAYTRANSITIONSTR_UNKNOWN</a>

Constructor Summary	
public	<a href="#">ModuleMediaCaster</a> ( )

Method Summary	
void	<a href="#">acquireMediaCaster</a> ( <a href="#">IClient</a> client, <a href="#">RequestFunction</a> function, <a href="#">AMFDataList</a> params) Increment the lock count of a media caster stream.
void	<a href="#">getLockCount</a> ( <a href="#">IClient</a> client, <a href="#">RequestFunction</a> function, <a href="#">AMFDataList</a> params) Get the current lock count for a stream
void	<a href="#">getPlayerCount</a> ( <a href="#">IClient</a> client, <a href="#">RequestFunction</a> function, <a href="#">AMFDataList</a> params) Get the numbers of players associated with a particular media caster
void	<a href="#">getStreamNames</a> ( <a href="#">IClient</a> client, <a href="#">RequestFunction</a> function, <a href="#">AMFDataList</a> params) Get an array of stream names that are media casters associate with this application instance (returned as AMFDataArray)
void	<a href="#">onAppStart</a> ( <a href="#">IApplicationInstance</a> appInstance) onAppStart

void	<a href="#">onAppStop</a> ( <a href="#">IApplicationInstance</a> appInstance) onAppStop
void	<a href="#">releaseMediaCaster</a> ( <a href="#">IClient</a> client, <a href="#">RequestFunction</a> function, <a href="#">AMFDataList</a> params) Decrement the lock count of a media caster stream
void	<a href="#">resetStream</a> ( <a href="#">IClient</a> client, <a href="#">RequestFunction</a> function, <a href="#">AMFDataList</a> params) Reset a media caster stream
void	<a href="#">shutdownStream</a> ( <a href="#">IClient</a> client, <a href="#">RequestFunction</a> function, <a href="#">AMFDataList</a> params) Force shutdown a media caster stream

#### Methods inherited from class [com.wowza.wms.module.ModuleBase](#)

[getAppInstance](#), [getApplication](#), [getCallbackParamCount](#), [getLogger](#), [getParam](#), [getParamBoolean](#), [getParamBoolean](#), [getParamCount](#), [getParamDate](#), [getParamDouble](#), [getParamDouble](#), [getParamInt](#), [getParamInt](#), [getParamLong](#), [getParamLong](#), [getParamMixedArray](#), [getParamObj](#), [getParamString](#), [getParamString](#), [getParamType](#), [getStream](#), [getVHost](#), [invokePrevious](#), [invokePrevious](#), [isSendResult](#), [sendClientOnStatusError](#), [sendResult](#), [sendResult](#), [sendResult](#), [sendResult](#), [sendResult](#), [sendStreamOnStatusError](#)

#### Methods inherited from class [java.lang.Object](#)

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

#### Methods inherited from interface [com.wowza.wms.module.IModuleOnApp](#)

[onAppStart](#), [onAppStop](#)

## Constructors

### ModuleMediaCaster

```
public ModuleMediaCaster()
```

## Methods

### onAppStart

```
public void onAppStart(IApplicationInstance appInstance)
```

onAppStart

### onAppStop

```
public void onAppStop(IApplicationInstance appInstance)
```

onAppStop

## getLockCount

```
public void getLockCount(IClient client,  
    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,  
    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,  
    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,  
    RequestFunction function,  
    AMFDataList params)
```

Reset a media caster stream

**Parameters:**

client - client  
function - function  
params - {streamName}

---

## shutdownStream

```
public void shutdownStream(IClient client,  
    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,  
    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,  
    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.shoutcast**

## com.wowza.wms.mediacaster.shoutcast Interface IShoutCastFrameReceiver

public interface **IShoutCastFrameReceiver**  
extends

### Method Summary

void	<a href="#"><code>onCodecConfigAAC</code></a> (com.wowza.wms.media.aac.AACFrame frame, byte[] buffer, long offset)
void	<a href="#"><code>onFrameAAC</code></a> (com.wowza.wms.media.aac.AACFrame frame, byte[] buffer, long offset)
void	<a href="#"><code>onFrameMP3</code></a> (int frequency, int samplesPerFrame, int channels, byte[] syncHeader, byte[] packetHeader, byte[] frameData)
void	<a href="#"><code>onHeaderData</code></a> (java.util.Map headerMap)
void	<a href="#"><code>onMetaData</code></a> (java.util.Map metaMap)
void	<a href="#"><code>onTrim</code></a> ()

### Methods

#### **onFrameMP3**

```
public void onFrameMP3(int frequency,  
    int samplesPerFrame,  
    int channels,  
    byte[] syncHeader,  
    byte[] packetHeader,  
    byte[] frameData)
```

#### **onCodecConfigAAC**

```
public void onCodecConfigAAC(com.wowza.wms.media.aac.AACFrame frame,  
    byte[] buffer,  
    long offset)
```

#### **onFrameAAC**

```
public void onFrameAAC(com.wowza.wms.media.aac.AACFrame frame,  
    byte[] buffer,  
    long offset)
```



---

## **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

---

### Method Summary

void	<a href="#">addMetaDataListener</a> ( <a href="#">IShoutCastMetaDataNotify</a> listener)
boolean	<a href="#">removeMetaDataListener</a> ( <a href="#">IShoutCastMetaDataNotify</a> listener)

---

### Methods

#### **addMetaDataListener**

public void **addMetaDataListener**([IShoutCastMetaDataNotify](#) listener)

---

#### **removeMetaDataListener**

public boolean **removeMetaDataListener**([IShoutCastMetaDataNotify](#) listener)

## com.wowza.wms.mediacaster.shoutcast Interface IShoutCastMetaDataNotify

public interface **IShoutCastMetaDataNotify**  
extends

### Method Summary

void	<a href="#"><code>onAACEncodeInfo</code></a> ( <a href="#"><code>IMediaCaster</code></a> mediaCaster, int frequency, int channels, int samplesPerFrame)
void	<a href="#"><code>onHeaderData</code></a> ( <a href="#"><code>IMediaCaster</code></a> mediaCaster, java.util.Map headerMap)
void	<a href="#"><code>onMetaData</code></a> ( <a href="#"><code>IMediaCaster</code></a> mediaCaster, java.util.Map metaMap)
void	<a href="#"><code>onMP3EncodeInfo</code></a> ( <a href="#"><code>IMediaCaster</code></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)
```

---

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="#">onResult</a> ( <a href="#">IClient</a> client, <a href="#">RequestFunction</a> function, <a href="#">AMFDataList</a> params) Triggered on client side result from call to IClient.call
------	---

### Methods

#### onResult

```
public void onResult(IClient client,  
    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

---

### Method Summary

void	<a href="#"><u>onModuleLoad</u></a> (ModuleItem item)
void	<a href="#"><u>onModuleUnload</u></a> (ModuleItem item)

---

### Methods

#### **onModuleLoad**

public void **onModuleLoad**(ModuleItem item)

---

#### **onModuleUnload**

public void **onModuleUnload**(ModuleItem item)

## com.wowza.wms.module Interface IModuleOnApp

All Known Implementing Classes:  
[ModuleMediaCaster](#)

public interface **IModuleOnApp**  
extends

IModuleCallResult: method interface examples for application level methods in a module.

Since module method must be implemented as static method a module cannot directly implements this interface. This interface only serves as an example of the method names and call signature needed to implement these application methods.

### Method Summary

void	<a href="#">onAppStart</a> ( <a href="#">IApplicationInstance</a> appInstance) Invoked when an application instance is started.
void	<a href="#">onAppStop</a> ( <a href="#">IApplicationInstance</a> appInstance) Invoked when an application instance is stopped (destroyed).

### Methods

#### onAppStart

public void **onAppStart**([IApplicationInstance](#) appInstance)

Invoked when an application instance is started.

**Parameters:**

appInstance - application instance

#### onAppStop

public void **onAppStop**([IApplicationInstance](#) appInstance)

Invoked when an application instance is stopped (destroyed).

**Parameters:**

appInstance - application instance

# com.wowza.wms.module

## Interface IModuleOnCall

public interface **IModuleOnCall**  
extends

IModuleOnCall: method interface example for the catch-all method handler onCall.

Since module method must be implemented as static method a module cannot directly implements this interface. This interface only serves as an example of the method name and call signature needed to implement this method. The onCall method, when defined in a module, is invoked for all handlers that are undefined in a given module. The onCall handler can also be used to catch calls to server side component calls.

Method Summary	
void	<code>onCall(String handlerName, <a href="#">IClient</a> client, <a href="#">RequestFunction</a> function, <a href="#">AMFDataList</a> params)</code> Catch-all method handler.

## Methods

### onCall

```
public void onCall(String handlerName,  
    IClient client,  
    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="#">onConnect</a> ( <a href="#">IClient</a> client, <a href="#">RequestFunction</a> function, <a href="#">AMFDataList</a> params) Invoked when a client connection is initiated.
void	<a href="#">onConnectAccept</a> ( <a href="#">IClient</a> client) Invoked when a client connection is accepted.
void	<a href="#">onConnectReject</a> ( <a href="#">IClient</a> client) Invoked when a client connection is rejected.
void	<a href="#">onDisconnect</a> ( <a href="#">IClient</a> client) Invoked when a client disconnects.

### Methods

#### onConnect

```
public void onConnect(IClient client,  
    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

### Method Summary

void	<a href="#">onHTTPCupertinoEncryptionKeyCreateLive</a> ( <a href="#">IApplicationInstance</a> appInstance, String streamName, byte[] encKey)
void	<a href="#">onHTTPCupertinoEncryptionKeyCreateVOD</a> (HTTPStreamerSessionCupertino httpSession, byte[] encKey)
void	<a href="#">onHTTPCupertinoEncryptionKeyRequest</a> (HTTPStreamerSessionCupertino httpSession, <a href="#">IHTTPRequest</a> req, <a href="#">IHTTPResponse</a> resp)

### Methods

#### onHTTPCupertinoEncryptionKeyRequest

```
public void onHTTPCupertinoEncryptionKeyRequest(HTTPStreamerSessionCupertino
    httpSession,
    IHTTPRequest req,
    IHTTPResponse resp)
```

#### onHTTPCupertinoEncryptionKeyCreateVOD

```
public void onHTTPCupertinoEncryptionKeyCreateVOD(HTTPStreamerSessionCupertino
    httpSession,
    byte[] encKey)
```

#### onHTTPCupertinoEncryptionKeyCreateLive

```
public void onHTTPCupertinoEncryptionKeyCreateLive(IApplicationInstance appInstance,
    String streamName,
    byte[] encKey)
```

## com.wowza.wms.module

# Interface IModuleOnHTTPCupertinoStreamingSession

public interface **IModuleOnHTTPCupertinoStreamingSession**  
extends

IModuleOnHTTPSession: method interface for HTTP Cupertino Streaming session create/destroy.

## Method Summary

void	<a href="#"><code>onHTTPCupertinoStreamingSessionCreate</code></a> (HTTPStreamerSessionCupertino httpCupertinoStreamingSession) Invoked when an HTTP Cupertino Streaming session is created.
void	<a href="#"><code>onHTTPCupertinoStreamingSessionDestroy</code></a> (HTTPStreamerSessionCupertino httpCupertinoStreamingSession) Invoked when an HTTP Cupertino Streaming session is destroyed.

## Methods

### **onHTTPCupertinoStreamingSessionCreate**

public void **onHTTPCupertinoStreamingSessionCreate**(HTTPStreamerSessionCupertino httpCupertinoStreamingSession)

Invoked when an HTTP Cupertino Streaming session is created.

**Parameters:**

httpCupertinoStreamingSession - httpCupertinoStreamingSession

### **onHTTPCupertinoStreamingSessionDestroy**

public void **onHTTPCupertinoStreamingSessionDestroy**(HTTPStreamerSessionCupertino httpCupertinoStreamingSession)

Invoked when an HTTP Cupertino Streaming session is destroyed.

**Parameters:**

httpCupertinoStreamingSession - httpCupertinoStreamingSession

com.wowza.wms.module

# Interface IModuleOnHTTPSanJoseStreamingSession

public interface IModuleOnHTTPSanJoseStreamingSession  
extends

IModuleOnHTTPSanJoseStreamingSession: method interface for HTTP SanJose Streaming session create/destroy.

Method Summary	
void	<a href="#">onHTTPSanJoseStreamingSessionCreate</a> (HTTPStreamerSessionSanJose httpSanJoseStreamingSession) Invoked when an HTTP SanJose Streaming session is created.
void	<a href="#">onHTTPSanJoseStreamingSessionDestroy</a> (HTTPStreamerSessionSanJose httpSanJoseStreamingSession) Invoked when an HTTP SanJose Streaming session is destroyed.

## Methods

### onHTTPSanJoseStreamingSessionCreate

public void **onHTTPSanJoseStreamingSessionCreate**(HTTPStreamerSessionSanJose httpSanJoseStreamingSession)

Invoked when an HTTP SanJose Streaming session is created.

**Parameters:**

httpSanJoseStreamingSession - httpSanJoseStreamingSession

### onHTTPSanJoseStreamingSessionDestroy

public void **onHTTPSanJoseStreamingSessionDestroy**(HTTPStreamerSessionSanJose httpSanJoseStreamingSession)

Invoked when an HTTP SanJose Streaming session is destroyed.

**Parameters:**

httpSanJoseStreamingSession - httpSanJoseStreamingSession

## com.wowza.wms.module Interface IModuleOnHTTPSession

public interface **IModuleOnHTTPSession**  
extends

IModuleOnHTTPSession: method interface for HTTP Streaming session create/destroy.

### Method Summary

void	<a href="#">onHTTPSessionCreate</a> ( <a href="#">IHTTPStreamerSession</a> httpSession) Invoked when an HTTP Session is created (both Smooth and Cupertino sessions).
void	<a href="#">onHTTPSessionDestroy</a> ( <a href="#">IHTTPStreamerSession</a> httpSession) Invoked when an HTTP Session is destroyed (both Smooth and Cupertino sessions).

### Methods

#### onHTTPSessionCreate

public void **onHTTPSessionCreate**([IHTTPStreamerSession](#) httpSession)

Invoked when an HTTP Session is created (both Smooth and Cupertino sessions).

**Parameters:**

httpSession - httpSession

#### onHTTPSessionDestroy

public void **onHTTPSessionDestroy**([IHTTPStreamerSession](#) httpSession)

Invoked when an HTTP Session is destroyed (both Smooth and Cupertino sessions).

**Parameters:**

httpSession - httpSession

com.wowza.wms.module

# Interface IModuleOnHTTPSmoothStreamingSession

public interface **IModuleOnHTTPSmoothStreamingSession**  
extends

IModuleOnHTTPSession: method interface for HTTP Smooth Streaming session create/destroy.

Method Summary	
void	<a href="#">onHTTPSmoothStreamingSessionCreate</a> (HTTPStreamerSessionSmoothStreamer httpSmoothStreamingSession) Invoked when an HTTP Smooth Streaming session is created.
void	<a href="#">onHTTPSmoothStreamingSessionDestroy</a> (HTTPStreamerSessionSmoothStreamer httpSmoothStreamingSession) Invoked when an HTTP Smooth Streaming session is destroyed.

## Methods

### onHTTPSmoothStreamingSessionCreate

public void **onHTTPSmoothStreamingSessionCreate**(HTTPStreamerSessionSmoothStreamer httpSmoothStreamingSession)

Invoked when an HTTP Smooth Streaming session is created.

**Parameters:**

httpSmoothStreamingSession - httpSmoothStreamingSession

### onHTTPSmoothStreamingSessionDestroy

public void **onHTTPSmoothStreamingSessionDestroy**(HTTPStreamerSessionSmoothStreamer httpSmoothStreamingSession)

Invoked when an HTTP Smooth Streaming session is destroyed.

**Parameters:**

httpSmoothStreamingSession - httpSmoothStreamingSession

---

## com.wowza.wms.module Interface IModuleOnRTPSession

---

public interface **IModuleOnRTPSession**  
extends

IModuleOnRTPSession: method interface for RTP session create/destroy.

---

### Method Summary

void	<a href="#">onRTPSessionCreate(RTPSession rtpSession)</a> Invoked when an RTP Session is created.
void	<a href="#">onRTPSessionDestroy(RTPSession rtpSession)</a> Invoked when an RTP Session is destroyed.

---

### Methods

#### onRTPSessionCreate

public void **onRTPSessionCreate**([RTPSession](#) rtpSession)

Invoked when an RTP Session is created.

**Parameters:**

rtpSession - rtpSession

---

#### onRTPSessionDestroy

public void **onRTPSessionDestroy**([RTPSession](#) rtpSession)

Invoked when an RTP Session is destroyed.

**Parameters:**

rtpSession - rtpSession

---



## com.wowza.wms.module Interface IModuleOnStream

public interface **IModuleOnStream**  
extends

IModuleOnStream: method interface examples for stream level methods in a module.

### Method Summary

void	<a href="#">onStreamCreate</a> ( <a href="#">IMediaStream</a> stream) Invoked when a stream is created.
void	<a href="#">onStreamDestroy</a> ( <a href="#">IMediaStream</a> stream) Invoked when a stream is destroyed.

### Methods

#### onStreamCreate

public void **onStreamCreate**([IMediaStream](#) stream)

Invoked when a stream is created.

**Parameters:**

stream - stream object

#### onStreamDestroy

public void **onStreamDestroy**([IMediaStream](#) stream)

Invoked when a stream is destroyed.

**Parameters:**

stream - stream object

## com.wowza.wms.module Interface IModulePingResult

public interface **IModulePingResult**  
extends

IModulePingResult: callback interface used by IClient ping.

### Method Summary

void	<a href="#"><code>onResult(IClient client, long pingTime, int pingId, boolean result)</code></a> Triggered on client side result from call to IClient.ping.
------	--

### Methods

#### onResult

```
public void onResult(IClient client,  
                    long pingTime,  
                    int pingId,  
                    boolean result)
```

Triggered on client side result from call to IClient.ping. This method will be called on a success or failure (ping timeout) of a ping.

#### Parameters:

`client` - client  
`pingTime` - time in server time (milliseconds) when the ping was initiated  
`pingId` - internal server id for the ping request  
`result` - true is ping was successful false if failure (timeout)

## com.wowza.wms.module Class ModuleBase

java.lang.Object

└─com.wowza.wms.module.ModuleBase

Direct Known Subclasses:

[ModuleProperties](#), [ModuleMediaCasterStreamMonitorAdvanced](#), [ModuleFastPlay](#), [ModuleCore](#), [ModuleClientLogging](#), [ModuleMediaCaster](#)

public abstract class **ModuleBase**  
extends Object

ModuleBase: Base class that all server side modules should extends. Provides basic utility functionality for handling function parameters and return data. Also provides a simplified API for logging.

### Field Summary

public static final	<a href="#">CALLBACK_PARAM1</a> Callback param: param1 Value: <b>1</b>
public static final	<a href="#">CALLBACK_PARAM10</a> Callback param: param10 Value: <b>10</b>
public static final	<a href="#">CALLBACK_PARAM2</a> Callback param: param2 Value: <b>2</b>
public static final	<a href="#">CALLBACK_PARAM3</a> Callback param: param3 Value: <b>3</b>
public static final	<a href="#">CALLBACK_PARAM4</a> Callback param: param4 Value: <b>4</b>
public static final	<a href="#">CALLBACK_PARAM5</a> Callback param: param5 Value: <b>5</b>
public static final	<a href="#">CALLBACK_PARAM6</a> Callback param: param6 Value: <b>6</b>
public static final	<a href="#">CALLBACK_PARAM7</a> Callback param: param7 Value: <b>7</b>

public static final	<a href="#">CALLBACK_PARAM8</a> Callback param: param8 Value: <b>8</b>
public static final	<a href="#">CALLBACK_PARAM9</a> Callback param: param9 Value: <b>9</b>
public static final	<a href="#">PARAM1</a> Method param: param1 Value: <b>3</b>
public static final	<a href="#">PARAM10</a> Method param: param10 Value: <b>12</b>
public static final	<a href="#">PARAM2</a> Method param: param2 Value: <b>4</b>
public static final	<a href="#">PARAM3</a> Method param: param3 Value: <b>5</b>
public static final	<a href="#">PARAM4</a> Method param: param4 Value: <b>6</b>
public static final	<a href="#">PARAM5</a> Method param: param5 Value: <b>7</b>
public static final	<a href="#">PARAM6</a> Method param: param6 Value: <b>8</b>
public static final	<a href="#">PARAM7</a> Method param: param7 Value: <b>9</b>
public static final	<a href="#">PARAM8</a> Method param: param8 Value: <b>10</b>
public static final	<a href="#">PARAM9</a> Method param: param9 Value: <b>11</b>
public static final	<a href="#">PARAMMETHODNAME</a> Method: method name Value: <b>0</b>
public static final	<a href="#">PLAYTRANSITION_APPEND</a> Value: <b>0</b>

public static final	<a href="#"><u>PLAYTRANSITION_APPEND_IMMEDIATE</u></a> Value: <b>2</b>
public static final	<a href="#"><u>PLAYTRANSITION_RESET</u></a> Value: <b>1</b>
public static final	<a href="#"><u>PLAYTRANSITION_RESET_IMMEDIATE</u></a> Value: <b>3</b>
public static final	<a href="#"><u>PLAYTRANSITION_STOP</u></a> Value: <b>10</b>
public static final	<a href="#"><u>PLAYTRANSITION_SWAP</u></a> Value: <b>12</b>
public static final	<a href="#"><u>PLAYTRANSITION_SWITCH</u></a> Value: <b>13</b>
public static final	<a href="#"><u>PLAYTRANSITION_UNKNOWN</u></a> Value: <b>14</b>
public static final	<a href="#"><u>PLAYTRANSITIONSTR_APPEND</u></a> Play2 transition: APPEND Value: <b>append</b>
public static final	<a href="#"><u>PLAYTRANSITIONSTR_RESET</u></a> Play2 transition: RESET Value: <b>reset</b>
public static final	<a href="#"><u>PLAYTRANSITIONSTR_STOP</u></a> Play2 transition: STOP Value: <b>stop</b>
public static final	<a href="#"><u>PLAYTRANSITIONSTR_SWAP</u></a> Play2 transition: SWAP Value: <b>swap</b>
public static final	<a href="#"><u>PLAYTRANSITIONSTR_SWITCH</u></a> Play2 transition: SWITCH Value: <b>switch</b>
public static final	<a href="#"><u>PLAYTRANSITIONSTR_UNKNOWN</u></a> Play2 transition: UNKNOWN Value: <b>unknown</b>

## Constructor Summary

public	<a href="#"><u>ModuleBase()</u></a>
--------	-------------------------------------

## Method Summary

static <a href="#">IApplicationInstance</a>	<a href="#">getAppInstance</a> ( <a href="#">IClient</a> client) Get applicationInstace of a client.
static <a href="#">IApplication</a>	<a href="#">getApplication</a> ( <a href="#">IClient</a> client) Get application of a client.
static int	<a href="#">getCallbackParamCount</a> ( <a href="#">AMFDataList</a> params) Get the total number of parameters passed to callback.
static <a href="#">WMSLogger</a>	<a href="#">getLogger</a> () Get the logging interface.
static <a href="#">AMFData</a>	<a href="#">getParam</a> ( <a href="#">AMFDataList</a> params, int index) Get parameter by index.
static boolean	<a href="#">getParamBoolean</a> ( <a href="#">AMFDataList</a> params, int index) Get parameter by index, Return as boolean.
static boolean	<a href="#">getParamBoolean</a> ( <a href="#">AMFDataList</a> params, int index, boolean defaultVal) Get parameter by index, Return as boolean.
static int	<a href="#">getParamCount</a> ( <a href="#">AMFDataList</a> params) Get the total number of parameters passed to method.
static java.util.Date	<a href="#">getParamDate</a> ( <a href="#">AMFDataList</a> params, int index) Get parameter by index, Return as Date.
static double	<a href="#">getParamDouble</a> ( <a href="#">AMFDataList</a> params, int index) Get parameter by index, Return as double.
static double	<a href="#">getParamDouble</a> ( <a href="#">AMFDataList</a> params, int index, double defaultVal) Get parameter by index, Return as double.
static int	<a href="#">getParamInt</a> ( <a href="#">AMFDataList</a> params, int index) Get parameter by index, Return as int.
static int	<a href="#">getParamInt</a> ( <a href="#">AMFDataList</a> params, int index, int defaultVal) Get parameter by index, Return as int.
static long	<a href="#">getParamLong</a> ( <a href="#">AMFDataList</a> params, int index) Get parameter by index, Return as long.
static long	<a href="#">getParamLong</a> ( <a href="#">AMFDataList</a> params, int index, long defaultVal) Get parameter by index, Return as long.
static <a href="#">AMFDataMixedArray</a>	<a href="#">getParamMixedArray</a> ( <a href="#">AMFDataList</a> params, int index) Get parameter by index, Return as AMFDataMixedArray.
static <a href="#">AMFDataObj</a>	<a href="#">getParamObj</a> ( <a href="#">AMFDataList</a> params, int index) Get parameter by index, Return as Object.
static String	<a href="#">getParamString</a> ( <a href="#">AMFDataList</a> params, int index) Get parameter by index, Return as String.
static String	<a href="#">getParamString</a> ( <a href="#">AMFDataList</a> params, int index, String defaultVal) Get parameter by index, Return as String.
static int	<a href="#">getParamType</a> ( <a href="#">AMFDataList</a> params, int index) Get parameter type.

static <a href="#">IMediaStream</a>	<a href="#">getStream</a> ( <a href="#">IClient</a> client, <a href="#">RequestFunction</a> 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, <a href="#">RequestFunction</a> 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, <a href="#">RequestFunction</a> 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 boolean value.
static boolean	<a href="#">sendResult</a> ( <a href="#">IClient</a> client, <a href="#">AMFDataList</a> params, int value) Send a result to client method call as a single int value.
static boolean	<a href="#">sendResult</a> ( <a href="#">IClient</a> client, <a href="#">AMFDataList</a> params, String value) Send a result to client method call as a single String value.
static void	<a href="#">sendStreamOnStatusError</a> ( <a href="#">IMediaStream</a> stream, String code, String description) Send an error to the client-side NetStream.onStatus handler

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Fields

### PARAMMETHODNAME

public static final int **PARAMMETHODNAME**

Method: method name  
Constant value: 0

(continued from last page)

## PARAM1

```
public static final int PARAM1
```

Method param: param1  
Constant value: **3**

---

## PARAM2

```
public static final int PARAM2
```

Method param: param2  
Constant value: **4**

---

## PARAM3

```
public static final int PARAM3
```

Method param: param3  
Constant value: **5**

---

## PARAM4

```
public static final int PARAM4
```

Method param: param4  
Constant value: **6**

---

## PARAM5

```
public static final int PARAM5
```

Method param: param5  
Constant value: **7**

---

## PARAM6

```
public static final int PARAM6
```

Method param: param6  
Constant value: **8**

---

## PARAM7

```
public static final int PARAM7
```

Method param: param7  
Constant value: **9**

---

## PARAM8

```
public static final int PARAM8
```

Method param: param8  
Constant value: **10**

---

## PARAM9

```
public static final int PARAM9
```



(continued from last page)

Method param: param9  
Constant value: **11**

---

## PARAM10

public static final int **PARAM10**

Method param: param10  
Constant value: **12**

---

## CALLBACK\_PARAM1

public static final int **CALLBACK\_PARAM1**

Callback param: param1  
Constant value: **1**

---

## CALLBACK\_PARAM2

public static final int **CALLBACK\_PARAM2**

Callback param: param2  
Constant value: **2**

---

## CALLBACK\_PARAM3

public static final int **CALLBACK\_PARAM3**

Callback param: param3  
Constant value: **3**

---

## CALLBACK\_PARAM4

public static final int **CALLBACK\_PARAM4**

Callback param: param4  
Constant value: **4**

---

## CALLBACK\_PARAM5

public static final int **CALLBACK\_PARAM5**

Callback param: param5  
Constant value: **5**

---

## CALLBACK\_PARAM6

public static final int **CALLBACK\_PARAM6**

Callback param: param6  
Constant value: **6**

---

## CALLBACK\_PARAM7

public static final int **CALLBACK\_PARAM7**

Callback param: param7  
Constant value: **7**

---

## CALLBACK\_PARAM8

```
public static final int CALLBACK_PARAM8
```

Callback param: param8  
Constant value: **8**

---

## CALLBACK\_PARAM9

```
public static final int CALLBACK_PARAM9
```

Callback param: param9  
Constant value: **9**

---

## CALLBACK\_PARAM10

```
public static final int CALLBACK_PARAM10
```

Callback param: param10  
Constant value: **10**

---

## PLAYTRANSITIONSTR\_APPEND

```
public static final java.lang.String PLAYTRANSITIONSTR_APPEND
```

Play2 transition: APPEND  
Constant value: **append**

---

## PLAYTRANSITIONSTR\_RESET

```
public static final java.lang.String PLAYTRANSITIONSTR_RESET
```

Play2 transition: RESET  
Constant value: **reset**

---

## PLAYTRANSITIONSTR\_STOP

```
public static final java.lang.String PLAYTRANSITIONSTR_STOP
```

Play2 transition: STOP  
Constant value: **stop**

---

## PLAYTRANSITIONSTR\_SWAP

```
public static final java.lang.String PLAYTRANSITIONSTR_SWAP
```

Play2 transition: SWAP  
Constant value: **swap**

---

## PLAYTRANSITIONSTR\_SWITCH

```
public static final java.lang.String PLAYTRANSITIONSTR_SWITCH
```

Play2 transition: SWITCH  
Constant value: **switch**

---

(continued from last page)

---

## PLAYTRANSITIONSTR\_UNKNOWN

```
public static final java.lang.String PLAYTRANSITIONSTR_UNKNOWN
```

Play2 transition: UNKNOWN  
Constant value: **unknown**

---

## PLAYTRANSITION\_APPEND

```
public static final int PLAYTRANSITION_APPEND
```

Constant value: **0**

---

## PLAYTRANSITION\_RESET

```
public static final int PLAYTRANSITION_RESET
```

Constant value: **1**

---

## PLAYTRANSITION\_APPEND\_IMMEDIATE

```
public static final int PLAYTRANSITION_APPEND_IMMEDIATE
```

Constant value: **2**

---

## PLAYTRANSITION\_RESET\_IMMEDIATE

```
public static final int PLAYTRANSITION_RESET_IMMEDIATE
```

Constant value: **3**

---

## PLAYTRANSITION\_STOP

```
public static final int PLAYTRANSITION_STOP
```

Constant value: **10**

---

## PLAYTRANSITION\_SWAP

```
public static final int PLAYTRANSITION_SWAP
```

Constant value: **12**

---

## PLAYTRANSITION\_SWITCH

```
public static final int PLAYTRANSITION_SWITCH
```

Constant value: **13**

---

## PLAYTRANSITION\_UNKNOWN

```
public static final int PLAYTRANSITION_UNKNOWN
```

---

(continued from last page)

Constant value: **14**

## Constructors

### ModuleBase

```
public ModuleBase( )
```

## Methods

### getParamCount

```
protected static int getParamCount(AMFDataList params)
```

Get the total number of parameters passed to method.

**Parameters:**

params - parameters

**Returns:**

total number of parameters

### getCallbackParamCount

```
protected static int getCallbackParamCount(AMFDataList params)
```

Get the total number of parameters passed to callback.

**Parameters:**

params - parameters

**Returns:**

total number of parameters

### getParamType

```
protected static int getParamType(AMFDataList params,  
int index)
```

Get parameter type.

**Parameters:**

params - parameters

index - parameter index

**Returns:**

parameter type (AMFData.DATA\_TYPE\_\*)

### getParam

```
protected static AMFData getParam(AMFDataList params,  
int index)
```

Get parameter by index. Return as AMFData object.

(continued from last page)

**Parameters:**

params - parameters  
index - parameter index

**Returns:**

parameter value as AMFData object, null if out of bounds

---

## getParamMixedArray

```
protected static AMFDataMixedArray getParamMixedArray(AMFDataList params,  
int index)
```

Get parameter by index, Return as AMFDataMixedArray.

**Parameters:**

params - parameters  
index - parameter index

**Returns:**

parameter value as AMFDataMixedArray object, null if out of bounds

---

## getParamObj

```
protected static AMFDataObj getParamObj(AMFDataList params,  
int index)
```

Get parameter by index, Return as Object.

**Parameters:**

params - parameters  
index - parameter index

**Returns:**

parameter value as Object object, null if out of bounds

---

## getParamString

```
protected static String getParamString(AMFDataList params,  
int index)
```

Get parameter by index, Return as String.

**Parameters:**

params - parameters  
index - parameter index

**Returns:**

parameter value as String object, null if out of bounds

---

## getParamString

```
protected static String getParamString(AMFDataList params,  
int index,  
String defaultVal)
```

Get parameter by index, Return as String.

**Parameters:**

params - parameters

---

---

(continued from last page)

index - parameter index  
defaultVal - default value

**Returns:**

parameter value as String object, defaultVal if out of bounds

---

## getParamDate

```
protected static java.util.Date getParamDate(AMFDataList params,  
int index)
```

Get parameter by index, Return as Date.

**Parameters:**

params - parameters  
index - parameter index

**Returns:**

parameter value as Date object, null if out of bounds

---

## getParamInt

```
protected static int getParamInt(AMFDataList params,  
int index)
```

Get parameter by index, Return as int.

**Parameters:**

params - parameters  
index - parameter index

**Returns:**

parameter value as int, 0 if out of bounds

---

## getParamInt

```
protected static int getParamInt(AMFDataList params,  
int index,  
int defaultVal)
```

Get parameter by index, Return as int.

**Parameters:**

params - parameters  
index - parameter index  
defaultVal - default value

**Returns:**

parameter value as int, defaultVal if out of bounds

---

## getParamDouble

```
protected static double getParamDouble(AMFDataList params,  
int index)
```

Get parameter by index, Return as double.

**Parameters:**

params - parameters

---

(continued from last page)

index - parameter index

**Returns:**

parameter value as double, 0 if out of bounds

---

## getParamDouble

```
protected static double getParamDouble(AMFDataList params,  
    int index,  
    double defaultVal)
```

Get parameter by index, Return as double.

**Parameters:**

params - parameters  
index - parameter index  
defaultVal - default value

**Returns:**

parameter value as double, 0 if out of bounds

---

## getParamLong

```
protected static long getParamLong(AMFDataList params,  
    int index)
```

Get parameter by index, Return as long.

**Parameters:**

params - parameters  
index - parameter index

**Returns:**

parameter value as long, 0 if out of bounds

---

## getParamLong

```
protected static long getParamLong(AMFDataList params,  
    int index,  
    long defaultVal)
```

Get parameter by index, Return as long.

**Parameters:**

params - parameters  
index - parameter index  
defaultVal - default value

**Returns:**

parameter value as long, defaultVal if out of bounds

---

## getParamBoolean

```
protected static boolean getParamBoolean(AMFDataList params,  
    int index)
```

Get parameter by index, Return as boolean.

**Parameters:**

params - parameters

---

(continued from last page)

index - parameter index

**Returns:**

parameter value as boolean, false if out of bounds

---

## getParamBoolean

```
protected static boolean getParamBoolean(AMFDataList params,  
int index,  
boolean defaultVal)
```

Get parameter by index, Return as boolean.

**Parameters:**

params - parameters  
index - parameter index  
defaultVal - default value

**Returns:**

parameter value as boolean, defaultVal if out of bounds

---

## getAppInstance

```
protected static IApplicationInstance getAppInstance(IClient client)
```

Get applicationInstance of a client.

**Parameters:**

client - client

**Returns:**

applicationInstance

---

## getApplication

```
protected static IApplication getApplication(IClient client)
```

Get application of a client.

**Parameters:**

client - client

**Returns:**

application

---

## isSendResult

```
protected static boolean isSendResult(AMFDataList params)
```

Is this method call expecting sendResult to be called. If on the client side the call to `NetConnection.call("handlerName", resultObj, param1...);` had a value for resultObj (non-null), then the method is expecting some type of result or return data. Calling a variant of sendResult will provide this callback.

**Parameters:**

params - parameters

**Returns:**

true if client side call is expecting call to sendResult

---



## getVHost

protected static [IVHost](#) **getVHost**([IClient](#) client)

Get vHost of a client.

**Parameters:**

client - client

**Returns:**

vHost

---

## sendResult

protected static boolean **sendResult**([IClient](#) client,  
[AMFDataList](#) params,  
String value)

Send a result to client method call as a single String value.

**Parameters:**

client - client

params - parameters

value - return value

**Returns:**

true if client side call is expecting call to sendResult

---

## sendResult

protected static boolean **sendResult**([IClient](#) client,  
[AMFDataList](#) params,  
boolean value)

Send a result to client method call as a single boolean value.

**Parameters:**

client - client

params - parameters

value - return value

**Returns:**

true if client side call is expecting call to sendResult

---

## sendResult

protected static boolean **sendResult**([IClient](#) client,  
[AMFDataList](#) params,  
int value)

Send a result to client method call as a single int value.

**Parameters:**

client - client

params - parameters

value - return value

**Returns:**

true if client side call is expecting call to sendResult

---

## sendResult

```
protected static boolean sendResult(IClient client,  
    AMFDataList params,  
    double value)
```

Send a result to client method call as a single 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,  
    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,  
    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,
    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,
    RequestFunction function,
    AMFDataList params)
```

Wowza Pro will determine command priority based on the order of the modules in the module list. Wowza Pro by default will only invoke the last module that defines a given command. This method enables you to call the method by the same name in the previous module that defines that command. For example if you have three modules defined in the module list; ModuleCore, ModuleSecureURLParams and ModuleMyModule and all three modules implement the "play" command. When "play" is invoked it will invoke the "play" command in ModuleMyModule. If in ModuleMyModule you would like to call the "play" command in ModuleSecureURLParams, you can call: this.invokePrevious(client, function, params); and it will invoke the "play" command defined in ModuleSecureURLParams.

### Parameters:

client - client object passed into command  
 function - function passed into command  
 params - parameters passed into command

## sendClientOnStatusError

```
protected static void sendClientOnStatusError(IClient client,
    String code,
    String description)
```

Send an error message to the client-side client.onStatus handler

### Parameters:

client - destination client  
 code - code  
 description - description

## sendStreamOnStatusError

```
protected static void sendStreamOnStatusError(IMediaStream stream,
    String code,
    String description)
```

Send an error to the client-side NetStream.onStatus handler

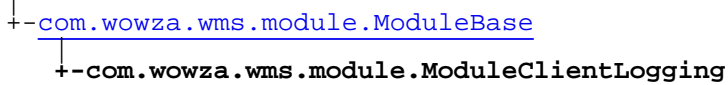
### Parameters:

(continued from last page)

stream - destination stream  
code - code  
description - description

## com.wowza.wms.module Class ModuleClientLogging

java.lang.Object



public class **ModuleClientLogging**  
extends [ModuleBase](#)

### Fields inherited from class [com.wowza.wms.module.ModuleBase](#)

[CALLBACK\\_PARAM1](#), [CALLBACK\\_PARAM10](#), [CALLBACK\\_PARAM2](#), [CALLBACK\\_PARAM3](#), [CALLBACK\\_PARAM4](#), [CALLBACK\\_PARAM5](#), [CALLBACK\\_PARAM6](#), [CALLBACK\\_PARAM7](#), [CALLBACK\\_PARAM8](#), [CALLBACK\\_PARAM9](#), [PARAM1](#), [PARAM10](#), [PARAM2](#), [PARAM3](#), [PARAM4](#), [PARAM5](#), [PARAM6](#), [PARAM7](#), [PARAM8](#), [PARAM9](#), [PARAMMETHODNAME](#), [PLAYTRANSITION\\_APPEND](#), [PLAYTRANSITION\\_APPEND\\_IMMEDIATE](#), [PLAYTRANSITION\\_RESET](#), [PLAYTRANSITION\\_RESET\\_IMMEDIATE](#), [PLAYTRANSITION\\_STOP](#), [PLAYTRANSITION\\_SWAP](#), [PLAYTRANSITION\\_SWITCH](#), [PLAYTRANSITION\\_UNKNOWN](#), [PLAYTRANSITIONSTR\\_APPEND](#), [PLAYTRANSITIONSTR\\_RESET](#), [PLAYTRANSITIONSTR\\_STOP](#), [PLAYTRANSITIONSTR\\_SWAP](#), [PLAYTRANSITIONSTR\\_SWITCH](#), [PLAYTRANSITIONSTR\\_UNKNOWN](#)

### Constructor Summary

public	<a href="#">ModuleClientLogging()</a>
--------	---------------------------------------

### Method Summary

static void	<a href="#">logDebug</a> ( <a href="#">IClient</a> client, <a href="#">RequestFunction</a> function, <a href="#">AMFDataList</a> params) Send a debug message to the logging system NetConnection.call("logDebug", null, message);
static void	<a href="#">logError</a> ( <a href="#">IClient</a> client, <a href="#">RequestFunction</a> function, <a href="#">AMFDataList</a> params) Send a error message to the logging system NetConnection.call("logError", null, message);
static void	<a href="#">logInfo</a> ( <a href="#">IClient</a> client, <a href="#">RequestFunction</a> function, <a href="#">AMFDataList</a> params) Send a info message to the logging system NetConnection.call("logInfo", null, message);
static void	<a href="#">logWarn</a> ( <a href="#">IClient</a> client, <a href="#">RequestFunction</a> function, <a href="#">AMFDataList</a> params) Send a warning message to the logging system NetConnection.call("logWarn", null, message);

### Methods inherited from class [com.wowza.wms.module.ModuleBase](#)

[getAppInstance](#), [getApplication](#), [getCallbackParamCount](#), [getLogger](#), [getParam](#), [getParamBoolean](#), [getParamBoolean](#), [getParamCount](#), [getParamDate](#), [getParamDouble](#), [getParamDouble](#), [getParamInt](#), [getParamInt](#), [getParamLong](#), [getParamLong](#), [getParamMixedArray](#), [getParamObj](#), [getParamString](#), [getParamString](#), [getParamType](#), [getStream](#), [getVHost](#), [invokePrevious](#), [invokePrevious](#), [isSendResult](#), [sendClientOnStatusError](#), [sendResult](#), [sendResult](#), [sendResult](#), [sendResult](#), [sendResult](#), [sendStreamOnStatusError](#)

### Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

## Constructors

### ModuleClientLogging

```
public ModuleClientLogging()
```

## Methods

### logDebug

```
public static void logDebug(IClient client,  
    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,  
    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,  
    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,  
    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, RequestFunction function, AMFDataList params)</a> Server side implementation of NetStream.close();
static void	<a href="#">createStream(IClient client, RequestFunction function, AMFDataList params)</a> Create new server side NetStream object (internal to Flash workings).
static void	<a href="#">deleteStream(IClient client, RequestFunction function, AMFDataList params)</a> Delete server side stream object (internal to Flash workings).
static void	<a href="#">FCPublish(IClient client, RequestFunction function, AMFDataList params)</a> FCPublish method called by FME 2.5
static void	<a href="#">FCSubscribe(IClient client, 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, RequestFunction function, AMFDataList params)</a>
static void	<a href="#">FCUnpublish(IClient client, RequestFunction function, AMFDataList params)</a> FCUnpublish method called by FME 2.5
static void	<a href="#">FCUnsubscribe(IClient client, RequestFunction function, AMFDataList params)</a> FCUnSubscribe to a live stream
static void	<a href="#">FCUnSubscribe(IClient client, RequestFunction function, AMFDataList params)</a>
static void	<a href="#">getClientID(IClient client, RequestFunction function, AMFDataList params)</a> Get the clientId for a client connection NetConnection.call("getClientID", resultObj);
static void	<a href="#">getLastStreamId(IClient client, RequestFunction function, AMFDataList params)</a> Get the id for the last created stream NetConnection.call("getLastStreamId", resultObj);
static void	<a href="#">getLiveStreamPacketizer(IClient client, 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, RequestFunction function, AMFDataList params)</a> getPageUrl returns the pageUrl from the onConnect metadata
static void	<a href="#">getReferrer(IClient client, RequestFunction function, AMFDataList params)</a> getReferrer returns the referrer from the onConnect metadata

static void	<a href="#">getRepeaterOriginUrl</a> ( <a href="#">IClient</a> client, <a href="#">RequestFunction</a> function, <a href="#">AMFDataList</a> params) Get the Repeater Origin URL for this client
void	<a href="#">getStreamBitrate</a> ( <a href="#">IClient</a> client, <a href="#">RequestFunction</a> function, <a href="#">AMFDataList</a> params) Get stream bitrate in bits-per-second for a video on demand stream (will not work for live).
static void	<a href="#">getStreamLength</a> ( <a href="#">IClient</a> client, <a href="#">RequestFunction</a> function, <a href="#">AMFDataList</a> params) Get stream length in seconds NetConnection.call("getStreamLength", resultObj, streamName); If you pass in an array of streamNames it will return an array of durations.
static void	<a href="#">getStreamType</a> ( <a href="#">IClient</a> client, <a href="#">RequestFunction</a> function, <a href="#">AMFDataList</a> params) Get the default stream type for a client connection NetConnection.call("getStreamType", resultObj);
static void	<a href="#">getVersion</a> ( <a href="#">IClient</a> client, <a href="#">RequestFunction</a> function, <a href="#">AMFDataList</a> params) Get the Wowza Pro server version and build number NetConnection.call("getVersion", resultObj);
static void	<a href="#">initLiveStreamRepeating</a> ( <a href="#">IClient</a> client, <a href="#">RequestFunction</a> function, <a href="#">AMFDataList</a> params) Initialize a stream for live stream repeating
static void	<a href="#">initStream</a> ( <a href="#">IClient</a> client, <a href="#">RequestFunction</a> function, <a href="#">AMFDataList</a> params) Create new server side NetStream object (internal to Flash workings).
static void	<a href="#">pause</a> ( <a href="#">IClient</a> client, <a href="#">RequestFunction</a> function, <a href="#">AMFDataList</a> params) Server side implementation of NetStream.pause([ flag : Boolean]);
static void	<a href="#">pauseRaw</a> ( <a href="#">IClient</a> client, <a href="#">RequestFunction</a> function, <a href="#">AMFDataList</a> params) pauseRaw method introduced in Flash player 10
static void	<a href="#">play</a> ( <a href="#">IClient</a> client, <a href="#">RequestFunction</a> function, <a href="#">AMFDataList</a> params) Server side implementation of NetStream.play(name : Object [,start : Number[, len : Number[, reset : Object]]]);
static void	<a href="#">play2</a> ( <a href="#">IClient</a> client, <a href="#">RequestFunction</a> function, <a href="#">AMFDataList</a> params) Server side implementation of NetStream.play(playOptions : NetStreamPlayOptions);
static void	<a href="#">publish</a> ( <a href="#">IClient</a> client, <a href="#">RequestFunction</a> function, <a href="#">AMFDataList</a> params) Server side implementation of NetStream.publish(name : String [, howToPublish : String]);
static void	<a href="#">receiveAudio</a> ( <a href="#">IClient</a> client, <a href="#">RequestFunction</a> function, <a href="#">AMFDataList</a> params) Server side implementation of NetStream.receiveAudio(receive : Boolean);
static void	<a href="#">receiveVideo</a> ( <a href="#">IClient</a> client, <a href="#">RequestFunction</a> function, <a href="#">AMFDataList</a> params) Server side implementation of NetStream.receiveVideo(receive : Boolean   FPS : Number); FPS does not work the same as FMS.
static void	<a href="#">releaseStream</a> ( <a href="#">IClient</a> client, <a href="#">RequestFunction</a> function, <a href="#">AMFDataList</a> params)
static void	<a href="#">seek</a> ( <a href="#">IClient</a> client, <a href="#">RequestFunction</a> function, <a href="#">AMFDataList</a> params) Server side implementation of NetStream.seek(offset : Number);

static void	<a href="#">setBandwidthLimit</a> ( <a href="#">IClient</a> client, <a href="#">RequestFunction</a> function, <a href="#">AMFDataList</a> params)
static void	<a href="#">setBufferTime</a> ( <a href="#">IClient</a> client, <a href="#">RequestFunction</a> function, <a href="#">AMFDataList</a> params) Server side implementation of NetStream.setBufferTime(bufferTime : Number);
static void	<a href="#">setLiveStreamPacketizer</a> ( <a href="#">IClient</a> client, <a href="#">RequestFunction</a> function, <a href="#">AMFDataList</a> params) Set the live stream packetizer for a stream
static void	<a href="#">setRepeaterOriginUrl</a> ( <a href="#">IClient</a> client, <a href="#">RequestFunction</a> function, <a href="#">AMFDataList</a> params) Set the Repeater Origin URL for this client
static void	<a href="#">setStreamType</a> ( <a href="#">IClient</a> client, <a href="#">RequestFunction</a> function, <a href="#">AMFDataList</a> params) Set the default stream type for a client connection NetConnection.call("setStreamType", null, streamType);

#### Methods inherited from class [com.wowza.wms.module.ModuleBase](#)

[getAppInstance](#), [getApplication](#), [getCallbackParamCount](#), [getLogger](#), [getParam](#), [getParamBoolean](#), [getParamBoolean](#), [getParamCount](#), [getParamDate](#), [getParamDouble](#), [getParamDouble](#), [getParamInt](#), [getParamInt](#), [getParamLong](#), [getParamLong](#), [getParamMixedArray](#), [getParamObj](#), [getParamString](#), [getParamString](#), [getParamType](#), [getStream](#), [getVHost](#), [invokePrevious](#), [invokePrevious](#), [isSendResult](#), [sendClientOnStatusError](#), [sendResult](#), [sendResult](#), [sendResult](#), [sendResult](#), [sendResult](#), [sendStreamOnStatusError](#)

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructors

### ModuleCore

```
public ModuleCore()
```

## Methods

### createStream

```
public static void createStream(IClient client,
    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,  
    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,  
    RequestFunction function,  
    AMFDataList params)
```

---

## getLastStreamId

```
public static void getLastStreamId(IClient client,  
    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,  
    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,  
    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,  
    RequestFunction function,  
    AMFDataList params)
```

---

## play2

```
public static void play2(IClient client,  
    RequestFunction function,  
    AMFDataList params)
```

Server side implementation of NetStream.play(playOptions : NetStreamPlayOptions);

### Parameters:

client  
function  
params

---

## play

```
public static void play(IClient client,  
    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,  
    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,  
    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,  
    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,  
    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,  
    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,  
    RequestFunction function,  
    AMFDataList params)
```

Get the Wowza Pro server version and build number NetConnection.call("getVersion", resultObj);

### Parameters:

client - client  
function - function  
params - params (no params)

---

## setLiveStreamPacketizer

```
public static void setLiveStreamPacketizer(IClient client,  
    RequestFunction function,  
    AMFDataList params)
```

Set the live stream packetizer for a stream

(continued from last page)

**Parameters:**

client - client  
function - function  
params - params (liveStreamPacketizer)

---

## initLiveStreamRepeating

```
public static void initLiveStreamRepeating(IClient client,  
    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,  
    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,  
    RequestFunction function,  
    AMFDataList params)
```

Set the default stream type for a client connection `NetConnection.call("setStreamType", null, streamType);`

**Parameters:**

client - client  
function - function  
params - params (streamType)

---

## getStreamType

```
public static void getStreamType(IClient client,  
    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)

---

(continued from last page)

## receiveAudio

```
public static void receiveAudio(IClient client,  
    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,  
    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,  
    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,  
    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



(continued from last page)

params - params (streamName:String or streamNames:Array)

---

## getRepeaterOriginUrl

```
public static void getRepeaterOriginUrl(IClient client,  
    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,  
    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,  
    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,  
    RequestFunction function,  
    AMFDataList params)
```

FCUnpublish method called by FME 2.5

### Parameters:

client - client  
function - function  
params - params

---

## FCUnpublish

```
public static void FCUnpublish(IClient client,  
    RequestFunction function,  
    AMFDataList params)
```

(continued from last page)

---

## pauseRaw

```
public static void pauseRaw(IClient client,  
    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,  
    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,  
    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,  
    RequestFunction function,  
    AMFDataList params)
```

FCSubscribe subscribes to a live stream (if origin edge will start the stream from the origin to the edge)

### Parameters:

client - client  
function - function  
params - params (streamName:String, [mediaCasterType:String])

---

## FCUnsubscribe

```
public static void FCUnsubscribe(IClient client,  
    RequestFunction function,  
    AMFDataList params)
```

(continued from last page)

FCUnSubscribe to a live stream

**Parameters:**

client - client

function - function

params - params (streamName:String)

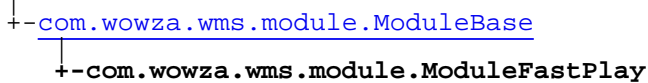
---

## FCUnSubscribe

```
public static void FCUnSubscribe(IClient client,  
    RequestFunction function,  
    AMFDataList params)
```

## com.wowza.wms.module Class ModuleFastPlay

java.lang.Object



```
public class ModuleFastPlay
extends ModuleBase
```

### Fields inherited from class [com.wowza.wms.module.ModuleBase](#)

[CALLBACK\\_PARAM1](#), [CALLBACK\\_PARAM10](#), [CALLBACK\\_PARAM2](#), [CALLBACK\\_PARAM3](#), [CALLBACK\\_PARAM4](#), [CALLBACK\\_PARAM5](#), [CALLBACK\\_PARAM6](#), [CALLBACK\\_PARAM7](#), [CALLBACK\\_PARAM8](#), [CALLBACK\\_PARAM9](#), [PARAM1](#), [PARAM10](#), [PARAM2](#), [PARAM3](#), [PARAM4](#), [PARAM5](#), [PARAM6](#), [PARAM7](#), [PARAM8](#), [PARAM9](#), [PARAMMETHODNAME](#), [PLAYTRANSITION\\_APPEND](#), [PLAYTRANSITION\\_APPEND\\_IMMEDIATE](#), [PLAYTRANSITION\\_RESET](#), [PLAYTRANSITION\\_RESET\\_IMMEDIATE](#), [PLAYTRANSITION\\_STOP](#), [PLAYTRANSITION\\_SWAP](#), [PLAYTRANSITION\\_SWITCH](#), [PLAYTRANSITION\\_UNKNOWN](#), [PLAYTRANSITIONSTR\\_APPEND](#), [PLAYTRANSITIONSTR\\_RESET](#), [PLAYTRANSITIONSTR\\_STOP](#), [PLAYTRANSITIONSTR\\_SWAP](#), [PLAYTRANSITIONSTR\\_SWITCH](#), [PLAYTRANSITIONSTR\\_UNKNOWN](#)

### Constructor Summary

public	<a href="#">ModuleFastPlay()</a>
--------	----------------------------------

### Method Summary

static void	<a href="#">setFastPlay</a> ( <a href="#">IClient</a> client, <a href="#">RequestFunction</a> function, <a href="#">AMFDataList</a> params) Turn on fast play NetStream.call("setFastPlay", null, multiplier, frames-per-second, direction);
-------------	---

### Methods inherited from class [com.wowza.wms.module.ModuleBase](#)

[getAppInstance](#), [getApplication](#), [getCallbackParamCount](#), [getLogger](#), [getParam](#), [getParamBoolean](#), [getParamBoolean](#), [getParamCount](#), [getParamDate](#), [getParamDouble](#), [getParamDouble](#), [getParamInt](#), [getParamInt](#), [getParamLong](#), [getParamLong](#), [getParamMixedArray](#), [getParamObj](#), [getParamString](#), [getParamString](#), [getParamType](#), [getStream](#), [getVHost](#), [invokePrevious](#), [invokePrevious](#), [isSendResult](#), [sendClientOnStatusError](#), [sendResult](#), [sendResult](#), [sendResult](#), [sendResult](#), [sendResult](#), [sendResult](#), [sendStreamOnStatusError](#)

### Methods inherited from class java.lang.Object

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

### Constructors

(continued from last page)

## ModuleFastPlay

```
public ModuleFastPlay()
```

## Methods

### setFastPlay

```
public static void setFastPlay(IClient client,  
    RequestFunction function,  
    AMFDataList params)
```

Turn on fast play `NetStream.call("setFastPlay", null, multiplier, frames-per-second, direction);`

**Parameters:**

client - client

function - function

params - params (multiplier, frames-per-second, direction)

## com.wowza.wms.module Class ModuleMediaCasterStreamMonitorAdvanced

```
java.lang.Object
|
+-com.wowza.wms.module.ModuleBase
|
+-com.wowza.wms.module.ModuleMediaCasterStreamMonitorAdvanced
```

All Implemented Interfaces:

[IMediaCasterValidateMediaCaster](#)

```
public class ModuleMediaCasterStreamMonitorAdvanced
extends ModuleBase
implements IMediaCasterValidateMediaCaster
```

ModuleMediaCasterStreamMonitorAdvanced: Advanced stream monitoring module.

Add this <Module> definition to the end of the <Modules> list in [install-dir]/conf/[application]/Application.xml:

```
<Module>
  <Name>ModuleMediaCasterStreamMonitorAdvanced</Name>
  <Description>ModuleMediaCasterStreamMonitorAdvanced</Description>
  <Class>com.wowza.wms.module.ModuleMediaCasterStreamMonitorAdvanced</Class>
</Module>
```

Add these properties to the application level <Properties> container at the bottom of [install-dir]/conf/[application]/Application.xml (be sure to get the correct <Properties> container - there are several in the Application.xml file). Carefully read the instructions above each set of properties and set accordingly:

```

<!--
Monitor incoming packets (stream, audio, video) to be sure packets continue to flow from
encoder to stream. The
streamMonitor[type]StartTimeout controls the timeout (milliseconds) for the first packet. The
streamMonitor[type]Timeout
controls the timeout (milliseconds) for packets after the first packet. The stream type
refers to a catch all of any packet of
any type (audio, video, data). If any of these values are set to zero, the test is turned
off.
-->
<Property>
    <Name>streamMonitorStreamStartTimeout</Name>
    <Value>20000</Value>
    <Type>Integer</Type>
</Property>
<Property>
    <Name>streamMonitorStreamTimeout</Name>
    <Value>12000</Value>
    <Type>Integer</Type>
</Property>
<Property>
    <Name>streamMonitorVideoStartTimeout</Name>
    <Value>0</Value>
    <Type>Integer</Type>
</Property>
<Property>
    <Name>streamMonitorVideoTimeout</Name>
    <Value>0</Value>
    <Type>Integer</Type>
</Property>
<Property>
    <Name>streamMonitorAudioStartTimeout</Name>
    <Value>0</Value>
    <Type>Integer</Type>
</Property>
<Property>
    <Name>streamMonitorAudioTimeout</Name>
    <Value>0</Value>
    <Type>Integer</Type>
</Property>

<!--
Monitor the incoming packet timecodes (audio, video or data) to be sure packets do not arrive
out of order or late. The
streamMonitor[type]TCPosTolerance and streamMonitor[type]TCNegTolerance (milliseconds) values
define a sliding window
based on the timecode of the previous packet. For example if these values are set to -500 and
3000 respectively then the
timecode difference between the current arriving packet and the previous packet of that type
must fall within
-500 and 3000 milliseconds. If not and streamMonitor[type]TCToleranceEnable is set to true

```

then stream will be considered  
unhealthy and will be reset.

-->

```
<Property>
    <Name>streamMonitorVideoTCToleranceEnable</Name>
    <Value>>false</Value>
    <Type>Boolean</Type>
</Property>
<Property>
    <Name>streamMonitorVideoTCPostTolerance</Name>
    <Value>3000</Value>
    <Type>Integer</Type>
</Property>
<Property>
    <Name>streamMonitorVideoTCNegTolerance</Name>
    <Value>-500</Value>
    <Type>Integer</Type>
</Property>
<Property>
    <Name>streamMonitorAudioTCToleranceEnable</Name>
    <Value>>false</Value>
    <Type>Boolean</Type>
</Property>
<Property>
    <Name>streamMonitorAudioTCPostTolerance</Name>
    <Value>3000</Value>
    <Type>Integer</Type>
</Property>
<Property>
    <Name>streamMonitorAudioTCNegTolerance</Name>
    <Value>-500</Value>
    <Type>Integer</Type>
</Property>
<Property>
    <Name>streamMonitorDataTCToleranceEnable</Name>
    <Value>>false</Value>
    <Type>Boolean</Type>
</Property>
<Property>
    <Name>streamMonitorDataTCPostTolerance</Name>
    <Value>3000</Value>
    <Type>Integer</Type>
</Property>
<Property>
    <Name>streamMonitorDataTCNegTolerance</Name>
    <Value>-500</Value>
    <Type>Integer</Type>
</Property>

<!--
```



Monitors the time difference between the audio and video channel of a stream. If the difference between the currently arriving video packet and the previous audio packet (or vice-versa) is greater than streamMonitorAVSyncTolerance and streamMonitorAVSyncToleranceEnable is set to true, then the stream will be considered unhealthy and will be reset.

-->

<Property>

<Name>streamMonitorAVSyncToleranceEnable</Name>

<Value>>false</Value>

<Type>Boolean</Type>

</Property>

<Property>

<Name>streamMonitorAVSyncTolerance</Name>

<Value>1500</Value>

<Type>Integer</Type>

</Property>

<!--

If set to true, then when a stream is reset and it belong to a MediaStreamNameGroup all streams in the group will be reset. If false only the unhealthy stream will be reset.

-->

<Property>

<Name>streamMonitorResetNameGroups</Name>

<Value>>true</Value>

<Type>Boolean</Type>

</Property>

<!--

Turns on debug logging of the monitoring.

-->

<Property>

<Name>streamMonitorDebug</Name>

<Value>>false</Value>

<Type>Boolean</Type>

</Property>

## Field Summary

protected	<a href="#">appInstance</a>
protected	<a href="#">badStreams</a>
protected	<a href="#">monitors</a>
protected	<a href="#">streamMonitorAudioStartTimeout</a>

protected	<a href="#">streamMonitorAudioTCNegTolerance</a>
protected	<a href="#">streamMonitorAudioTCPosTolerance</a>
protected	<a href="#">streamMonitorAudioTCToleranceEnable</a>
protected	<a href="#">streamMonitorAudioTimeout</a>
protected	<a href="#">streamMonitorAVSyncTolerance</a>
protected	<a href="#">streamMonitorAVSyncToleranceEnable</a>
protected	<a href="#">streamMonitorDataTCNegTolerance</a>
protected	<a href="#">streamMonitorDataTCPosTolerance</a>
protected	<a href="#">streamMonitorDataTCToleranceEnable</a>
protected	<a href="#">streamMonitorDebug</a>
protected	<a href="#">streamMonitorResetNameGroups</a>
protected	<a href="#">streamMonitorStreamStartTimeout</a>
protected	<a href="#">streamMonitorStreamTimeout</a>
protected	<a href="#">streamMonitorVideoStartTimeout</a>
protected	<a href="#">streamMonitorVideoTCNegTolerance</a>
protected	<a href="#">streamMonitorVideoTCPosTolerance</a>
protected	<a href="#">streamMonitorVideoTCToleranceEnable</a>
protected	<a href="#">streamMonitorVideoTimeout</a>

Fields inherited from class [com.wowza.wms.module.ModuleBase](#)

[CALLBACK\\_PARAM1](#), [CALLBACK\\_PARAM10](#), [CALLBACK\\_PARAM2](#), [CALLBACK\\_PARAM3](#), [CALLBACK\\_PARAM4](#), [CALLBACK\\_PARAM5](#), [CALLBACK\\_PARAM6](#), [CALLBACK\\_PARAM7](#), [CALLBACK\\_PARAM8](#), [CALLBACK\\_PARAM9](#), [PARAM1](#), [PARAM10](#), [PARAM2](#), [PARAM3](#), [PARAM4](#), [PARAM5](#), [PARAM6](#), [PARAM7](#), [PARAM8](#), [PARAM9](#), [PARAMMETHODNAME](#), [PLAYTRANSITION\\_APPEND](#), [PLAYTRANSITION\\_APPEND\\_IMMEDIATE](#), [PLAYTRANSITION\\_RESET](#), [PLAYTRANSITION\\_RESET\\_IMMEDIATE](#), [PLAYTRANSITION\\_STOP](#), [PLAYTRANSITION\\_SWAP](#), [PLAYTRANSITION\\_SWITCH](#), [PLAYTRANSITION\\_UNKNOWN](#), [PLAYTRANSITIONSTR\\_APPEND](#), [PLAYTRANSITIONSTR\\_RESET](#), [PLAYTRANSITIONSTR\\_STOP](#), [PLAYTRANSITIONSTR\\_SWAP](#), [PLAYTRANSITIONSTR\\_SWITCH](#), [PLAYTRANSITIONSTR\\_UNKNOWN](#)

## Constructor Summary

public	<a href="#">ModuleMediaCasterStreamMonitorAdvanced()</a>
--------	--

## Method Summary

void	<a href="#">onAppStart</a> ( <a href="#">IApplicationInstance</a> appInstance)
void	<a href="#">onAppStop</a> ( <a href="#">IApplicationInstance</a> appInstance)
boolean	<a href="#">onResetMediaCaster</a> ( <a href="#">IApplicationInstance</a> appInstance, <a href="#">IMediaCaster</a> mediaCaster)
void	<a href="#">onStreamCreate</a> ( <a href="#">IMediaStream</a> stream)
void	<a href="#">onStreamDestroy</a> ( <a href="#">IMediaStream</a> stream)
boolean	<a href="#">onValidateMediaCaster</a> ( <a href="#">IApplicationInstance</a> appInstance, <a href="#">IMediaCaster</a> mediaCaster)
void	<a href="#">onValidateMediaCastersStart</a> ( <a href="#">IApplicationInstance</a> appInstance)
void	<a href="#">onValidateMediaCastersStop</a> ( <a href="#">IApplicationInstance</a> appInstance)

### Methods inherited from class [com.wowza.wms.module.ModuleBase](#)

[getAppInstance](#), [getApplication](#), [getCallbackParamCount](#), [getLogger](#), [getParam](#), [getParamBoolean](#), [getParamBoolean](#), [getParamCount](#), [getParamDate](#), [getParamDouble](#), [getParamDouble](#), [getParamInt](#), [getParamInt](#), [getParamLong](#), [getParamLong](#), [getParamMixedArray](#), [getParamObj](#), [getParamString](#), [getParamString](#), [getParamType](#), [getStream](#), [getVHost](#), [invokePrevious](#), [invokePrevious](#), [isSendResult](#), [sendClientOnStatusError](#), [sendResult](#), [sendResult](#), [sendResult](#), [sendResult](#), [sendResult](#), [sendStreamOnStatusError](#)

### Methods inherited from class [java.lang.Object](#)

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

### Methods inherited from interface [com.wowza.wms.mediacaster.IMediaCasterValidateMediaCaster](#)

[onResetMediaCaster](#), [onValidateMediaCaster](#), [onValidateMediaCastersStart](#), [onValidateMediaCastersStop](#)

## Fields

### streamMonitorStreamStartTimeout

protected int **streamMonitorStreamStartTimeout**

(continued from last page)

---

**streamMonitorStreamTimeout**

protected int **streamMonitorStreamTimeout**

---

---

**streamMonitorVideoTCToleranceEnable**

protected boolean **streamMonitorVideoTCToleranceEnable**

---

---

**streamMonitorVideoTCPosTolerance**

protected int **streamMonitorVideoTCPosTolerance**

---

---

**streamMonitorVideoTCNegTolerance**

protected int **streamMonitorVideoTCNegTolerance**

---

---

**streamMonitorAudioTCToleranceEnable**

protected boolean **streamMonitorAudioTCToleranceEnable**

---

---

**streamMonitorAudioTCPosTolerance**

protected int **streamMonitorAudioTCPosTolerance**

---

---

**streamMonitorAudioTCNegTolerance**

protected int **streamMonitorAudioTCNegTolerance**

---

---

**streamMonitorDataTCToleranceEnable**

protected boolean **streamMonitorDataTCToleranceEnable**

---

---

**streamMonitorDataTCPosTolerance**

protected int **streamMonitorDataTCPosTolerance**

---

---

**streamMonitorDataTCNegTolerance**

protected int **streamMonitorDataTCNegTolerance**

---

(continued from last page)

---

**streamMonitorAVSyncToleranceEnable**protected boolean **streamMonitorAVSyncToleranceEnable**

---

**streamMonitorAVSyncTolerance**protected int **streamMonitorAVSyncTolerance**

---

**streamMonitorVideoStartTimeout**protected int **streamMonitorVideoStartTimeout**

---

**streamMonitorVideoTimeout**protected int **streamMonitorVideoTimeout**

---

**streamMonitorAudioStartTimeout**protected int **streamMonitorAudioStartTimeout**

---

**streamMonitorAudioTimeout**protected int **streamMonitorAudioTimeout**

---

**streamMonitorResetNameGroups**protected boolean **streamMonitorResetNameGroups**

---

**streamMonitorDebug**protected boolean **streamMonitorDebug**

---

**appInstance**protected com.wowza.wms.application.IApplicationInstance **appInstance**

---

(continued from last page)

## monitors

```
protected java.util.Map monitors
```

## badStreams

```
protected java.util.Set badStreams
```

## Constructors

### ModuleMediaCasterStreamMonitorAdvanced

```
public ModuleMediaCasterStreamMonitorAdvanced()
```

## Methods

### onAppStart

```
public void onAppStart(IApplicationInstance appInstance)
```

### onAppStop

```
public void onAppStop(IApplicationInstance appInstance)
```

### onStreamCreate

```
public void onStreamCreate(IMediaStream stream)
```

### onStreamDestroy

```
public void onStreamDestroy(IMediaStream stream)
```

### onValidateMediaCastersStart

```
public void onValidateMediaCastersStart(IApplicationInstance appInstance)
```

### onValidateMediaCaster

```
public boolean onValidateMediaCaster(IApplicationInstance appInstance,  
    IMediaCaster mediaCaster)
```

---

## onValidateMediaCastersStop

```
public void onValidateMediaCastersStop(IApplicationInstance appInstance)
```

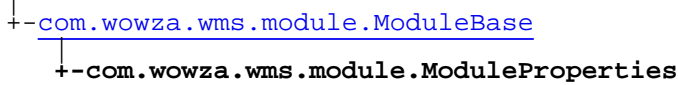
---

## onResetMediaCaster

```
public boolean onResetMediaCaster(IApplicationInstance appInstance,  
    IMediaCaster mediaCaster)
```

## com.wowza.wms.module Class ModuleProperties

java.lang.Object



public class **ModuleProperties**  
extends [ModuleBase](#)

### Fields inherited from class [com.wowza.wms.module.ModuleBase](#)

[CALLBACK\\_PARAM1](#), [CALLBACK\\_PARAM10](#), [CALLBACK\\_PARAM2](#), [CALLBACK\\_PARAM3](#), [CALLBACK\\_PARAM4](#), [CALLBACK\\_PARAM5](#), [CALLBACK\\_PARAM6](#), [CALLBACK\\_PARAM7](#), [CALLBACK\\_PARAM8](#), [CALLBACK\\_PARAM9](#), [PARAM1](#), [PARAM10](#), [PARAM2](#), [PARAM3](#), [PARAM4](#), [PARAM5](#), [PARAM6](#), [PARAM7](#), [PARAM8](#), [PARAM9](#), [PARAMMETHODNAME](#), [PLAYTRANSITION\\_APPEND](#), [PLAYTRANSITION\\_APPEND\\_IMMEDIATE](#), [PLAYTRANSITION\\_RESET](#), [PLAYTRANSITION\\_RESET\\_IMMEDIATE](#), [PLAYTRANSITION\\_STOP](#), [PLAYTRANSITION\\_SWAP](#), [PLAYTRANSITION\\_SWITCH](#), [PLAYTRANSITION\\_UNKNOWN](#), [PLAYTRANSITIONSTR\\_APPEND](#), [PLAYTRANSITIONSTR\\_RESET](#), [PLAYTRANSITIONSTR\\_STOP](#), [PLAYTRANSITIONSTR\\_SWAP](#), [PLAYTRANSITIONSTR\\_SWITCH](#), [PLAYTRANSITIONSTR\\_UNKNOWN](#)

### Constructor Summary

public	<a href="#">ModuleProperties</a> ( )
--------	--------------------------------------

### Method Summary

static void	<a href="#">getAppInstanceProperty</a> ( <a href="#">IClient</a> client, <a href="#">RequestFunction</a> function, <a href="#">AMFDataList</a> params) Get application instance level property value NetConnection.call("getAppInstanceProperty", resultObj, name);
static void	<a href="#">getApplicationProperty</a> ( <a href="#">IClient</a> client, <a href="#">RequestFunction</a> function, <a href="#">AMFDataList</a> params) Get application level property value NetConnection.call("getApplicationProperty", resultObj, name);
static void	<a href="#">getClientProperty</a> ( <a href="#">IClient</a> client, <a href="#">RequestFunction</a> function, <a href="#">AMFDataList</a> params) Get client level property value NetConnection.call("getClientProperty", resultObj, name);
static void	<a href="#">getStreamProperty</a> ( <a href="#">IClient</a> client, <a href="#">RequestFunction</a> function, <a href="#">AMFDataList</a> params) Get stream level property value NetConnection.call("getStreamProperty", resultObj, streamId, name);
static void	<a href="#">setAppInstanceProperty</a> ( <a href="#">IClient</a> client, <a href="#">RequestFunction</a> function, <a href="#">AMFDataList</a> params) Set application instance level property NetConnection.call("setAppInstanceProperty", null, name, value);
static void	<a href="#">setApplicationProperty</a> ( <a href="#">IClient</a> client, <a href="#">RequestFunction</a> function, <a href="#">AMFDataList</a> params) Set application level property NetConnection.call("setApplicationProperty", null, name, value);



static void	<a href="#">setClientProperty</a> ( <a href="#">IClient</a> client, <a href="#">RequestFunction</a> function, <a href="#">AMFDataList</a> params) Set client level property NetConnection.call("setClientProperty", null, name, value);
static void	<a href="#">setStreamProperty</a> ( <a href="#">IClient</a> client, <a href="#">RequestFunction</a> function, <a href="#">AMFDataList</a> params) Set stream level property NetConnection.call("setStreamProperty", null, streamId, name, value);

#### Methods inherited from class [com.wowza.wms.module.ModuleBase](#)

[getAppInstance](#), [getApplication](#), [getCallbackParamCount](#), [getLogger](#), [getParam](#),  
[getParamBoolean](#), [getParamBoolean](#), [getParamCount](#), [getParamDate](#), [getParamDouble](#),  
[getParamDouble](#), [getParamInt](#), [getParamInt](#), [getParamLong](#), [getParamLong](#),  
[getParamMixedArray](#), [getParamObj](#), [getParamString](#), [getParamString](#), [getParamType](#),  
[getStream](#), [getVHost](#), [invokePrevious](#), [invokePrevious](#), [isSendResult](#),  
[sendClientOnStatusError](#), [sendResult](#), [sendResult](#), [sendResult](#), [sendResult](#), [sendResult](#),  
[sendStreamOnStatusError](#)

#### Methods inherited from class [java.lang.Object](#)

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#),  
[wait](#)

## Constructors

### ModuleProperties

```
public ModuleProperties()
```

## Methods

### setApplicationProperty

```
public static void setApplicationProperty(IClient client,  

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,  

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,  
    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,  
    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,  
    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,  
    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,  
    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,  
    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.request**

## com.wowza.wms.request Class RequestFunction

java.lang.Object

└─com.wowza.wms.request.RequestFunction

public class **RequestFunction**  
extends Object

### Constructor Summary

public	<a href="#">RequestFunction()</a>
--------	-----------------------------------

### Method Summary

void	<a href="#">addData</a> (byte[] data, int offset, int size)
byte[]	<a href="#">getData</a> ()
java.nio.ByteBuffer	<a href="#">getDataByteBuffer</a> ()
java.nio.ByteBuffer	<a href="#">getExtraData</a> ()
int	<a href="#">getHeaderSize</a> ()
<a href="#">AMFDataList</a>	<a href="#">getMessage</a> ()
int	<a href="#">getMissing</a> ()
int	<a href="#">getSize</a> ()
int	<a href="#">getSrc</a> ()
long	<a href="#">getTimecode</a> ()
int	<a href="#">getType</a> ()
int	<a href="#">getWmsNumber</a> ()
boolean	<a href="#">isPartial</a> ()
void	<a href="#">setExtraData</a> (java.nio.ByteBuffer extraData)
void	<a href="#">setHeaderSize</a> (int headerSize)

void	<a href="#">setSize</a> (int size)
void	<a href="#">setSrc</a> (int src)
void	<a href="#">setTimecode</a> (long timecode)
void	<a href="#">setType</a> (int type)
void	<a href="#">setWmsNumber</a> (int wmsNumber)
String	<a href="#">toString</a> ()

**Methods inherited from class `java.lang.Object`**

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## Constructors

### RequestFunction

```
public RequestFunction()
```

## Methods

### getData

```
public byte[] getData()
```

### getDataByteBuffer

```
public java.nio.ByteBuffer getDataByteBuffer()
```

### getSize

```
public int getSize()
```

### getMissing

```
public int getMissing()
```

(continued from last page)

---

**setSize**

```
public void setSize(int size)
```

---

**getSrc**

```
public int getSrc()
```

---

**setSrc**

```
public void setSrc(int src)
```

---

**addData**

```
public void addData(byte[] data,  
                    int offset,  
                    int size)
```

---

**isPartial**

```
public boolean isPartial()
```

---

**getMessage**

```
public AMFDataList getMessage()
```

---

**toString**

```
public String toString()
```

---

**getType**

```
public int getType()
```

---

**setType**

```
public void setType(int type)
```

---

(continued from last page)

## getTimecode

```
public long getTimecode()
```

---

## setTimecode

```
public void setTimecode(long timecode)
```

---

## getWmsNumber

```
public int getWmsNumber()
```

---

## setWmsNumber

```
public void setWmsNumber(int wmsNumber)
```

---

## getExtraData

```
public java.nio.ByteBuffer getExtraData()
```

---

## setExtraData

```
public void setExtraData(java.nio.ByteBuffer extraData)
```

---

## getHeaderSize

```
public int getHeaderSize()
```

---

## setHeaderSize

```
public void setHeaderSize(int headerSize)
```

---



---

Package

**com.wowza.wms.response**

## com.wowza.wms.response Class ResponseFunction

java.lang.Object

└-com.wowza.wms.response.ResponseFunction

public class **ResponseFunction**  
extends Object

ResponseFunction: class for sending status messages to the client.

### Field Summary

public	<a href="#">doBreak</a>
--------	-------------------------

### Constructor Summary

public	<a href="#">ResponseFunction</a> ( <a href="#">IClient</a> client) Create an empty ResponseFunction
public	<a href="#">ResponseFunction</a> (com.wowza.wms.netconnection.INetConnection netConnection) Create an empty ResponseFunction
public	<a href="#">ResponseFunction</a> (com.wowza.wms.netconnection.INetConnection netConnection, <a href="#">AMFObj</a> amfObj) Create an empty ResponseFunction
public	<a href="#">ResponseFunction</a> ( <a href="#">IClient</a> client, <a href="#">AMFObj</a> amfObj) Create an empty ResponseFunction
public	<a href="#">ResponseFunction</a> ( <a href="#">IMediaStream</a> stream, <a href="#">AMFObj</a> amfObj) Create an empty ResponseFunction

### Method Summary

void	<a href="#">addBody</a> ( <a href="#">AMFData</a> body) Add AMFData to the body of the message.
void	<a href="#">addBytes</a> (byte[] inbytes) Add raw AMFData bytes[] to message body
void	<a href="#">createBroadcastMessage</a> (java.nio.ByteBuffer messageBytes) Create broadcast message.
void	<a href="#">createConnectMessage</a> (String inName, double inResultNum) Create net connection connect message.
void	<a href="#">createDefaultMessage</a> (String inName, double inResultNum) Create a default message (onStatus type messages).
void	<a href="#">createEnhancedSeekMessage</a> (byte[] messageBuffer, int src, int tc)

void	<a href="#"><u>createPlayStatusMessage</u></a> (String inName) Create onPlayStatus message.
void	<a href="#"><u>createSeekMessage</u></a> (String inName) Create a seek result message.
void	<a href="#"><u>createSOMessage</u></a> (byte[] messageBuffer, int objectEncoding) Create shared object message.
long	<a href="#"><u>getTimecode</u></a> () Get function timecode (milliseconds).
int	<a href="#"><u>getType</u></a> () Get message type.
boolean	<a href="#"><u>isForceAMF0</u></a> ()
void	<a href="#"><u>setBody</u></a> (int index, <a href="#"><u>AMFData</u></a> body) Add AMFData to the body of the message.
void	<a href="#"><u>setForceAMF0</u></a> (boolean forceAMF0)
void	<a href="#"><u>setMessageBytes</u></a> (byte[] messageBytes)
void	<a href="#"><u>setRetAMFNumber</u></a> (int innum) Set the return channel id
void	<a href="#"><u>setSrc</u></a> (int src) Set the src (stream id) for the message.
void	<a href="#"><u>setTimecode</u></a> (long timecode) Set function timecode (milliseconds).
void	<a href="#"><u>setType</u></a> (int type) Set message type.
int	<a href="#"><u>write</u></a> (java.io.OutputStream out, boolean isAbsTimecode, int chunkSize) Write message directly to OutputStream
int	<a href="#"><u>write</u></a> (java.io.OutputStream out, int chunkSize) Write message directly to OutputStream.

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Fields

### doBreak

public boolean **doBreak**

(continued from last page)

## Constructors

### ResponseFunction

```
public ResponseFunction(IClient client)
```

Create an empty ResponseFunction

**Parameters:**

client - client

---

### ResponseFunction

```
public ResponseFunction(com.wowza.wms.netconnection.INetConnection netConnection)
```

Create an empty ResponseFunction

**Parameters:**

netConnection - net connection

---

### ResponseFunction

```
public ResponseFunction(com.wowza.wms.netconnection.INetConnection netConnection,  
AMFObj amfObj)
```

Create an empty ResponseFunction

**Parameters:**

netConnection - net connection

amfObj - amf object

---

### ResponseFunction

```
public ResponseFunction(IClient client,  
AMFObj amfObj)
```

Create an empty ResponseFunction

**Parameters:**

client - client object

amfObj - amf object

---

### ResponseFunction

```
public ResponseFunction(IMediaStream stream,  
AMFObj amfObj)
```

Create an empty ResponseFunction

**Parameters:**

stream - media stream object

amfObj - amf object

## Methods

### setSrc

```
public void setSrc(int src)
```

(continued from last page)

Set the src (stream id) for the message.

**Parameters:**

src - src (stream id)

---

## createDefaultMessage

```
public void createDefaultMessage(String inName,  
    double inResultNum)
```

Create a default message (onStatus type messages).

**Parameters:**

inName - handler name (example: onStatus)

inResultNum - result number of 0 if not result

---

## createConnectMessage

```
public void createConnectMessage(String inName,  
    double inResultNum)
```

Create net connection connect message.

**Parameters:**

inName - handler name (example: connection)

inResultNum - result number if 0 not a result

---

## createSeekMessage

```
public void createSeekMessage(String inName)
```

Create a seek result message.

**Parameters:**

inName - handler name (example: onStatus)

---

## createPlayStatusMessage

```
public void createPlayStatusMessage(String inName)
```

Create onPlayStatus message.

**Parameters:**

inName - handler name (example: onPlayStatus)

---

## setMessageBytes

```
public void setMessageBytes(byte[] messageBytes)
```

---

## createBroadcastMessage

```
public void createBroadcastMessage(java.nio.ByteBuffer messageBytes)
```

Create broadcast message. Used for ByteBuffer handler calls.

**Parameters:**

(continued from last page)

messageBytes - ByteBuffer with raw AMFData bytes to be sent to client

---

## createSOMessage

```
public void createSOMessage(byte[] messageBuffer,  
    int objectEncoding)
```

Create shared object message. Used for ByteBuffer handler calls.

**Parameters:**

messageBuffer

---

## createEnhancedSeekMessage

```
public void createEnhancedSeekMessage(byte[] messageBuffer,  
    int src,  
    int tc)
```

---

## addBody

```
public void addBody(AMFData body)
```

Add AMFData to the body of the message.

**Parameters:**

body - AMFData message

---

## setBody

```
public void setBody(int index,  
    AMFData body)
```

Add AMFData to the body of the message.

**Parameters:**

index - index in body list

body - AMFData message

---

## addBytes

```
public void addBytes(byte[] inbytes)
```

Add raw AMFData bytes[] to message body

**Parameters:**

inbytes - raw AMFData bytes[]

---

## setRetAMFNumber

```
public void setRetAMFNumber(int innum)
```

Set the return channel id

**Parameters:**

innum - return channel id

(continued from last page)

## getTimecode

```
public long getTimecode()
```

Get function timecode (milliseconds).

**Returns:**

function timecode (milliseconds)

---

## setTimecode

```
public void setTimecode(long timecode)
```

Set function timecode (milliseconds).

**Parameters:**

timecode - function timecode (milliseconds)

---

## setType

```
public void setType(int type)
```

Set message type. See IVHost.CONTENTTYPE\_\*

**Parameters:**

type - message type

---

## getType

```
public int getType()
```

Get message type. See IVHost.CONTENTTYPE\_\*

**Returns:**

message type

---

## write

```
public int write(java.io.OutputStream out,  
                int chunkSize)
```

Write message directly to OutputStream.

**Parameters:**

out - OutputStream

**Returns:**

number of bytes written

---

## isForceAMF0

```
public boolean isForceAMF0()
```

---

## setForceAMF0

```
public void setForceAMF0(boolean forceAMF0)
```

(continued from last page)

---

**write**

```
public int write(java.io.OutputStream out,  
                boolean isAbsTimecode,  
                int chunkSize)
```

Write message directly to OutputStream

**Parameters:**

out - OutputStream  
isAbsTimecode - is timecode absolute

**Returns:**

number of bytes written



## com.wowza.wms.response Class ResponseFunctions

java.lang.Object

└─com.wowza.wms.response.ResponseFunctions

public class **ResponseFunctions**  
extends Object

ResponseFunctions: collection of ResponseFunction objects. This interface is used to asynchronously collect up a set of client responses that will be sent to the client at the next opportunity.

### Constructor Summary

public	<a href="#">ResponseFunctions()</a>
--------	-------------------------------------

### Method Summary

void	<a href="#">add(ResponseFunction wmsResponseFunction)</a> Add a function.
void	<a href="#">clear()</a>
boolean	<a href="#">isPending()</a> Are there any pending items in the list.
int	<a href="#">output(java.io.OutputStream out, int sendChunkSize)</a> Write all functions (in order added to list) to OutputStream.

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructors

### ResponseFunctions

public **ResponseFunctions()**

## Methods

### add

public void **add**([ResponseFunction wmsResponseFunction](#))

Add a function.

(continued from last page)

**Parameters:**wmsResponseFunction - function

---

**clear**

```
public void clear()
```

---

**isPending**

```
public boolean isPending()
```

Are there any pending items in the list.

**Returns:**

true if items in list

---

**output**

```
public int output(java.io.OutputStream out,  
int sendChunkSize)
```

Write all functions (in order added to list) to OutputStream.

**Parameters:**

out - OutputStream

**Returns:**

total number of bytes written

---

---

Package

**com.wowza.wms.rtp.model**

---

## com.wowza.wms.rtp.model Interface IRTTPMessageHandler

---

public interface **IRTTPMessageHandler**  
extends

---

### Method Summary

void	<a href="#">handleMessage</a> (java.net.SocketAddress socketAddr, byte[] buffer, int offset, int len)
------	---

---

### Methods

#### **handleMessage**

```
public void handleMessage(java.net.SocketAddress socketAddr,  
    byte[] buffer,  
    int offset,  
    int len)
```

com.wowza.wms.rtp.model

# Interface IRTPMetadataProvider

public interface **IRTPMetadataProvider**  
extends

Method Summary	
byte[]	<a href="#">getMetadataPacket</a> (RTPStream stream)

## Methods

**getMetadataPacket**

public byte[] **getMetadataPacket**(RTPStream stream)

com.wowza.wms.rtp.model

**Interface IRTPSessionNotify**

public interface **IRTPSessionNotify**  
extends

Method Summary	
void	<a href="#">onRTPSessionCreate</a> ( <a href="#">RTPSession</a> rtpSession)
void	<a href="#">onRTPSessionDestroy</a> ( <a href="#">RTPSession</a> rtpSession)

Methods

**onRTPSessionCreate**

public void **onRTPSessionCreate**([RTPSession](#) rtpSession)

**onRTPSessionDestroy**

public void **onRTPSessionDestroy**([RTPSession](#) rtpSession)

## com.wowza.wms.rtp.model Interface IRTSPActionNotify

public interface IRTSPActionNotify  
extends

### Method Summary

void	<a href="#">onAnnounce(RTPSession</a> rtspSession, <a href="#">com.wowza.wms.rtsp.RTSPRequestMessage</a> req, <a href="#">com.wowza.wms.rtsp.RTSPResponseMessages</a> resp)
void	<a href="#">onDescribe(RTPSession</a> rtspSession, <a href="#">com.wowza.wms.rtsp.RTSPRequestMessage</a> req, <a href="#">com.wowza.wms.rtsp.RTSPResponseMessages</a> resp)
void	<a href="#">onGetParameter(RTPSession</a> rtspSession, <a href="#">com.wowza.wms.rtsp.RTSPRequestMessage</a> req, <a href="#">com.wowza.wms.rtsp.RTSPResponseMessages</a> resp)
void	<a href="#">onOptions(RTPSession</a> rtspSession, <a href="#">com.wowza.wms.rtsp.RTSPRequestMessage</a> req, <a href="#">com.wowza.wms.rtsp.RTSPResponseMessages</a> resp)
void	<a href="#">onPause(RTPSession</a> rtspSession, <a href="#">com.wowza.wms.rtsp.RTSPRequestMessage</a> req, <a href="#">com.wowza.wms.rtsp.RTSPResponseMessages</a> resp)
void	<a href="#">onPlay(RTPSession</a> rtspSession, <a href="#">com.wowza.wms.rtsp.RTSPRequestMessage</a> req, <a href="#">com.wowza.wms.rtsp.RTSPResponseMessages</a> resp)
void	<a href="#">onRecord(RTPSession</a> rtspSession, <a href="#">com.wowza.wms.rtsp.RTSPRequestMessage</a> req, <a href="#">com.wowza.wms.rtsp.RTSPResponseMessages</a> resp)
void	<a href="#">onRedirect(RTPSession</a> rtspSession, <a href="#">com.wowza.wms.rtsp.RTSPRequestMessage</a> req, <a href="#">com.wowza.wms.rtsp.RTSPResponseMessages</a> resp)
void	<a href="#">onSetParameter(RTPSession</a> rtspSession, <a href="#">com.wowza.wms.rtsp.RTSPRequestMessage</a> req, <a href="#">com.wowza.wms.rtsp.RTSPResponseMessages</a> resp)
void	<a href="#">onSetup(RTPSession</a> rtspSession, <a href="#">com.wowza.wms.rtsp.RTSPRequestMessage</a> req, <a href="#">com.wowza.wms.rtsp.RTSPResponseMessages</a> resp)
void	<a href="#">onTeardown(RTPSession</a> rtspSession, <a href="#">com.wowza.wms.rtsp.RTSPRequestMessage</a> req, <a href="#">com.wowza.wms.rtsp.RTSPResponseMessages</a> resp)

### Methods

(continued from last page)

---

## onDescribe

```
public void onDescribe(RTPSession rtspSession,  
    com.wowza.wms.rtp.RTSPRequestMessage req,  
    com.wowza.wms.rtp.RTSPResponseMessages resp)
```

---

## onAnnounce

```
public void onAnnounce(RTPSession rtspSession,  
    com.wowza.wms.rtp.RTSPRequestMessage req,  
    com.wowza.wms.rtp.RTSPResponseMessages resp)
```

---

## onGetParameter

```
public void onGetParameter(RTPSession rtspSession,  
    com.wowza.wms.rtp.RTSPRequestMessage req,  
    com.wowza.wms.rtp.RTSPResponseMessages resp)
```

---

## onSetParameter

```
public void onSetParameter(RTPSession rtspSession,  
    com.wowza.wms.rtp.RTSPRequestMessage req,  
    com.wowza.wms.rtp.RTSPResponseMessages resp)
```

---

## onOptions

```
public void onOptions(RTPSession rtspSession,  
    com.wowza.wms.rtp.RTSPRequestMessage req,  
    com.wowza.wms.rtp.RTSPResponseMessages resp)
```

---

## onPause

```
public void onPause(RTPSession rtspSession,  
    com.wowza.wms.rtp.RTSPRequestMessage req,  
    com.wowza.wms.rtp.RTSPResponseMessages resp)
```

---

## onPlay

```
public void onPlay(RTPSession rtspSession,  
    com.wowza.wms.rtp.RTSPRequestMessage req,  
    com.wowza.wms.rtp.RTSPResponseMessages resp)
```

---



(continued from last page)

## onRecord

```
public void onRecord(RTSPSession rtspSession,  
    com.wowza.wms.rtsp.RTSPRequestMessage req,  
    com.wowza.wms.rtsp.RTSPResponseMessages resp)
```

---

## onRedirect

```
public void onRedirect(RTSPSession rtspSession,  
    com.wowza.wms.rtsp.RTSPRequestMessage req,  
    com.wowza.wms.rtsp.RTSPResponseMessages resp)
```

---

## onSetup

```
public void onSetup(RTSPSession rtspSession,  
    com.wowza.wms.rtsp.RTSPRequestMessage req,  
    com.wowza.wms.rtsp.RTSPResponseMessages resp)
```

---

## onTeardown

```
public void onTeardown(RTSPSession rtspSession,  
    com.wowza.wms.rtsp.RTSPRequestMessage req,  
    com.wowza.wms.rtsp.RTSPResponseMessages resp)
```

## com.wowza.wms.rtp.model Class RTPPushPublishSession

java.lang.Object

└─com.wowza.wms.rtp.model.RTPPushPublishSession

public class **RTPPushPublishSession**  
extends Object

### Constructor Summary

public	<a href="#">RTPPushPublishSession()</a>
--------	---

### Method Summary

<a href="#">RTPSession</a>	<a href="#">getRTPSession()</a>
String	<a href="#">getSDPData()</a>
void	<a href="#">setRTPSession(<a href="#">RTPSession</a> rtpSession)</a>
void	<a href="#">setSDPData(String sdpData)</a>

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Constructors

#### **RTPPushPublishSession**

public **RTPPushPublishSession()**

### Methods

#### **getSDPData**

public String **getSDPData()**

#### **setSDPData**

public void **setSDPData**(String sdpData)

(continued from last page)

---

## getRTPSession

```
public RTPSession getRTPSession()
```

---

## setRTPSession

```
public void setRTPSession(RTPSession rtpSession)
```

## com.wowza.wms.rtp.model Class RTPSession

java.lang.Object

└-com.wowza.wms.rtp.model.RTPSession

public class **RTPSession**  
extends Object

### Field Summary

protected	<a href="#">actionListeners</a>
protected	<a href="#">appInstance</a>
protected	<a href="#">appInstanceShutdown</a>
protected	<a href="#">authenticatePlayHandler</a>
protected	<a href="#">authenticatePublishHandler</a>
public static final	<a href="#">AUTHMETHOD_PLAY</a> Value: <b>1</b>
public static final	<a href="#">AUTHMETHOD_PUBLISH</a> Value: <b>2</b>
public static final	<a href="#">AUTHMETHOD_UNKNOWN</a> Value: <b>0</b>
protected	<a href="#">connectionHolder</a>
protected	<a href="#">cookieStr</a>
protected	<a href="#">debugRTSPSession</a>
protected	<a href="#">elapsedTime</a>
protected	<a href="#">idleFrequency</a>
protected	<a href="#">idleHandler</a>
protected	<a href="#">ioPerformanceCounter</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="#">loggedConnect</a>
protected	<a href="#">properties</a>
protected	<a href="#">queryStr</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="#">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>
-----------	-----------------------

## Constructor Summary

public	<a href="#">RTPSession</a> (String sessionId)
--------	---

## Method Summary

void	<a href="#">acceptSession</a> ()
void	<a href="#">addActionListener</a> ( <a href="#">IRTSPActionNotify</a> actionListener)
void	<a href="#">addIOPerformance</a> ( <a href="#">IOPerformanceCounter</a> totalIOPerformanceResult)
void	<a href="#">addIOPerformance2</a> ( <a href="#">IOPerformanceCounter</a> totalIOPerformanceResult)
void	<a href="#">addRTSPStream</a> (RTPStream stream)
void	<a href="#">clearLoggingValues</a> ()
<a href="#">IApplicationInstance</a>	<a href="#">getAppInstance</a> ()
<a href="#">IAuthenticateRTSP</a>	<a href="#">getAuthenticatePlayHandler</a> ()
<a href="#">IAuthenticateRTSP</a>	<a href="#">getAuthenticatePublishHandler</a> ()
ConnectionHolder	<a href="#">getConnectionHolder</a> ()
String	<a href="#">getCookieStr</a> ()
int	<a href="#">getIdleFrequency</a> ()
RTPIdleHandler	<a href="#">getIdleHandler</a> ()
<a href="#">IOPerformanceCounter</a>	<a href="#">getIOPerformanceCounter</a> ()
String	<a href="#">getIp</a> ()
int	<a href="#">getLastAuthenticateMethod</a> ()
<a href="#">WMSProperties</a>	<a href="#">getProperties</a> ()
String	<a href="#">getQueryStr</a> ()
String	<a href="#">getReferrer</a> ()
RTPWriteListener	<a href="#">getRTPWriteListener</a> ()

double	<a href="#"><u>getRTSPPlayRangeStart()</u></a>
double	<a href="#"><u>getRTSPPlayRangeStop()</u></a>
RTPStream	<a href="#"><u>getRTSPStream()</u></a>
RTPStream	<a href="#"><u>getRTSPStream</u></a> (String streamId)
String	<a href="#"><u>getRTSPTunnelingSessionId()</u></a>
String	<a href="#"><u>getServerIp()</u></a>
int	<a href="#"><u>getServerPort()</u></a>
String	<a href="#"><u>getSessionId()</u></a>
String	<a href="#"><u>getUri()</u></a>
String	<a href="#"><u>getUserAgent()</u></a>
<a href="#"><u>IVHost</u></a>	<a href="#"><u>getVHost()</u></a>
boolean	<a href="#"><u>isAnnounce()</u></a>
boolean	<a href="#"><u>isAnnounceOrDescribe()</u></a>
boolean	<a href="#"><u>isConnected()</u></a>
boolean	<a href="#"><u>isDebugRTSPSession()</u></a>
boolean	<a href="#"><u>isDescribe()</u></a>
boolean	<a href="#"><u>isLoggedConnect()</u></a>
boolean	<a href="#"><u>isSessionValid()</u></a>
boolean	<a href="#"><u>isShutdownClient()</u></a>
void	<a href="#"><u>onAnnounce</u></a> (com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp)
void	<a href="#"><u>onDescribe</u></a> (com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp)
void	<a href="#"><u>onGetParameter</u></a> (com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp)
void	<a href="#"><u>onOptions</u></a> (com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp)

void	<a href="#"><u>onPause</u></a> (com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp)
void	<a href="#"><u>onPlay</u></a> (com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp)
void	<a href="#"><u>onRecord</u></a> (com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp)
void	<a href="#"><u>onRedirect</u></a> (com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp)
void	<a href="#"><u>onSetParameter</u></a> (com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp)
void	<a href="#"><u>onSetup</u></a> (com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp)
void	<a href="#"><u>onTeardown</u></a> (com.wowza.wms.rtsp.RTSPRequestMessage req, com.wowza.wms.rtsp.RTSPResponseMessages resp)
void	<a href="#"><u>rejectSession</u></a> ()
void	<a href="#"><u>removeActionListener</u></a> ( <a href="#"><u>IRTSPActionNotify</u></a> actionListener)
RTPStream	<a href="#"><u>removeRTSPStream</u></a> (String streamId)
void	<a href="#"><u>setAnnounce</u></a> (boolean isAnnounce)
void	<a href="#"><u>setAppInstance</u></a> ( <a href="#"><u>IApplicationInstance</u></a> appInstance)
void	<a href="#"><u>setAuthenticatePlayHandler</u></a> ( <a href="#"><u>IAuthenticateRTSP</u></a> authenticatePlayHandler)
void	<a href="#"><u>setAuthenticatePublishHandler</u></a> ( <a href="#"><u>IAuthenticateRTSP</u></a> authenticatePublishHandler)
void	<a href="#"><u>setConnected</u></a> (boolean isConnected)
void	<a href="#"><u>setCookieStr</u></a> (String cookieStr)
void	<a href="#"><u>setDebugRTSPSession</u></a> (boolean debugRTSPSession)
void	<a href="#"><u>setDescribe</u></a> (boolean isDescribe)
void	<a href="#"><u>setIdleFrequency</u></a> (int idleFrequency)
void	<a href="#"><u>setIdleHandler</u></a> (RTPIIdleHandler idleHandler)
void	<a href="#"><u>setIOPerformanceCounter</u></a> ( <a href="#"><u>IOPerformanceCounter</u></a> ioPerformanceCounter)
void	<a href="#"><u>setIp</u></a> (String ip)



void	<a href="#"><u>setLastAuthenticateMethod</u></a> (int lastAuthenticateMethod)
void	<a href="#"><u>setLoggedConnect</u></a> (boolean loggedConnect)
void	<a href="#"><u>setQueryStr</u></a> (String queryStr)
void	<a href="#"><u>setReferrer</u></a> (String referrer)
void	<a href="#"><u>setRTSPPlayRangeStart</u></a> (double rtspPlayRangeStart)
void	<a href="#"><u>setRTSPPlayRangeStop</u></a> (double rtspPlayRangeStop)
void	<a href="#"><u>setRTSPTunnelingSessionId</u></a> (String rtspTunnelingSessionId)
void	<a href="#"><u>setServerIp</u></a> (String serverIp)
void	<a href="#"><u>setServerPort</u></a> (int serverPort)
void	<a href="#"><u>setSessionId</u></a> (String sessionId)
void	<a href="#"><u>setSessionValid</u></a> (boolean isSessionValid)
void	<a href="#"><u>setShutdownClient</u></a> (boolean shutdownClient)
void	<a href="#"><u>setUri</u></a> (String uri)
void	<a href="#"><u>setUserAgent</u></a> (String userAgent)
void	<a href="#"><u>setVHost</u></a> ( <a href="#"><u>IVHost</u></a> vhost)
void	<a href="#"><u>shutdown</u></a> ()
void	<a href="#"><u>shutdown</u></a> (RTPRequestStatus status)
void	<a href="#"><u>touch</u></a> ()
void	<a href="#"><u>updateLoggingValues</u></a> ()

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Fields

(continued from last page)

---

## 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
```

---

## vhost

```
protected com.wowza.wms.vhost.IVHost vhost
```

---

## userAgent

```
protected java.lang.String userAgent
```

---

(continued from last page)

**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`

---

**loggedConnect**`protected boolean loggedConnect`

---

**properties**`protected com.wowza.wms.application.WMSProperties properties`

---

(continued from last page)

---

**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**

---

**connectionHolder**protected com.wowza.wms.client.ConnectionHolder **connectionHolder**

---

**isConnected**protected boolean **isConnected**

---

(continued from last page)

## idleHandler

protected com.wowza.wms.rtp.model.RTPIdleHandler **idleHandler**

---

## serverIp

protected java.lang.String **serverIp**

---

## serverPort

protected int **serverPort**

---

## uri

protected java.lang.String **uri**

---

## referrer

protected java.lang.String **referrer**

---

## queryStr

protected java.lang.String **queryStr**

---

## actionListeners

protected java.util.List **actionListeners**

---

## elapsedTime

protected com.wowza.util.ElapsedTimer **elapsedTime**

---

## idleFrequency

protected int **idleFrequency**

---

## debugRTSPSession

protected boolean **debugRTSPSession**

---

(continued from last page)

---

## rtspPlayRangeStart

protected double **rtspPlayRangeStart**

---

## rtspPlayRangeStop

protected double **rtspPlayRangeStop**

---

## shutdownClient

protected boolean **shutdownClient**

---

## Constructors

### RTPSession

public **RTPSession**(String sessionId)

## Methods

### acceptSession

public void **acceptSession**()

---

### rejectSession

public void **rejectSession**()

---

### getSessionId

public String **getSessionId**()

---

### setSessionId

public void **setSessionId**(String sessionId)

---

### addRTSPStream

public void **addRTSPStream**(RTPStream stream)

---

(continued from last page)

---

**removeRTSPStream**

```
public RTPStream removeRTSPStream(String streamId)
```

---

**getRTSPStream**

```
public RTPStream getRTSPStream(String streamId)
```

---

**getRTSPStream**

```
public RTPStream getRTSPStream()
```

---

**getVHost**

```
public IVHost getVHost()
```

---

**setVHost**

```
public void setVHost(IVHost vhost)
```

---

**getUserAgent**

```
public String getUserAgent()
```

---

**setUserAgent**

```
public void setUserAgent(String userAgent)
```

---

**touch**

```
public void touch()
```

---

**shutdown**

```
public void shutdown()
```

---

(continued from last page)

---

## shutdown

```
public void shutdown(RTPRequestStatus status)
```

---

## isSessionValid

```
public boolean isSessionValid()
```

---

## setSessionValid

```
public void setSessionValid(boolean isSessionValid)
```

---

## getAuthenticatePublishHandler

```
public IAuthenticateRTSP getAuthenticatePublishHandler()
```

---

## setAuthenticatePublishHandler

```
public void setAuthenticatePublishHandler(IAuthenticateRTSP  
authenticatePublishHandler)
```

---

## getAuthenticatePlayHandler

```
public IAuthenticateRTSP getAuthenticatePlayHandler()
```

---

## setAuthenticatePlayHandler

```
public void setAuthenticatePlayHandler(IAuthenticateRTSP authenticatePlayHandler)
```

---

## getLastAuthenticateMethod

```
public int getLastAuthenticateMethod()
```

---

## setLastAuthenticateMethod

```
public void setLastAuthenticateMethod(int lastAuthenticateMethod)
```

---

## getAppInstance

```
public IApplicationInstance getAppInstance()
```

---



(continued from last page)

---

## setAppInstance

```
public void setAppInstance(IApplicationInstance appInstance)
```

---

---

## isLoggedConnect

```
public boolean isLoggedConnect()
```

---

---

## setLoggedConnect

```
public void setLoggedConnect(boolean loggedConnect)
```

---

---

## getIp

```
public String getIp()
```

---

---

## setIp

```
public void setIp(String ip)
```

---

---

## getProperties

```
public WMSProperties getProperties()
```

---

---

## isDescribe

```
public boolean isDescribe()
```

---

---

## setDescribe

```
public void setDescribe(boolean isDescribe)
```

---

---

## isAnnounce

```
public boolean isAnnounce()
```

---

(continued from last page)

---

## setAnnounce

```
public void setAnnounce(boolean isAnnounce)
```

---

## isAnnounceOrDescribe

```
public boolean isAnnounceOrDescribe()
```

---

## getRTPWriteListener

```
public RTPWriteListener getRTPWriteListener()
```

---

## addIOPerformance

```
public void addIOPerformance(IOPerformanceCounter totalIOPerformanceResult)
```

---

## addIOPerformance2

```
public void addIOPerformance2(IOPerformanceCounter totalIOPerformanceResult)
```

---

## getConnectionHolder

```
public ConnectionHolder getConnectionHolder()
```

---

## isConnected

```
public boolean isConnected()
```

---

## setConnected

```
public void setConnected(boolean isConnected)
```

---

## getIdleHandler

```
public RTPIdleHandler getIdleHandler()
```

---

## setIdleHandler

```
public void setIdleHandler(RTPIdleHandler idleHandler)
```

---

(continued from last page)

---

**getServerIp**

```
public String getServerIp()
```

---

**setServerIp**

```
public void setServerIp(String serverIp)
```

---

**getServerPort**

```
public int getServerPort()
```

---

**setServerPort**

```
public void setServerPort(int serverPort)
```

---

**getUri**

```
public String getUri()
```

---

**setUri**

```
public void setUri(String uri)
```

---

**getReferrer**

```
public String getReferrer()
```

---

**setReferrer**

```
public void setReferrer(String referrer)
```

---

**getQueryStr**

```
public String getQueryStr()
```

---

(continued from last page)

---

## setQueryStr

```
public void setQueryStr(String queryStr)
```

---

## updateLoggingValues

```
public void updateLoggingValues()
```

---

## clearLoggingValues

```
public void clearLoggingValues()
```

---

## addActionListener

```
public void addActionListener(IRTSPActionNotify actionListener)
```

---

## removeActionListener

```
public void removeActionListener(IRTSPActionNotify actionListener)
```

---

## onDescribe

```
public void onDescribe(com.wowza.wms.rtsp.RTSPRequestMessage req,  
    com.wowza.wms.rtsp.RTSPResponseMessages resp)
```

---

## onAnnounce

```
public void onAnnounce(com.wowza.wms.rtsp.RTSPRequestMessage req,  
    com.wowza.wms.rtsp.RTSPResponseMessages resp)
```

---

## onGetParameter

```
public void onGetParameter(com.wowza.wms.rtsp.RTSPRequestMessage req,  
    com.wowza.wms.rtsp.RTSPResponseMessages resp)
```

---

## onSetParameter

```
public void onSetParameter(com.wowza.wms.rtsp.RTSPRequestMessage req,  
    com.wowza.wms.rtsp.RTSPResponseMessages resp)
```

---

(continued from last page)

## onOptions

```
public void onOptions(com.wowza.wms.rtsp.RTSPRequestMessage req,  
    com.wowza.wms.rtsp.RTSPResponseMessages resp)
```

---

## onPause

```
public void onPause(com.wowza.wms.rtsp.RTSPRequestMessage req,  
    com.wowza.wms.rtsp.RTSPResponseMessages resp)
```

---

## onPlay

```
public void onPlay(com.wowza.wms.rtsp.RTSPRequestMessage req,  
    com.wowza.wms.rtsp.RTSPResponseMessages resp)
```

---

## onRecord

```
public void onRecord(com.wowza.wms.rtsp.RTSPRequestMessage req,  
    com.wowza.wms.rtsp.RTSPResponseMessages resp)
```

---

## onRedirect

```
public void onRedirect(com.wowza.wms.rtsp.RTSPRequestMessage req,  
    com.wowza.wms.rtsp.RTSPResponseMessages resp)
```

---

## onSetup

```
public void onSetup(com.wowza.wms.rtsp.RTSPRequestMessage req,  
    com.wowza.wms.rtsp.RTSPResponseMessages resp)
```

---

## onTeardown

```
public void onTeardown(com.wowza.wms.rtsp.RTSPRequestMessage req,  
    com.wowza.wms.rtsp.RTSPResponseMessages resp)
```

---

## getIdleFrequency

```
public int getIdleFrequency()
```

---

## setIdleFrequency

```
public void setIdleFrequency(int idleFrequency)
```

---

(continued from last page)

---

### isDebugRTSPSession

```
public boolean isDebugRTSPSession()
```

---

### setDebugRTSPSession

```
public void setDebugRTSPSession(boolean debugRTSPSession)
```

---

### getRTSPTunnelingSessionId

```
public String getRTSPTunnelingSessionId()
```

---

### setRTSPTunnelingSessionId

```
public void setRTSPTunnelingSessionId(String rtspTunnelingSessionId)
```

---

### getRTSPPlayRangeStart

```
public double getRTSPPlayRangeStart()
```

---

### setRTSPPlayRangeStart

```
public void setRTSPPlayRangeStart(double rtspPlayRangeStart)
```

---

### getRTSPPlayRangeStop

```
public double getRTSPPlayRangeStop()
```

---

### setRTSPPlayRangeStop

```
public void setRTSPPlayRangeStop(double rtspPlayRangeStop)
```

---

### getCookieStr

```
public String getCookieStr()
```

---

(continued from last page)

**setCookieStr**

```
public void setCookieStr(String cookieStr)
```

---

**getIOPerformanceCounter**

```
public IOPerformanceCounter getIOPerformanceCounter()
```

---

**setIOPerformanceCounter**

```
public void setIOPerformanceCounter(IOPerformanceCounter ioPerformanceCounter)
```

---

**isShutdownClient**

```
public boolean isShutdownClient()
```

---

**setShutdownClient**

```
public void setShutdownClient(boolean shutdownClient)
```

---

---

Package

**com.wowza.wms.server**



---

## com.wowza.wms.server Interface IResponseListener

---

public interface **IResponseListener**  
extends

---

### Method Summary

void	<a href="#"><u>onResponseWriteStart</u></a> (RtmpResponseMessage response)
void	<a href="#"><u>onResponseWriteStop</u></a> (RtmpResponseMessage response)

---

### Methods

#### onResponseWriteStart

public void **onResponseWriteStart**(RtmpResponseMessage response)

---

#### onResponseWriteStop

public void **onResponseWriteStop**(RtmpResponseMessage response)

## com.wowza.wms.server Interface IServer

All Known Implementing Classes:  
[Server](#)

public interface **IServer**  
extends

IServer: public interface to Server object.

### Method Summary

void	<a href="#">addServerListener</a> ( <a href="#">IServerNotify</a> serverListener) Add server listener
java.util.List	<a href="#">getAdminInterfaceObjectList</a> () Get the list of objects exposed through JMX interface
RandomIdGenerator	<a href="#">getClientIdGenerator</a> () Get the client id generator for the server
<a href="#">HostPort</a>	<a href="#">getCommandInterfaceHostPort</a> () Get the definition of the command interface
<a href="#">ConnectionCounter</a>	<a href="#">getConnectionCounter</a> () Get the server connection counter.
ConnectionCounterSimple	<a href="#">getConnectionCounter</a> (int counterIndex) Get the server connection counter for a specific technology (see IVHost.COUNTER_*)
int	<a href="#">getCoreHandlerPoolSize</a> () Get the handler core thread pool size.
int	<a href="#">getCoreTransportPoolSize</a> () Get the transport core thread pool size.
String	<a href="#">getDateStarted</a> () Get the date and time the server was started.
java.util.Properties	<a href="#">getDynamicLogProperties</a> () Get the dynamic log properties defined at the server level in conf/log4j.properties
<a href="#">ThreadPool</a>	<a href="#">getHandlerThreadPool</a> () Get the server handler thread pool.
<a href="#">IOPerformanceCounter</a>	<a href="#">getIoPerformanceCounter</a> () Get the server performance counter.
<a href="#">IOPerformanceCounter</a>	<a href="#">getIoPerformanceCounter</a> (int counterIndex) Get the server performance counter for a specific technology (see IVHost.COUNTER_*)
<a href="#">WMSProperties</a>	<a href="#">getProperties</a> () Get server level properties collection

<a href="#">ThreadPool</a>	<a href="#">getThreadPool()</a> Get the server handler thread pool.
String	<a href="#">getTimeRunning()</a> Get a formatted String of how long the server has been running.
double	<a href="#">getTimeRunningSeconds()</a> Get time running in seconds
<a href="#">ThreadPool</a>	<a href="#">getTransportThreadPool()</a> Get the server transport thread pool.
com.wowza.wms.transport.udp.UDPPortManager	<a href="#">getUDPPortManager()</a> Get the UDP port manager which manages the allocation of incoming UDP port binding to be sure there are not port conflicts
String[]	<a href="#">getUserAgents()</a> Get a pipe " " delimited list of user agents that the server recognizes as RTMPT client.
String	<a href="#">getVersion()</a> Get server version number.
<a href="#">VHostList</a>	<a href="#">getVHostList()</a> Returns the interface to the VHostList for the server
boolean	<a href="#">isDynamicLogContextLoaded(String logContext)</a> Returns true if the given dynamic log context is already loaded.
boolean	<a href="#">isSuspended()</a> Is the server currently suspended
String	<a href="#">readConfig(String sName)</a> Method to read xml config file..
void	<a href="#">reloadVHostConfig()</a> Reload the VHosts.xml file.
void	<a href="#">removeServerListener(IServerNotify serverListener)</a> Remove server listener
void	<a href="#">setCommandInterfaceHostPort(HostPort commandInterfaceHostPort)</a> Set the definition for the command interface.
void	<a href="#">setCoreHandlerPoolSize(int corePoolSize)</a> Set the handler core thread pool size.
void	<a href="#">setCoreTransportPoolSize(int corePoolSize)</a> Set the transport core thread pool size.
void	<a href="#">setDynamicLogProperties(java.util.Properties dynamicLogProperties)</a> Set the dynamic log properties set at the server level
void	<a href="#">setUserAgents(String[] userAgents)</a> Set a pipe " " delimited list of user agents that the server recognizes as RTMPT client.
void	<a href="#">startCommandInterface()</a> Start the command interface as defined in Server.xml.
void	<a href="#">startVHost(String vhostName)</a> 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(String vhostName)</u></a> 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(String sName, String data)</u></a> Method to write xml config file..

## Methods

### startCommandInterface

```
public void startCommandInterface()
```

Start the command interface as defined in Server.xml. The command interface is used by shutdown.sh script to stop the server. It is also used by ant task to stop and start the server on build events.

### stopCommandInterface

```
public void stopCommandInterface()
```

Stop the command interface as defined in Server.xml. The command interface is used by shutdown.sh script to stop the server. It is also used by ant task to stop and start the server on build events.

### getVersion

```
public String getVersion()
```

Get server version number.

**Returns:**

server version number

### reloadVHostConfig

```
public void reloadVHostConfig()
```

(continued from last page)

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.

---

(continued from last page)

**Returns:**connection counter

---

**getConnectionCounter**

```
public ConnectionCounterSimple getConnectionCounter(int counterIndex)
```

Get the server connection counter for a specific technology (see IVHost.COUNTER\_\*)

**Parameters:**

counterIndex - counter index (see IVHost.COUNTER\_\*)

**Returns:**connection counter

---

**getDateStarted**

```
public String getDateStarted()
```

Get the date and time the server was started.

**Returns:**date and time the server was started

---

**getTimeRunning**

```
public String getTimeRunning()
```

Get a formatted String of how long the server has been running.

**Returns:**formatted String of how long the server has been running

---

**getTimeRunningSeconds**

```
public double getTimeRunningSeconds()
```

Get time running in seconds

**Returns:**time running in seconds

---

**getCommandInterfaceHostPort**

```
public HostPort getCommandInterfaceHostPort()
```

Get the definition of the command interface

**Returns:**host port definition of command interface

---

**setCommandInterfaceHostPort**

```
public void setCommandInterfaceHostPort(HostPort commandInterfaceHostPort)
```

Set the definition for the command interface.

---

(continued from last page)

**Parameters:**commandInterfaceHostPort - host port definition of command interface

---

**getUserAgents**

```
public String[] getUserAgents()
```

Get a pipe "|" delimited list of user agents that the server recognizes as RTMPT client.

**Returns:**

pipe "|" delimited list of user agents that the server recognizes as RTMPT client

---

**setUserAgents**

```
public void setUserAgents(String[] userAgents)
```

Set a pipe "|" delimited list of user agents that the server recognizes as RTMPT client.

**Parameters:**

userAgents - pipe "|" delimited list of user agents that the server recognizes as RTMPT client

---

**addServerListener**

```
public void addServerListener(IServerNotify serverListener)
```

Add server listener

**Parameters:**

serverListener - server listener

---

**removeServerListener**

```
public void removeServerListener(IServerNotify serverListener)
```

Remove server listener

**Parameters:**

serverListener - server listener

---

**getAdminInterfaceObjectList**

```
public java.util.List getAdminInterfaceObjectList()
```

Get the list of objects exposed through JMX interface

**Returns:**

list of objects exposed through JMX interface

---

**getCoreTransportPoolSize**

```
public int getCoreTransportPoolSize()
```

Get the transport core thread pool size.

**Returns:**

default core thread pool size

---

(continued from last page)

## setCoreTransportPoolSize

```
public void setCoreTransportPoolSize(int corePoolSize)
```

Set the transport core thread pool size.

**Parameters:**

corePoolSize - core thread pool size

---

## getCoreHandlerPoolSize

```
public int getCoreHandlerPoolSize()
```

Get the handler core thread pool size.

**Returns:**

default core thread pool size

---

## setCoreHandlerPoolSize

```
public void setCoreHandlerPoolSize(int corePoolSize)
```

Set the handler core thread pool size.

**Parameters:**

corePoolSize - core thread pool size

---

## getThreadPool

```
public ThreadPool getThreadPool()
```

Get the server handler thread pool. Same as getHandlerThreadPool.

**Returns:**

server handler thread pool

---

## getTransportThreadPool

```
public ThreadPool getTransportThreadPool()
```

Get the server transport thread pool. This thread pool is used to read/write data from the transports sockets.

**Returns:**

server transport thread pool

---

## getHandlerThreadPool

```
public ThreadPool getHandlerThreadPool()
```

Get the server handler thread pool. This thread pool is used to process the incoming events.

**Returns:**

server handler thread pool

---

## getProperties

```
public WMSProperties getProperties()
```

---



(continued from last page)

Get server level properties collection

**Returns:**

server level properties collection

---

## 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 current suspended

---

## **stopAdminAgent**

```
public void stopAdminAgent()
```

Stop the JMX interface

---

## **getUDPPortManager**

```
public com.wowza.wms.transport.udp.UDPPortManager getUDPPortManager()
```

Get the UDP port manager which manages the allocation of incoming UDP port binding to be sure there are not port conflicts

**Returns:**

UDP port manager

---

## **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..

---

## com.wowza.wms.server Interface IServerNotify

All Subinterfaces:

[IServerNotify2](#)

public interface **IServerNotify**  
extends

IServerNotify: listener interface. Configured by adding class entries definitions to Server.xml.Startup order is: [constructor]; onServerConfigLoaded, onServerCreate, onServerInit Shutdown order is: onServerShutdownStart, onServerShutdownComplete, [exit]

### Method Summary

void	<a href="#">onServerCreate</a> ( <a href="#">IServer</a> server) Triggered when server object is first created.
void	<a href="#">onServerInit</a> ( <a href="#">IServer</a> server) Triggered when server initialization is complete and all VHosts have been started
void	<a href="#">onServerShutdownComplete</a> ( <a href="#">IServer</a> server) Triggered at the end of server shutdown
void	<a href="#">onServerShutdownStart</a> ( <a href="#">IServer</a> server) Triggered at the beginning of server shutdown

### Methods

#### onServerCreate

public void **onServerCreate**([IServer](#) server)

Triggered when server object is first created.

**Parameters:**

server - server object

#### onServerInit

public void **onServerInit**([IServer](#) server)

Triggered when server initialization is complete and all VHosts have been started

**Parameters:**

server - server object

#### onServerShutdownStart

public void **onServerShutdownStart**([IServer](#) server)

Triggered at the beginning of server shutdown

(continued from last page)

**Parameters:**

server - server object

---

## onServerShutdownComplete

```
public void onServerShutdownComplete(IServer server)
```

Triggered at the end of server shutdown

**Parameters:**

server - server object

## com.wowza.wms.server Interface IServerNotify2

All Superinterfaces:  
[IServerNotify](#)

public interface **IServerNotify2**  
extends [IServerNotify](#)

IServerNotify2: listener interface. Configured by adding class entries definitions to Server.xml.Startup order is: [constructor]; onServerConfigLoaded, onServerCreate, onServerInit Shutdown order is: onServerShutdownStart, onServerShutdownComplete, [exit]

### Method Summary

void	<a href="#">onServerConfigLoaded</a> ( <a href="#">IServer</a> server) Triggered when server configuration is loaded
------	---

Methods inherited from interface [com.wowza.wms.server.IServerNotify](#)

[onServerCreate](#), [onServerInit](#), [onServerShutdownComplete](#), [onServerShutdownStart](#)

### Methods

#### onServerConfigLoaded

public void **onServerConfigLoaded**([IServer](#) server)

Triggered when server configuration is loaded

**Parameters:**

server - server object

## com.wowza.wms.server

### Class Server

java.lang.Object

└─com.wowza.wms.server.Server

All Implemented Interfaces:

[IServer](#)

public class **Server**  
 extends Object  
 implements [IServer](#)

#### Field Summary

public static	<a href="#">logNotifier</a>
---------------	-----------------------------

#### Constructor Summary

public	<a href="#">Server()</a>
--------	--------------------------

#### Method Summary

void	<a href="#">addServerListener</a> ( <a href="#">IServerNotify</a> serverListener)
------	---

String	<a href="#">decodeStorageDir</a> ( <a href="#">IVHost</a> vhost, String storageDir)
--------	---

void	<a href="#">doWatchdog()</a>
------	------------------------------

com.wowza.wms.admin.AdminAgent	<a href="#">getAdminAgent()</a>
--------------------------------	---------------------------------

java.util.List	<a href="#">getAdminInterfaceObjectList()</a>
----------------	---

RandomIdGenerator	<a href="#">getClientIdGenerator()</a>
-------------------	--

int	<a href="#">getClientIdGeneratorRecycleDelaySize()</a>
-----	--

int	<a href="#">getClientIdGeneratorRecycleSize()</a>
-----	---

long	<a href="#">getClientIdGeneratorTimeout()</a>
------	---

Object	<a href="#">getCommandInterface()</a>
--------	---------------------------------------

<a href="#">HostPort</a>	<a href="#">getCommandInterfaceHostPort()</a>
--------------------------	---

long	<a href="#">getCommittedVirtualMemory()</a>
<a href="#">ConnectionCounter</a>	<a href="#">getConnectionCounter()</a>
ConnectionCounterSimple	<a href="#">getConnectionCounter(int counterIndex)</a>
<a href="#">IConnectionValidator</a>	<a href="#">getConnectionValidator()</a>
int	<a href="#">getCoreHandlerPoolSize()</a>
int	<a href="#">getCoreTransportPoolSize()</a>
int	<a href="#">getCryptoPoolActiveCount()</a>
int	<a href="#">getCryptoPoolMaxSize()</a>
long	<a href="#">getCurrentHeapSize()</a>
String	<a href="#">getDateStarted()</a>
java.util.Properties	<a href="#">getDynamicLogProperties()</a>
String	<a href="#">getGUID()</a>
<a href="#">ThreadPool</a>	<a href="#">getHandlerThreadPool()</a>
static <a href="#">Server</a>	<a href="#">getInstance()</a>
<a href="#">IOPerformanceCounter</a>	<a href="#">getIoPerformanceCounter()</a>
<a href="#">IOPerformanceCounter</a>	<a href="#">getIoPerformanceCounter(int counterIndex)</a>
JMXRemoteConfig	<a href="#">getJmxRemoteConfig()</a>
long	<a href="#">getLiveThreads()</a>
long	<a href="#">getMaxHeapSize()</a>
long	<a href="#">getPeakThreads()</a>
<a href="#">WMSProperties</a>	<a href="#">getProperties()</a>
String	<a href="#">getSessionGUID()</a>
<a href="#">ThreadPool</a>	<a href="#">getThreadPool()</a>
String	<a href="#">getTimeRunning()</a>

double	<a href="#">getTimeRunningSeconds()</a>
<a href="#">ThreadPool</a>	<a href="#">getTransportThreadPool()</a>
com.wowza.wms.transpo rt.udp.UDPPortManager	<a href="#">getUDPPortManager()</a>
String[]	<a href="#">getUserAgents()</a>
String	<a href="#">getVersion()</a>
<a href="#">VHostList</a>	<a href="#">getVHostList()</a>
boolean	<a href="#">isDynamicLogContextLoaded()</a> (String logContext)
boolean	<a href="#">isSuspended()</a>
static void	<a href="#">main()</a> (String[] args)
void	<a href="#">onNewVHost()</a> ( <a href="#">IVHost</a> vhost)
String	<a href="#">readConfig()</a> (String sName)
static String	<a href="#">readXMLConfig()</a> (String sPath)
void	<a href="#">reloadVHostConfig()</a>
void	<a href="#">removeServerListener()</a> ( <a href="#">IServerNotify</a> serverListener)
void	<a href="#">setCommandInterface()</a> (Object commandInterface)
void	<a href="#">setCommandInterfaceHostPort()</a> ( <a href="#">HostPort</a> commandInterfaceHostPort)
void	<a href="#">setCoreHandlerPoolSize()</a> (int corePoolSize)
void	<a href="#">setCoreTransportPoolSize()</a> (int corePoolSize)
void	<a href="#">setDynamicLogProperties()</a> (java.util.Properties dynamicLogProperties)
void	<a href="#">setIoPerformanceCounter()</a> ( <a href="#">IOPerformanceCounter</a> ioPerformanceCounter)
void	<a href="#">setUserAgents()</a> (String[] userAgents)
static void	<a href="#">start()</a>
void	<a href="#">startCommandInterface()</a>
void	<a href="#">startServer()</a>



void	<a href="#">startVHost</a> (String vhostName)
void	<a href="#">startVHosts</a> ()
void	<a href="#">stopAdminAgent</a> ()
void	<a href="#">stopCommandInterface</a> ()
void	<a href="#">stopServer</a> ()
void	<a href="#">stopVHost</a> (String vhostName)
void	<a href="#">stopVHosts</a> ()
void	<a href="#">suspendAllVHosts</a> ()
void	<a href="#">suspendCommandInterface</a> ()
void	<a href="#">suspendServer</a> ()
void	<a href="#">unbindAllVHosts</a> ()
<a href="#">IOPerformanceCounter</a>	<a href="#">updateIOPerformance</a> ()
void	<a href="#">updateLoggingDuration</a> ()
boolean	<a href="#">writeConfig</a> (String sName, String data)
static boolean	<a href="#">writeXMLConfig</a> (String sPath, String data)

**Methods inherited from class java.lang.Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

**Methods inherited from interface [com.wowza.wms.server.IServer](#)**

[addServerListener](#), [getAdminInterfaceObjectList](#), [getClientIdGenerator](#), [getCommandInterfaceHostPort](#), [getConnectionCounter](#), [getConnectionCounter](#), [getCoreHandlerPoolSize](#), [getCoreTransportPoolSize](#), [getDateStarted](#), [getDynamicLogProperties](#), [getHandlerThreadPool](#), [getIoPerformanceCounter](#), [getIoPerformanceCounter](#), [getProperties](#), [getThreadPool](#), [getTimeRunning](#), [getTimeRunningSeconds](#), [getTransportThreadPool](#), [getUDPPortManager](#), [getUserAgents](#), [getVersion](#), [getVHostList](#), [isDynamicLogContextLoaded](#), [isSuspended](#), [readConfig](#), [reloadVHostConfig](#), [removeServerListener](#), [setCommandInterfaceHostPort](#), [setCoreHandlerPoolSize](#), [setCoreTransportPoolSize](#), [setDynamicLogProperties](#), [setUserAgents](#), [startCommandInterface](#), [startVHost](#), [startVHosts](#), [stopAdminAgent](#), [stopCommandInterface](#), [stopVHost](#), [stopVHosts](#), [suspendAllVHosts](#), [suspendServer](#), [unbindAllVHosts](#), [writeConfig](#)

(continued from last page)

## Fields

### logNotifier

```
public static com.wowza.wms.logging.ILogNotify logNotifier
```

## 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()
```

### suspendCommandInterface

```
public void suspendCommandInterface()
```

(continued from last page)

---

### startCommandInterface

```
public void startCommandInterface()
```

---

### stopCommandInterface

```
public void stopCommandInterface()
```

---

### startServer

```
public void startServer()
```

---

### getVersion

```
public String getVersion()
```

---

### reloadVHostConfig

```
public void reloadVHostConfig()
```

---

### stopVHost

```
public void stopVHost(String vhostName)
```

---

### stopVHosts

```
public void stopVHosts()
```

---

### decodeStorageDir

```
public String decodeStorageDir(IVHost vhost,  
                                String storageDir)
```

---

### startVHost

```
public void startVHost(String vhostName)
```

---

(continued from last page)

---

## 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()
```

---

---

## getIoPerformanceCounter

```
public IOPerformanceCounter getIoPerformanceCounter(int counterIndex)
```

---

---

## setIoPerformanceCounter

```
public void setIoPerformanceCounter(IOPerformanceCounter ioPerformanceCounter)
```

---

(continued from last page)

---

### getConnectionCounter

```
public ConnectionCounter getConnectionCounter()
```

---

### getConnectionCounter

```
public ConnectionCounterSimple getConnectionCounter(int counterIndex)
```

---

### getConnectionValidator

```
public IConnectionValidator getConnectionValidator()
```

---

### onNewVHost

```
public void onNewVHost(IVHost vhost)
```

---

### getDateStarted

```
public String getDateStarted()
```

---

### getTimeRunning

```
public String getTimeRunning()
```

---

### getTimeRunningSeconds

```
public double getTimeRunningSeconds()
```

---

### getCommandInterfaceHostPort

```
public HostPort getCommandInterfaceHostPort()
```

---

### setCommandInterfaceHostPort

```
public void setCommandInterfaceHostPort(HostPort commandInterfaceHostPort)
```

---

(continued from last page)

---

## getCommandInterface

```
public Object getCommandInterface()
```

---

## setCommandInterface

```
public void setCommandInterface(Object commandInterface)
```

---

## getUserAgents

```
public String[] getUserAgents()
```

---

## setUserAgents

```
public void setUserAgents(String[] userAgents)
```

---

## updateLoggingDuration

```
public void updateLoggingDuration()
```

---

## addServerListener

```
public void addServerListener(IServerNotify serverListener)
```

---

## removeServerListener

```
public void removeServerListener(IServerNotify serverListener)
```

---

## getJmxRemoteConfig

```
public JMXRemoteConfig getJmxRemoteConfig()
```

---

## getAdminInterfaceObjectList

```
public java.util.List getAdminInterfaceObjectList()
```

---

## getCoreTransportPoolSize

```
public int getCoreTransportPoolSize()
```

---

(continued from last page)

---

## setCoreTransportPoolSize

```
public void setCoreTransportPoolSize(int corePoolSize)
```

---

## getCoreHandlerPoolSize

```
public int getCoreHandlerPoolSize()
```

---

## setCoreHandlerPoolSize

```
public void setCoreHandlerPoolSize(int corePoolSize)
```

---

## getThreadPool

```
public ThreadPool getThreadPool()
```

---

## getTransportThreadPool

```
public ThreadPool getTransportThreadPool()
```

---

## getHandlerThreadPool

```
public ThreadPool getHandlerThreadPool()
```

---

## getProperties

```
public WMSProperties getProperties()
```

---

## getClientIdGenerator

```
public RandomIdGenerator getClientIdGenerator()
```

---

## getDynamicLogProperties

```
public java.util.Properties getDynamicLogProperties()
```

---

(continued from last page)

---

## setDynamicLogProperties

```
public void setDynamicLogProperties(java.util.Properties dynamicLogProperties)
```

---

## isDynamicLogContextLoaded

```
public boolean isDynamicLogContextLoaded(String logContext)
```

---

## getVHostList

```
public VHostList getVHostList()
```

---

## getSessionGUID

```
public String getSessionGUID()
```

---

## getGUID

```
public String getGUID()
```

---

## 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()
```

---



(continued from last page)

---

**getCurrentHeapSize**

```
public long getCurrentHeapSize()
```

---

**getMaxHeapSize**

```
public long getMaxHeapSize()
```

---

**getCommittedVirtualMemory**

```
public long getCommittedVirtualMemory()
```

---

**readConfig**

```
public String readConfig(String sName)
```

---

**writeConfig**

```
public boolean writeConfig(String sName,  
    String data)
```

---

**readXMLConfig**

```
public static String readXMLConfig(String sPath)
```

---

**writeXMLConfig**

```
public static boolean writeXMLConfig(String sPath,  
    String data)
```

---

**getClientIdGeneratorTimeout**

```
public long getClientIdGeneratorTimeout()
```

---

**getClientIdGeneratorRecycleSize**

```
public int getClientIdGeneratorRecycleSize()
```

---

---

## getClientIdGeneratorRecycleDelaySize

```
public int getClientIdGeneratorRecycleDelaySize()
```

---

Package

**com.wowza.wms.sharedobject**

## com.wowza.wms.sharedobject Interface ISharedObject

All Known Implementing Classes:

[SharedObject](#)

public interface **ISharedObject**  
extends

ISharedObject: public interface to SharedObject class.

### Field Summary

public static final	<a href="#">FILEEXTENSION</a> Value: <b>rso</b>
public static final	<a href="#">SHARED_OBJECT_CMD_CONNECT</a> shared object command: connect Value: <b>1</b>
public static final	<a href="#">SHARED_OBJECT_CMD_CONNECTSUCCESS</a> shared object command: clear Value: <b>11</b>
public static final	<a href="#">SHARED_OBJECT_CMD_DELETE</a> shared object command: delete Value: <b>10</b>
public static final	<a href="#">SHARED_OBJECT_CMD_DISCONNECT</a> shared object command: disconnect Value: <b>2</b>
public static final	<a href="#">SHARED_OBJECT_CMD_ERROR</a> shared object command: error Value: <b>7</b>
public static final	<a href="#">SHARED_OBJECT_CMD_SEND</a> shared object command: send Value: <b>6</b>
public static final	<a href="#">SHARED_OBJECT_CMD_SETVALUE</a> shared object command: setvalue Value: <b>3</b>
public static final	<a href="#">SHARED_OBJECT_STATUS_CHANGE</a> shared object status: change Value: <b>4</b>
public static final	<a href="#">SHARED_OBJECT_STATUS_CLEAR</a> shared object status: clear Value: <b>8</b>

public static final	<a href="#"><u>SHARED_OBJECT_STATUS_DELETE</u></a> shared object status: delete Value: <b>9</b>
public static final	<a href="#"><u>SHARED_OBJECT_STATUS_SUCCESS</u></a> shared object status: success Value: <b>5</b>

## Method Summary

void	<a href="#"><u>acquire()</u></a> Increment the reference count to this shared object.
void	<a href="#"><u>addClient(IClient client)</u></a> Add a client to this shared object.
void	<a href="#"><u>addSlotListener(ISharedObjectSlotNotify slotListener)</u></a> Add a slot listener.
void	<a href="#"><u>clear()</u></a> Clear all properties from a shared object
void	<a href="#"><u>close()</u></a> Force close this shared object (not implemented)
boolean	<a href="#"><u>containsProperty(String slotName)</u></a> Returns true is slot/property name exists
boolean	<a href="#"><u>containsSlot(String slotName)</u></a> Returns true is slot/property name exists
void	<a href="#"><u>deleteSlot(IClient client, String slotName)</u></a> Remove a slot (property)
void	<a href="#"><u>deleteSlot(String slotName)</u></a> Remove slot (property)
void	<a href="#"><u>disconnect(IClient client)</u></a> Disconnect client from shared object.
void	<a href="#"><u>flush()</u></a> Flush (write to disk) shared object
java.util.List	<a href="#"><u>getClients()</u></a> Get a list of client that are connected to this shared object.
String	<a href="#"><u>getName()</u></a> Get shared object name
<a href="#"><u>ISharedObjects</u></a>	<a href="#"><u>getParent()</u></a> Get the shared object container to which this shared object belongs.
<a href="#"><u>AMFData</u></a>	<a href="#"><u>getProperty(String slotName)</u></a> Get slot (property) value.
int	<a href="#"><u>getRefCount()</u></a> Get the current reference (clients) connected to this shared object.

<a href="#"><u>ISharedObjectSlot</u></a>	<a href="#"><u>getSlot</u></a> (String name) Get ISharedObjectSlot interface to a slot (property) by name
java.util.List	<a href="#"><u>getSlotNames</u></a> ( ) Get a list of slot (property) names
java.util.List	<a href="#"><u>getSlots</u></a> ( ) Get a list of active slots
String	<a href="#"><u>getStorageDir</u></a> ( ) Get path used to store shared object.
int	<a href="#"><u>getVersion</u></a> ( ) Get the interval version number.
boolean	<a href="#"><u>isClient</u></a> ( <a href="#"><u>IClient</u></a> client) Is this client connected to shared object
boolean	<a href="#"><u>isPersistent</u></a> ( ) Is this shared object being persisted.
void	<a href="#"><u>lock</u></a> ( ) Lock a shared object for write access
int	<a href="#"><u>purge</u></a> (int version) Purge all deleted properties older than the version number
void	<a href="#"><u>putSlot</u></a> (String name, <a href="#"><u>ISharedObjectSlot</u></a> slot) Add a new slot (property) to a shared object.
void	<a href="#"><u>release</u></a> ( ) Decrement the reference count to this shared object.
void	<a href="#"><u>removeClient</u></a> ( <a href="#"><u>IClient</u></a> client) Remove a client from this shared object.
void	<a href="#"><u>removeSlotListener</u></a> ( <a href="#"><u>ISharedObjectSlotNotify</u></a> slotListener) Remove slot listener
void	<a href="#"><u>send</u></a> (String handlerName) Call client side handler attached to shared object (no parameters).
void	<a href="#"><u>send</u></a> (String handlerName, Object[] params) Call client side handler attached to shared object.
void	<a href="#"><u>setName</u></a> (String name) Set shared object name
void	<a href="#"><u>setPersistent</u></a> (boolean isPersistent) Set is shared object persisted.
void	<a href="#"><u>setProperty</u></a> (String slotName, <a href="#"><u>AMFData</u></a> data) Set slot (property) value as AMFData object.
void	<a href="#"><u>setProperty</u></a> (String slotName, boolean value) Set slot (property) value as a boolean value (will be wrapped in an AMFDataItem object)
void	<a href="#"><u>setProperty</u></a> (String slotName, java.util.Date value) Set slot (property) value as a date value (will be wrapped in an AMFDataItem object)

void	<a href="#"><code>setProperty</code></a> (String slotName, double value) Set slot (property) value as a double value (will be wrapped in an AMFDataItem object)
void	<a href="#"><code>setProperty</code></a> (String slotName, int value) Set slot (property) value as a int value (will be wrapped in an AMFDataItem object)
void	<a href="#"><code>setProperty</code></a> (String slotName, long value) Set slot (property) value as a long value (will be wrapped in an AMFDataItem object)
void	<a href="#"><code>setProperty</code></a> (String slotName, String value) Set slot (property) value as a string value (will be wrapped in an AMFDataItem object)
void	<a href="#"><code>setStorageDir</code></a> (String storageDir) Set path used to store shared object.
void	<a href="#"><code>setVersion</code></a> (int version) Set the internal version number.
int	<a href="#"><code>size</code></a> () Get the number of active slot (properties).
void	<a href="#"><code>unlock</code></a> () Unlock a shared object for write access
void	<a href="#"><code>writeDeleteError</code></a> (IClient client, String soName, boolean isPersistent, String slotName, String errorMsg) Write an delete error message back to the client
void	<a href="#"><code>writeSetValueError</code></a> (IClient client, String soName, boolean isPersistent, String slotName, String errorMsg) Write an set value error message back to the client

## Fields

### FILEEXTENSION

```
public static final java.lang.String FILEEXTENSION
```

Constant value: **rs0**

### SHARED\_OBJECT\_CMD\_CONNECT

```
public static final byte SHARED_OBJECT_CMD_CONNECT
```

shared object command: connect

Constant value: **1**

### SHARED\_OBJECT\_CMD\_DISCONNECT

```
public static final byte SHARED_OBJECT_CMD_DISCONNECT
```

shared object command: disconnect

Constant value: **2**

(continued from last page)

---

## SHARED\_OBJECT\_CMD\_SETVALUE

public static final byte **SHARED\_OBJECT\_CMD\_SETVALUE**

shared object command: setvalue  
Constant value: **3**

---

## SHARED\_OBJECT\_CMD\_SEND

public static final byte **SHARED\_OBJECT\_CMD\_SEND**

shared object command: send  
Constant value: **6**

---

## SHARED\_OBJECT\_CMD\_ERROR

public static final byte **SHARED\_OBJECT\_CMD\_ERROR**

shared object command: error  
Constant value: **7**

---

## SHARED\_OBJECT\_CMD\_DELETE

public static final byte **SHARED\_OBJECT\_CMD\_DELETE**

shared object command: delete  
Constant value: **10**

---

## SHARED\_OBJECT\_CMD\_CONNECTSUCCESS

public static final byte **SHARED\_OBJECT\_CMD\_CONNECTSUCCESS**

shared object command: clear  
Constant value: **11**

---

## SHARED\_OBJECT\_STATUS\_CHANGE

public static final byte **SHARED\_OBJECT\_STATUS\_CHANGE**

shared object status: change  
Constant value: **4**

---

## SHARED\_OBJECT\_STATUS\_SUCCESS

public static final byte **SHARED\_OBJECT\_STATUS\_SUCCESS**

shared object status: success  
Constant value: **5**

---

## SHARED\_OBJECT\_STATUS\_CLEAR

public static final byte **SHARED\_OBJECT\_STATUS\_CLEAR**

shared object status: clear  
Constant value: **8**

---

## SHARED\_OBJECT\_STATUS\_DELETE

public static final byte **SHARED\_OBJECT\_STATUS\_DELETE**

---



(continued from last page)

shared object status: delete  
Constant value: **9**

## Methods

### getSlotNames

```
public java.util.List getSlotNames()
```

Get a list of slot (property) names

**Returns:**

list of slot (property) names

---

### getSlot

```
public ISharedObjectSlot getSlot(String name)
```

Get ISharedObjectSlot interface to a slot (property) by name

**Parameters:**

name - slot (property) name

**Returns:**

ISharedObjectSlot interface

---

### putSlot

```
public void putSlot(String name,  
    ISharedObjectSlot slot)
```

Add a new slot (property) to a shared object.

**Parameters:**

name - slot (property) name

slot - new slot

---

### getClients

```
public java.util.List getClients()
```

Get a list of client that are connected to this shared object.

**Returns:**

list of client that are connected to this shared object

---

### isClient

```
public boolean isClient(IClient client)
```

Is this client connected to shared object

**Parameters:**

client - client

**Returns:**

true if client is conencted to shared object

---

## addClient

```
public void addClient(IClient client)
```

Add a client to this shared object.

**Parameters:**

client - client

---

## removeClient

```
public void removeClient(IClient client)
```

Remove a client from this shared object.

**Parameters:**

client - client

---

## size

```
public int size()
```

Get the number of active slot (properties). Deleted slots are not counted.

**Returns:**

number of active slot (properties). Deleted slots are not counted

---

## acquire

```
public void acquire()
```

Increment the reference count to this shared object.

---

## release

```
public void release()
```

Descrement the reference count to this shared object.

---

## close

```
public void close()
```

Force close this shared object (not implemented)

---

## getRefCount

```
public int getRefCount()
```

Get the current reference (clients) connected to this shared object. Includes unbalanced count of calls to acquire.

**Returns:**

the current reference (clients) connected to this shared object

---

## isPersistent

```
public boolean isPersistent()
```

---

(continued from last page)

Is this shared object being persisted.

**Returns:**

true if shared object is being persisted

---

**setPersistent**

```
public void setPersistent(boolean isPersistent)
```

Set is shared object persisted.

**Parameters:**

isPersistent - true if shared object is being persisted

---

**getVersion**

```
public int getVersion()
```

Get the interval version number.

**Returns:**

interval version number

---

**setVersion**

```
public void setVersion(int version)
```

Set the internal version number.

**Parameters:**

version - internal version number

---

**flush**

```
public void flush()
```

Flush (write to disk) shared object

---

**deleteSlot**

```
public void deleteSlot(IClient client,  
String slotName)
```

Remove a slot (property)

**Parameters:**

client - client removing slot or null if server side code  
slotName - slot (property) name

---

**deleteSlot**

```
public void deleteSlot(String slotName)
```

Remove slot (property)

**Parameters:**

slotName - slot (property) name

## getProperty

```
public AMFData getProperty(String slotName)
```

Get slot (property) value.

**Parameters:**

slotName - slot (property) name

**Returns:**

slot (property) value as AMFData object

---

## setProperty

```
public void setProperty(String slotName,  
    String value)
```

Set slot (property) value as a string value (will be wrapped in an AMFDataItem object)

**Parameters:**

slotName - slot (property) name

value - string value

---

## setProperty

```
public void setProperty(String slotName,  
    double value)
```

Set slot (property) value as a double value (will be wrapped in an AMFDataItem object)

**Parameters:**

slotName - slot (property) name

value - double value

---

## setProperty

```
public void setProperty(String slotName,  
    int value)
```

Set slot (property) value as a int value (will be wrapped in an AMFDataItem object)

**Parameters:**

slotName - slot (property) name

value - int value

---

## setProperty

```
public void setProperty(String slotName,  
    long value)
```

Set slot (property) value as a long value (will be wrapped in an AMFDataItem object)

**Parameters:**

slotName - slot (property) name

value - long value

---

(continued from last page)

## setProperty

```
public void setProperty(String slotName,  
    java.util.Date value)
```

Set slot (property) value as a date value (will be wrapped in an AMFDataItem object)

### Parameters:

slotName - slot (property) name  
value - date value

---

## setProperty

```
public void setProperty(String slotName,  
    boolean value)
```

Set slot (property) value as a boolean value (will be wrapped in an AMFDataItem object)

### Parameters:

slotName - slot (property) name  
value - boolean value

---

## setProperty

```
public void setProperty(String slotName,  
    AMFData data)
```

Set slot (property) value as AMFData object.

### Parameters:

slotName - slot (property) name  
data - slot value as AMFData object. Example: new AMFDataItem((double)1.234) or new AMFDataItem()

---

## disconnect

```
public void disconnect(IClient client)
```

Disconnect client from shared object.

### Parameters:

client - client to disconnect

---

## getName

```
public String getName()
```

Get shared object name

### Returns:

shared object name

---

## setName

```
public void setName(String name)
```

Set shared object name

### Parameters:

name - shared object name

## send

```
public void send(String handlerName,  
                Object[] params)
```

Call client side handler attached to shared object.

**Parameters:**

handlerName - handler name

params - variable argument list of parameters

---

## send

```
public void send(String handlerName)
```

Call client side handler attached to shared object (no parameters).

**Parameters:**

handlerName - handler name

---

## addSlotListener

```
public void addSlotListener(ISharedObjectSlotNotify slotListener)
```

Add a slot listener. Will receive the following events: onSlotSetValue and onSlotDelete

**Parameters:**

slotListener - slot listener

---

## removeSlotListener

```
public void removeSlotListener(ISharedObjectSlotNotify slotListener)
```

Remove slot listener

**Parameters:**

slotListener - slot listener

---

## getStorageDir

```
public String getStorageDir()
```

Get path used to store shared object.

**Returns:**

path used to store shared object

---

## setStorageDir

```
public void setStorageDir(String storageDir)
```

Set path used to store shared object.

**Parameters:**

storageDir - path used to store shared object

---

(continued from last page)

---

## writeSetValueError

```
public void writeSetValueError(IClient client,  
    String soName,  
    boolean isPersistent,  
    String slotName,  
    String errorMsg)
```

Write an set value error message back to the client

### Parameters:

client - client removing slot or null if server side code  
soName - sharedObject name  
isPersistent - is persistent  
slotName - slot name  
errorMsg - error message

---

## writeDeleteError

```
public void writeDeleteError(IClient client,  
    String soName,  
    boolean isPersistent,  
    String slotName,  
    String errorMsg)
```

Write an delete error message back to the client

### Parameters:

client - client removing slot or null if server side code  
soName - sharedObject name  
isPersistent - is persistent  
slotName - slot name  
errorMsg - error message

---

## clear

```
public void clear()
```

Clear all properties from a shared object

---

## purge

```
public int purge(int version)
```

Purge all deleted properties older than the version number

### Returns:

number of slots purged

---

## getSlots

```
public java.util.List getSlots()
```

Get a list of active slots

### Returns:

list of active slots

---

(continued from last page)

## containsProperty

```
public boolean containsProperty(String slotName)
```

Returns true is slot/property name exists

**Parameters:**

slotName - slot/property name

**Returns:**

true is slot/property name exists

---

## containsSlot

```
public boolean containsSlot(String slotName)
```

Returns true is slot/property name exists

**Parameters:**

slotName - slot/property name

**Returns:**

true is slot/property name exists

---

## lock

```
public void lock()
```

Lock a shared object for write access

---

## unlock

```
public void unlock()
```

Unlock a shared object for write access

---

## getParent

```
public ISharedObjects getParent()
```

Get the shared object container to which this shared object belongs.

**Returns:**

shared object container

---



## com.wowza.wms.sharedobject Interface ISharedObjectNotify

public interface **ISharedObjectNotify**  
extends

ISharedObjectNotify: listener interface used by IApplicationInstance addSharedObjectListener

### Method Summary

void	<a href="#">onSharedObjectConnect</a> ( <a href="#">ISharedObject</a> sharedObject, <a href="#">IClient</a> client) Triggered when client connects to sharedObject
void	<a href="#">onSharedObjectCreate</a> ( <a href="#">ISharedObject</a> sharedObject) Triggered when sharedObject created
void	<a href="#">onSharedObjectDestroy</a> ( <a href="#">ISharedObject</a> sharedObject) Triggered when sharedObject destroyed
void	<a href="#">onSharedObjectDisconnect</a> ( <a href="#">ISharedObject</a> sharedObject, <a href="#">IClient</a> client) Triggered when client disconnects from sharedObject

### Methods

#### onSharedObjectCreate

public void **onSharedObjectCreate**([ISharedObject](#) sharedObject)

Triggered when sharedObject created

**Parameters:**

sharedObject - sharedObject

#### onSharedObjectDestroy

public void **onSharedObjectDestroy**([ISharedObject](#) sharedObject)

Triggered when sharedObject destroyed

**Parameters:**

sharedObject - sharedObject

#### onSharedObjectConnect

public void **onSharedObjectConnect**([ISharedObject](#) sharedObject, [IClient](#) client)

Triggered when client connects to sharedObject

**Parameters:**

sharedObject - sharedObject

client - client

## onSharedObjectDisconnect

```
public void onSharedObjectDisconnect(ISharedObject sharedObject,  
    IClient client)
```

Triggered when client disconnects from sharedObject

**Parameters:**

sharedObject - sharedObject

client - client

## com.wowza.wms.sharedobject Interface ISharedObjects

public interface **ISharedObjects**  
extends

ISharedObjects: public interface to SharedObjects. Represent the list of shared objects managed by IApplicationInstance.

### Method Summary

void	<a href="#"><u>addSharedObjectListener</u></a> ( <a href="#"><u>ISharedObjectNotify</u></a> sharedObjectListener)
	Add a shared object listener.
void	<a href="#"><u>disconnect</u></a> ( <a href="#"><u>IClient</u></a> client)
	Disconnect client from all shared objects in list.
boolean	<a href="#"><u>exists</u></a> ( <a href="#"><u>ISharedObject</u></a> sharedObject)
	Is sharedObject in this list (by shared object reference).
boolean	<a href="#"><u>exists</u></a> (String objectName)
	Is sharedObject in this list (by name).
void	<a href="#"><u>flush</u></a> ()
	Flush all persistent shared objects to disk.
<a href="#"><u>ISharedObject</u></a>	<a href="#"><u>get</u></a> (String name)
	Get shared object by name.
java.util.List	<a href="#"><u>getObjectNames</u></a> ()
	Get a list of shared object names.
<a href="#"><u>ISharedObject</u></a>	<a href="#"><u>getOrCreate</u></a> (String name)
	Get shared object by name if it does not exist create a new shared object with the given name.
String	<a href="#"><u>getStorageDir</u></a> ()
	Get the storage directory for all shared objects in list.
boolean	<a href="#"><u>isPersistent</u></a> ()
	Are shared objects in list persistent.
void	<a href="#"><u>load</u></a> ()
	Load persistent shared objects from file system.
void	<a href="#"><u>put</u></a> (String objectName, <a href="#"><u>ISharedObject</u></a> sharedObject)
	Add or replace a shared object.
void	<a href="#"><u>remove</u></a> (String objectName)
	Remove a shared object.
void	<a href="#"><u>removeClient</u></a> ( <a href="#"><u>IClient</u></a> client)
	Remove a client from any shared object that it is connected to in this list.
void	<a href="#"><u>removeSharedObjectListener</u></a> ( <a href="#"><u>ISharedObjectNotify</u></a> sharedObjectListener)
	Remove a shared object listener.

void	<a href="#"><code>setPersistent</code></a> (boolean isPersistent) Set is shared object in list persistent
void	<a href="#"><code>setStorageDir</code></a> (String storageDir) Set the storage directory for all shared objects in list.
int	<a href="#"><code>size</code></a> () Get number of shared objects.

## Methods

### size

```
public int size()
```

Get number of shared objects.

**Returns:**

number of shared objects

### get

```
public ISharedObject get(String name)
```

Get shared object by name. If the shared object does not exists null will be returned. (see `ISharedObjects.getOrCreate`)

**Parameters:**

name - shared object name

**Returns:**

shared object

### getOrCreate

```
public ISharedObject getOrCreate(String name)
```

Get shared object by name if it does not exist create a new shared object with the given name.

**Parameters:**

name - shared object name

**Returns:**

shared object

### getObjectNames

```
public java.util.List getObjectNames()
```

Get a list of shared object names.

**Returns:**

list of shared object names

(continued from last page)

## put

```
public void put(String objectName,  
    ISharedObject sharedObject)
```

Add or replace a shared object.

### Parameters:

objectName - shared object name  
sharedObject - shared object

---

## remove

```
public void remove(String objectName)
```

Remove a shared object.

### Parameters:

objectName - shared object name

---

## exists

```
public boolean exists(String objectName)
```

Is sharedObject in this list (by name).

### Parameters:

objectName - shared object name

### Returns:

true if shared object in list

---

## exists

```
public boolean exists(ISharedObject sharedObject)
```

Is sharedObject in this list (by shared object reference).

### Parameters:

sharedObject - shared object

### Returns:

true if shared object in list

---

## isPersistent

```
public boolean isPersistent()
```

Are shared objects in list persistent.

### Returns:

true if shared objects in list are persistent

---

## setPersistent

```
public void setPersistent(boolean isPersistent)
```

Set if shared object in list persistent

(continued from last page)

**Parameters:**

isPersistent - true is shared objects in list are persistent

---

**removeClient**

```
public void removeClient(IClient client)
```

Remove a client from any shared object that it is connected to in this list.

**Parameters:**

client - client

---

**flush**

```
public void flush()
```

Flush all persistent shared objects to disk.

---

**disconnect**

```
public void disconnect(IClient client)
```

Disconnect client from all shared objects in list.

**Parameters:**

client - client

---

**addSharedObjectListener**

```
public void addSharedObjectListener(ISharedObjectNotify sharedObjectListener)
```

Add a shared object listener. The listener will receive the following events: onSharedObjectCreate, onSharedObjectDestroy, onSharedObjectConnect, onSharedObjectDisconnect.

**Parameters:**

sharedObjectListener

---

**removeSharedObjectListener**

```
public void removeSharedObjectListener(ISharedObjectNotify sharedObjectListener)
```

Remove a shared object listener.

**Parameters:**

sharedObjectListener

---

**getStorageDir**

```
public String getStorageDir()
```

Get the storage directory for all shared objects in list.

**Returns:**

storage dir

---

**setStorageDir**

```
public void setStorageDir(String storageDir)
```

(continued from last page)

Set the storage directory for all shared objects in list.

**Parameters:**

storageDir

---

## **load**

```
public void load()
```

Load persistent shared objects from file system.

## com.wowza.wms.sharedobject Interface ISharedObjectSlot

public interface **ISharedObjectSlot**  
extends

ISharedObjectSlot: public interface to SharedObjectSlot class.

### Method Summary

<a href="#">AMFData</a>	<a href="#">getData()</a> Get slot data as AMFData object.
int	<a href="#">getLastClientId()</a> Get client id of client that performed last operation on slot.
int	<a href="#">getLastOperation()</a> Get last slot (property) operation.
String	<a href="#">getName()</a> Get slot (property) name.
int	<a href="#">getSlotVersion()</a> Get slot version
int	<a href="#">getSoVersion()</a> Get parent shared object version
void	<a href="#">incSlotVersion()</a> Increment slot version by 1.
void	<a href="#">init</a> (String name, <a href="#">AMFData</a> data, int slotVersion) Initialize shared object slot
void	<a href="#">setData</a> ( <a href="#">AMFData</a> data) Set slot data as AMFData object.
void	<a href="#">setData</a> (byte[] data) Set slot data as byte[].
void	<a href="#">setData</a> (byte[] data, <a href="#">AMFDataContextDeserialize</a> context) Set slot data as byte[].
void	<a href="#">setLastClientId</a> (int lastClientId) Set client id of client that performed last operation on slot.
void	<a href="#">setLastOperation</a> (int lastOperation) Set last slot (property) operation.
void	<a href="#">setName</a> (String name) Set slot (property) name
void	<a href="#">setSlotVersion</a> (int slotVersion) Set slot version



void	<a href="#">setSoVersion</a> (int soVersion) Set parent shared object version
------	--

## Methods

### init

```
public void init(String name,  
    AMFData data,  
    int slotVersion)
```

Initialize shared object slot

**Parameters:**

name - slot (property) name  
data - data  
slotVersion - version

### getData

```
public AMFData getData()
```

Get slot data as AMFData object.

**Returns:**

slot data as AMFData object

### setData

```
public void setData(byte[] data,  
    AMFDataContextDeserialize context)
```

Set slot data as byte[]. Data will be deserialized as AMFData object.

**Parameters:**

data - data as byte[]  
context - deserialization context

### setData

```
public void setData(byte[] data)
```

Set slot data as byte[]. Data will be deserialized as AMFData object.

**Parameters:**

data - data as byte[]

### setData

```
public void setData(AMFData data)
```

Set slot data as AMFData object.

**Parameters:**

data - data as AMFData object

## getName

```
public String getName()
```

Get slot (property) name.

**Returns:**

slot (property) name

---

## setName

```
public void setName(String name)
```

Set slot (property) name

**Parameters:**

name - slot (property) name

---

## getSlotVersion

```
public int getSlotVersion()
```

Get slot version

**Returns:**

slot version

---

## setSlotVersion

```
public void setSlotVersion(int slotVersion)
```

Set slot version

**Parameters:**

slotVersion - slot version

---

## incSlotVersion

```
public void incSlotVersion()
```

Increment slot version by 1.

---

## getSoVersion

```
public int getSoVersion()
```

Get parent shared object version

**Returns:**

parent shared object version

---

## setSoVersion

```
public void setSoVersion(int soVersion)
```

Set parent shared object version

---

(continued from last page)

**Parameters:**

soVersion - parent shared object version

---

**getLastOperation**

```
public int getLastOperation()
```

Get last slot (property) operation. See ISharedObject.SHAREDOBJECT\_CMD\_\*.

**Returns:**

last slot (property) operation (ISharedObject.SHAREDOBJECT\_CMD\_\*)

---

**setLastOperation**

```
public void setLastOperation(int lastOperation)
```

Set last slot (property) operation. See ISharedObject.SHAREDOBJECT\_CMD\_\*.

**Parameters:**

lastOperation - last slot (property) operation (ISharedObject.SHAREDOBJECT\_CMD\_\*)

---

**getLastClientId**

```
public int getLastClientId()
```

Get client id of client that performed last operation on slot.

**Returns:**

client id of client that performed last operation on slot

---

**setLastClientId**

```
public void setLastClientId(int lastClientId)
```

Set client id of client that performed last operation on slot.

**Parameters:**

lastClientId - client id of client that performed last operation on slot

## com.wowza.wms.sharedobject Interface ISharedObjectSlotNotify

public interface **ISharedObjectSlotNotify**  
extends

ISharedObjectNotify: listener interface used by ISharedObject addSlotListener

### Method Summary

void	<a href="#">onSlotDelete</a> ( <a href="#">ISharedObject</a> sharedObject, <a href="#">ISharedObjectSlot</a> slot) Triggered when sharedObject slot value deleted
void	<a href="#">onSlotSetValue</a> ( <a href="#">ISharedObject</a> sharedObject, <a href="#">ISharedObjectSlot</a> slot) Triggered when sharedObject slot value set

### Methods

#### onSlotSetValue

public void **onSlotSetValue**([ISharedObject](#) sharedObject,  
[ISharedObjectSlot](#) slot)

Triggered when sharedObject slot value set

**Parameters:**

sharedObject  
slot

#### onSlotDelete

public void **onSlotDelete**([ISharedObject](#) sharedObject,  
[ISharedObjectSlot](#) slot)

Triggered when sharedObject slot value deleted

**Parameters:**

sharedObject  
slot

## com.wowza.wms.sharedobject Class SharedObject

java.lang.Object

└─com.wowza.wms.sharedobject.SharedObject

All Implemented Interfaces:

[ISharedObject](#)

public class **SharedObject**  
extends Object  
implements [ISharedObject](#)

### Field Summary

public static final	<a href="#">READACCESS</a> Value: <b>0</b>
public static final	<a href="#">WRITEACCESS</a> Value: <b>1</b>

Fields inherited from interface [com.wowza.wms.sharedobject.ISharedObject](#)

[FILEEXTENSION](#), [SHARED\\_OBJECT\\_CMD\\_CONNECT](#), [SHARED\\_OBJECT\\_CMD\\_CONNECTSUCCESS](#),  
[SHARED\\_OBJECT\\_CMD\\_DELETE](#), [SHARED\\_OBJECT\\_CMD\\_DISCONNECT](#), [SHARED\\_OBJECT\\_CMD\\_ERROR](#),  
[SHARED\\_OBJECT\\_CMD\\_SEND](#), [SHARED\\_OBJECT\\_CMD\\_SETVALUE](#), [SHARED\\_OBJECT\\_STATUS\\_CHANGE](#),  
[SHARED\\_OBJECT\\_STATUS\\_CLEAR](#), [SHARED\\_OBJECT\\_STATUS\\_DELETE](#), [SHARED\\_OBJECT\\_STATUS\\_SUCCESS](#)

### Constructor Summary

public	<a href="#">SharedObject</a> (String name) Create new shared object
public	<a href="#">SharedObject</a> (String name, boolean isPersistent, String storageDir) Create new shared object

### Method Summary

void	<a href="#">acquire</a> ()
void	<a href="#">addClient</a> ( <a href="#">IClient</a> client)
void	<a href="#">addSlotListener</a> ( <a href="#">ISharedObjectSlotNotify</a> slotListener)
void	<a href="#">clear</a> ()
void	<a href="#">close</a> ()

boolean	<a href="#"><u>containsProperty</u></a> (String slotName)
boolean	<a href="#"><u>containsSlot</u></a> (String slotName)
void	<a href="#"><u>deleteSlot</u></a> ( <a href="#"><u>IClient</u></a> client, String slotName)
void	<a href="#"><u>deleteSlot</u></a> (String slotName)
void	<a href="#"><u>disconnect</u></a> ( <a href="#"><u>IClient</u></a> client)
void	<a href="#"><u>flush</u></a> ()
static boolean[]	<a href="#"><u>getAccess</u></a> ( <a href="#"><u>IClient</u></a> client, String soName)
java.util.List	<a href="#"><u>getClients</u></a> ()
void	<a href="#"><u>getClientUpdates</u></a> ( <a href="#"><u>IClient</u></a> client)
String	<a href="#"><u>getName</u></a> ()
<a href="#"><u>ISharedObjects</u></a>	<a href="#"><u>getParent</u></a> ()
<a href="#"><u>AMFData</u></a>	<a href="#"><u>getProperty</u></a> (String slotName)
int	<a href="#"><u>getRefCount</u></a> ()
<a href="#"><u>ISharedObjectSlot</u></a>	<a href="#"><u>getSlot</u></a> (String name)
java.util.List	<a href="#"><u>getSlotNames</u></a> ()
java.util.List	<a href="#"><u>getSlots</u></a> ()
String	<a href="#"><u>getStorageDir</u></a> ()
int	<a href="#"><u>getVersion</u></a> ()
boolean	<a href="#"><u>isClient</u></a> ( <a href="#"><u>IClient</u></a> client)
boolean	<a href="#"><u>isPersistent</u></a> ()
void	<a href="#"><u>load</u></a> ()
void	<a href="#"><u>lock</u></a> ()
void	<a href="#"><u>notifySlotDelete</u></a> ( <a href="#"><u>ISharedObjectSlot</u></a> slot)
void	<a href="#"><u>notifySlotSetValue</u></a> ( <a href="#"><u>ISharedObjectSlot</u></a> slot)

int	<a href="#">purge</a> (int version)
void	<a href="#">putSlot</a> (String name, <a href="#">ISharedObjectSlot</a> slot)
void	<a href="#">release</a> ()
void	<a href="#">removeClient</a> ( <a href="#">IClient</a> client)
void	<a href="#">removeSlotListener</a> ( <a href="#">ISharedObjectSlotNotify</a> slotListener)
void	<a href="#">send</a> (String handlerName)
void	<a href="#">send</a> (String handlerName, Object[] params)
void	<a href="#">sendInternal</a> ( <a href="#">IClient</a> client, String handlerName, byte[] msgBytes)
void	<a href="#">sendInternal</a> ( <a href="#">IClient</a> client, String handlerName, byte[] msgBytes, <a href="#">AMFDataContextDeserialize</a> context)
void	<a href="#">sendInternal</a> (String handlerName, <a href="#">AMFData[]</a> params)
void	<a href="#">setName</a> (String name)
void	<a href="#">setParent</a> ( <a href="#">ISharedObjects</a> parent)
void	<a href="#">setPersistent</a> (boolean isPersistent)
void	<a href="#">setProperty</a> (String slotName, <a href="#">AMFData</a> data)
void	<a href="#">setProperty</a> (String slotName, boolean value)
void	<a href="#">setProperty</a> (String slotName, java.util.Date value)
void	<a href="#">setProperty</a> (String slotName, double value)
void	<a href="#">setProperty</a> (String slotName, int value)
void	<a href="#">setProperty</a> (String slotName, long value)
void	<a href="#">setProperty</a> (String slotName, String value)
void	<a href="#">setSlotValue</a> ( <a href="#">IClient</a> client, String slotName, <a href="#">AMFData</a> amfData)
void	<a href="#">setSlotValue</a> ( <a href="#">IClient</a> client, String slotName, byte[] byteData)
void	<a href="#">setSlotValue</a> ( <a href="#">IClient</a> client, String slotName, byte[] byteData, <a href="#">AMFData</a> amfData)

void	<a href="#">setSlotValue(IClient client, String slotName, byte[] byteData, <a href="#">AMFData</a> amfData, <a href="#">AMFDataContextDeserialize</a> context)</a>
void	<a href="#">setSlotValue(IClient client, String slotName, byte[] byteData, <a href="#">AMFDataContextDeserialize</a> context)</a>
void	<a href="#">setStorageDir</a> (String storageDir)
void	<a href="#">setVersion</a> (int version)
int	<a href="#">size</a> ()
void	<a href="#">unlock</a> ()
void	<a href="#">writeDeleteError(IClient client, String soName, boolean isPersistent, String slotName, String errorMsg)</a>
static void	<a href="#">writeError(IClient client, String soName, boolean isPersistent, String errorMsg, boolean isConnect)</a>
void	<a href="#">writeSetValueError(IClient client, String soName, boolean isPersistent, String slotName, String errorMsg)</a>

#### Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

#### Methods inherited from interface [com.wowza.wms.sharedobject.ISharedObject](#)

[acquire](#), [addClient](#), [addSlotListener](#), [clear](#), [close](#), [containsProperty](#), [containsSlot](#), [deleteSlot](#), [deleteSlot](#), [disconnect](#), [flush](#), [getClients](#), [getName](#), [getParent](#), [getProperty](#), [getRefCount](#), [getSlot](#), [getSlotNames](#), [getSlots](#), [getStorageDir](#), [getVersion](#), [isClient](#), [isPersistent](#), [lock](#), [purge](#), [putSlot](#), [release](#), [removeClient](#), [removeSlotListener](#), [send](#), [send](#), [setName](#), [setPersistent](#), [setProperty](#), [setProperty](#), [setProperty](#), [setProperty](#), [setProperty](#), [setProperty](#), [setStorageDir](#), [setVersion](#), [size](#), [unlock](#), [writeDeleteError](#), [writeSetValueError](#)

## Fields

### READACCESS

```
public static final int READACCESS
```

Constant value: **0**

### WRITEACCESS

```
public static final int WRITEACCESS
```

Constant value: **1**



(continued from last page)

## Constructors

### SharedObject

```
public SharedObject(String name)
```

Create new shared object

**Parameters:**

name - shared object name

---

### SharedObject

```
public SharedObject(String name,  
                    boolean isPersistent,  
                    String storageDir)
```

Create new shared object

**Parameters:**

name - shared object name

isPersistent - is persistent

storageDir - storage directory for persistent shared object

## Methods

### size

```
public int size()
```

---

### purge

```
public int purge(int version)
```

---

### clear

```
public void clear()
```

---

### getSlots

```
public java.util.List getSlots()
```

---

### getSlotNames

```
public java.util.List getSlotNames()
```

---

(continued from last page)

---

## getSlot

```
public ISharedObjectSlot getSlot(String name)
```

---

## putSlot

```
public void putSlot(String name,  
    ISharedObjectSlot slot)
```

---

## getClients

```
public java.util.List getClients()
```

---

## isClient

```
public boolean isClient(IClient client)
```

---

## addClient

```
public void addClient(IClient client)
```

---

## removeClient

```
public void removeClient(IClient client)
```

---

## acquire

```
public void acquire()
```

---

## release

```
public void release()
```

---

## getRefCount

```
public int getRefCount()
```

---

## isPersistent

```
public boolean isPersistent()
```

---

(continued from last page)

---

### setPersistent

```
public void setPersistent(boolean isPersistent)
```

---

### getVersion

```
public int getVersion()
```

---

### setVersion

```
public void setVersion(int version)
```

---

### deleteSlot

```
public void deleteSlot(String slotName)
```

---

### containsProperty

```
public boolean containsProperty(String slotName)
```

---

### containsSlot

```
public boolean containsSlot(String slotName)
```

---

### deleteSlot

```
public void deleteSlot(IClient client,  
                        String slotName)
```

---

### getProperty

```
public AMFData getProperty(String slotName)
```

---

### setProperty

```
public void setProperty(String slotName,  
                        boolean value)
```

---

---

### setProperty

```
public void setProperty(String slotName,  
    java.util.Date value)
```

---

### setProperty

```
public void setProperty(String slotName,  
    double value)
```

---

### setProperty

```
public void setProperty(String slotName,  
    long value)
```

---

### setProperty

```
public void setProperty(String slotName,  
    int value)
```

---

### setProperty

```
public void setProperty(String slotName,  
    String value)
```

---

### setProperty

```
public void setProperty(String slotName,  
    AMFData data)
```

---

### setSlotValue

```
public void setSlotValue(IClient client,  
    String slotName,  
    AMFData amfData)
```

---

### setSlotValue

```
public void setSlotValue(IClient client,  
    String slotName,  
    byte[] byteData)
```

---

(continued from last page)

---

## setSlotValue

```
public void setSlotValue(IClient client,  
    String slotName,  
    byte[] byteData,  
    AMFDataContextDeserialize context)
```

---

## setSlotValue

```
public void setSlotValue(IClient client,  
    String slotName,  
    byte[] byteData,  
    AMFData amfData)
```

---

## setSlotValue

```
public void setSlotValue(IClient client,  
    String slotName,  
    byte[] byteData,  
    AMFData amfData,  
    AMFDataContextDeserialize context)
```

---

## writeError

```
public static void writeError(IClient client,  
    String soName,  
    boolean isPersistent,  
    String errorMsg,  
    boolean isConnect)
```

---

## writeSetValueError

```
public void writeSetValueError(IClient client,  
    String soName,  
    boolean isPersistent,  
    String slotName,  
    String errorMsg)
```

---

## writeDeleteError

```
public void writeDeleteError(IClient client,  
    String soName,  
    boolean isPersistent,  
    String slotName,  
    String errorMsg)
```

---

## getClientUpdates

```
public void getClientUpdates(IClient client)
```

---

(continued from last page)

---

## disconnect

```
public void disconnect(IClient client)
```

---

## getName

```
public String getName()
```

---

## setName

```
public void setName(String name)
```

---

## send

```
public void send(String handlerName,  
                 Object[] params)
```

---

## send

```
public void send(String handlerName)
```

---

## sendInternal

```
public void sendInternal(String handlerName,  
                        AMFData\[\] params)
```

---

## sendInternal

```
public void sendInternal(IClient client,  
                        String handlerName,  
                        byte[] msgBytes)
```

---

## sendInternal

```
public void sendInternal(IClient client,  
                        String handlerName,  
                        byte[] msgBytes,  
                        AMFDataContextDeserialize context)
```

---

(continued from last page)

---

## addSlotListener

```
public void addSlotListener(ISharedObjectSlotNotify slotListener)
```

---

## removeSlotListener

```
public void removeSlotListener(ISharedObjectSlotNotify slotListener)
```

---

## notifySlotSetValue

```
public void notifySlotSetValue(ISharedObjectSlot slot)
```

---

## notifySlotDelete

```
public void notifySlotDelete(ISharedObjectSlot slot)
```

---

## getStorageDir

```
public String getStorageDir()
```

---

## setStorageDir

```
public void setStorageDir(String storageDir)
```

---

## flush

```
public void flush()
```

---

## load

```
public void load()
```

---

## close

```
public void close()
```

---

## lock

```
public void lock()
```

---

(continued from last page)

---

## unlock

```
public void unlock()
```

---

## setParent

```
public void setParent(ISharedObjects parent)
```

---

## getParent

```
public ISharedObjects getParent()
```

---

## getAccess

```
public static boolean[] getAccess(IClient client,  
    String soName)
```



---

Package

**com.wowza.wms.stream**

## com.wowza.wms.stream

### Class FastPlaySettings

java.lang.Object

└─com.wowza.wms.stream.FastPlaySettings

public class **FastPlaySettings**  
extends Object

FastPlaySettings: data container for fast play settings

#### Constructor Summary

public	<a href="#"><u>FastPlaySettings()</u></a> Create empty object
public	<a href="#"><u>FastPlaySettings(double multiplier, int fps, int direction)</u></a> Create object

#### Method Summary

int	<a href="#"><u>getDirection()</u></a> Get direction (1 forward, -1 reverse)
int	<a href="#"><u>getFps()</u></a> Get frames per second
double	<a href="#"><u>getMultiplier()</u></a> Get speed of fast play
long	<a href="#"><u>getStartTC()</u></a> Get the timecode (milliseconds) where this fast play started
long	<a href="#"><u>getStartTCOffset()</u></a> Get the timecode (milliseconds) where this fast play started (not sure why we have both values)
void	<a href="#"><u>setDirection(int direction)</u></a> Set direction (1 forward, -1 reverse)
void	<a href="#"><u>setFps(int fps)</u></a> Set frames per second
void	<a href="#"><u>setMultiplier(double multiplier)</u></a> Set speed of fast play
void	<a href="#"><u>setStartTC(long startTC)</u></a> Set the timecode (milliseconds) where this fast play started
void	<a href="#"><u>setStartTCOffset(long startTCOffset)</u></a> Set the timecode (milliseconds) where this fast play started (not sure why we have both values)

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

---

## Constructors

### FastPlaySettings

```
public FastPlaySettings()
```

Create empty object

---

### FastPlaySettings

```
public FastPlaySettings(double multiplier,  
                        int fps,  
                        int direction)
```

Create object

**Parameters:**

multiplier - speed

fps - frames per second

direction - direction (1 forward, -1 reverse)

## Methods

### getFps

```
public int getFps()
```

Get frames per second

**Returns:**

frames per second

---

### setFps

```
public void setFps(int fps)
```

Set frames per second

**Parameters:**

fps - frames per second

---

### getMultiplier

```
public double getMultiplier()
```

Get speed of fast play

**Returns:**

speed of fast play

---

(continued from last page)

---

## setMultiplier

```
public void setMultiplier(double multiplier)
```

Set speed of fast play

### Parameters:

multiplier - speed of fast play

---

## getDirection

```
public int getDirection()
```

Get direction (1 forward, -1 reverse)

### Returns:

direction (1 forward, -1 reverse)

---

## setDirection

```
public void setDirection(int direction)
```

Set direction (1 forward, -1 reverse)

### Parameters:

direction - direction (1 forward, -1 reverse)

---

## getStartTC

```
public long getStartTC()
```

Get the timecode (milliseconds) where this fast play started

### Returns:

timecode of start of fast play (milliseconds)

---

## setStartTC

```
public void setStartTC(long startTC)
```

Set the timecode (milliseconds) where this fast play started

### Parameters:

startTC - timecode of start of fast play (milliseconds)

---

## getStartTCOffset

```
public long getStartTCOffset()
```

Get the timecode (milliseconds) where this fast play started (not sure why we have both values)

### Returns:

timecode of start of fast play (milliseconds)

---

## setStartTCOffset

```
public void setStartTCOffset(long startTCOffset)
```

---

(continued from last page)

Set the timecode (milliseconds) where this fast play started (not sure why we have both values)

**Parameters:**

`startTCOffset` - timecode of start of fast play (milliseconds)

## com.wowza.wms.stream Interface IMediaIndexItem

public interface **IMediaIndexItem**  
extends

IMediaIndexItem: generic interface to mediaIndex

### Field Summary

public static final	<a href="#">MISSING</a> media file is missing Value: <b>**missing**</b>
---------------------	---

### Method Summary

<a href="#">IMediaIndexItem</a>	<a href="#">clone()</a> Make a clone copy of media index
String	<a href="#">getChecksum()</a> Get checksum for media index
int	<a href="#">getExtent()</a> Get the duration (milliseconds) of media index
long	<a href="#">getLastAccessed()</a> Get last time media index was accessed (milliseconds)
boolean	<a href="#">isHitEnd()</a> Is media index complete
void	<a href="#">setChecksum(IRandomAccessReader file)</a> Set media index checksum.
void	<a href="#">setChecksum(String checksum)</a> Set media index checksum
void	<a href="#">setLastAccessed(long time)</a> Set last time media index accessed (milliseconds)
void	<a href="#">touch()</a> Set last time accessed to current time

### Fields

#### MISSING

public static final java.lang.String **MISSING**

media file is missing  
Constant value: **\*\*missing\*\***

(continued from last page)

## Methods

### getExtent

```
public int getExtent()
```

Get the duration (milliseconds) of media index

**Returns:**

duration (milliseconds)

### getChecksum

```
public String getChecksum()
```

Get checksum for media index

**Returns:**

checksum

### setChecksum

```
public void setChecksum(String checksum)
```

Set media index checksum

**Parameters:**

checksum

### setChecksum

```
public void setChecksum(IRandomAccessReader file)
```

Set media index checksum. Extract data from File spec

**Parameters:**

file

### clone

```
public IMediaIndexItem clone()
```

Make a clone copy of media index

**Returns:**

shallow copy of media index

### getLastAccessed

```
public long getLastAccessed()
```

Get last time media index was accessed (milliseconds)

**Returns:**

time media index was accessed (milliseconds)

(continued from last page)

## setLastAccessed

```
public void setLastAccessed(long time)
```

Set last time media index accessed (milliseconds)

### Parameters:

time - media index was accessed (milliseconds)

---

## isHitEnd

```
public boolean isHitEnd()
```

Is media index complete

### Returns:

true if media index is complete

---

## touch

```
public void touch()
```

Set last time accessed to current time



## com.wowza.wms.stream Interface 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="#">DEFAULT_RANDOMACCESSREADER</a> Value: <b>com.wowza.io.DirectRandomAccessReader</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> ( <a href="#">IApplicationInstance</a> appInstance, <a href="#">IMediaStream</a> stream, String mediaReadType, String basePath, String mediaName) Initialize mediaReader
boolean	<a href="#">isOpen()</a> is file open
void	<a href="#">open()</a> (String basePath, String name) Open the file for reading
void	<a href="#">rewind()</a> rewind file to start
PlaylistSeekResult	<a href="#">seek()</a> (long timecode, int seektype) seek to timecode in file
int	<a href="#">sendZeroLengthPacket()</a> (int type, int timecode, boolean isAbsolute, <a href="#">IMediaStream</a> stream, java.io.OutputStream out, <a href="#">AMFObj</a> wmsObj, PlaylistCursor flvCursor, PlaylistWriteControl control, PlaylistReaderWriteResults results, long[] sizes, boolean isForceTCZero)
void	<a href="#">setMediaReaderItem()</a> (MediaReaderItem mediaReaderItem) Set the mediaReader item definition
void	<a href="#">setProperties()</a> ( <a href="#">WMSProperties</a> properties) Set the properties for this media reader

void	<a href="#">setStreamPosition</a> ( <a href="#">IMediaReaderStreamPosition</a> pos) Set the file position within the media file
void	<a href="#">startPlayback</a> () Called each time the player being playback (before the buffer it filled)
int	<a href="#">writeGeneratedKeyFrame</a> ( <a href="#">IMediaStream</a> stream, java.io.OutputStream out, <a href="#">AMFObj</a> wmsObjAudio, <a href="#">AMFObj</a> wmsObjVideo, <a href="#">AMFObj</a> wmsObjData, <a href="#">PlaylistCursor</a> flvCursor, <a href="#">PlaylistWriteControl</a> control, <a href="#">PlaylistReaderWriteResults</a> results, long[] sizes, <a href="#">FastPlaySettings</a> fastPlaySettings, boolean isForceTCZero) From current location in file generate a key frame (enhanced seek) and write it to out
int	<a href="#">writePackets</a> ( <a href="#">IMediaStream</a> stream, java.io.OutputStream out, <a href="#">AMFObj</a> wmsObjAudio, <a href="#">AMFObj</a> wmsObjVideo, <a href="#">AMFObj</a> wmsObjData, <a href="#">PlaylistCursor</a> flvCursor, <a href="#">PlaylistWriteControl</a> control, <a href="#">PlaylistReaderWriteResults</a> results, long[] sizes, <a href="#">FastPlaySettings</a> fastPlaySettings, boolean isForceTCZero) From current location in file write packets to output
int	<a href="#">writePackets</a> (java.util.List packetList, <a href="#">PlaylistCursor</a> flvCursor, <a href="#">PlaylistWriteControl</a> control, <a href="#">PlaylistReaderWriteResults</a> results, long[] sizes, <a href="#">FastPlaySettings</a> fastPlaySettings) Write packets to the packetList as AMFPackets

## Fields

### DEFAULT\_RANDOMACCESSREADER

```
public static final java.lang.String DEFAULT_RANDOMACCESSREADER
```

Constant value: **com.wowza.io.DirectRandomAccessReader**

### SEEKTARGET\_VIDEOKEYFRAME

```
public static final int SEEKTARGET_VIDEOKEYFRAME
```

Constant value: **1**

### SEEKTARGET\_AUDIO

```
public static final int SEEKTARGET_AUDIO
```

Constant value: **3**

### SEEKTARGET\_ENHANCED

```
public static final int SEEKTARGET_ENHANCED
```

Constant value: **4**

### PLAYEVENT\_STARTPLAYBACK

```
public static final int PLAYEVENT_STARTPLAYBACK
```

(continued from last page)

Constant value: **1**

---

## PLAYEVENT\_BEFOREMETADATA

```
public static final int PLAYEVENT_BEFOREMETADATA
```

Constant value: **2**

---

## PLAYEVENT\_AFTERMETADATA

```
public static final int PLAYEVENT_AFTERMETADATA
```

Constant value: **3**

---

## PLAYEVENT\_BEFOREBUFFERFILL

```
public static final int PLAYEVENT_BEFOREBUFFERFILL
```

Constant value: **4**

---

## PLAYEVENT\_AFTERBUFFERFILL

```
public static final int PLAYEVENT_AFTERBUFFERFILL
```

Constant value: **5**

---

## SEEK\_KEYUP

```
public static final int SEEK_KEYUP
```

Seek direction: up to closets key frame

Constant value: **1**

---

## 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**

(continued from last page)

## Methods

### init

```
public void init(IApplicationInstance appInstance,  
                IMediaStream stream,  
                String mediaReadType,  
                String basePath,  
                String mediaName)
```

Initialize mediaReader

**Parameters:**

stream - parent stream  
mediaReadType - media reader type id. Example: flv  
basePath - Base path for application instance  
mediaName - name of the media item

---

### setMediaReaderItem

```
public void setMediaReaderItem(MediaReaderItem mediaReaderItem)
```

Set the mediaReader item definition

**Parameters:**

mediaReaderItem

---

### open

```
public void open(String basePath,  
                String name)
```

Open the file for reading

**Parameters:**

basePath - base path to file  
name - file name without extension

---

### getMetadata

```
public java.util.List getMetadata()
```

Get a collection of metadata packets in ByteBuffers for this file. You can use 'new AMFDataList(data)' to convert to AMF objects.

**Returns:**

collection of metadata packets in ByteBuffers

---

### rewind

```
public void rewind()
```

rewind file to start

---

### close

```
public void close()
```

close file

---

## isOpen

```
public boolean isOpen()
```

is file open

### Returns:

is file open

---

## seek

```
public PlaylistSeekResult seek(long timecode,  
    int seektype)
```

seek to timecode in file

### Parameters:

timecode - timecode (milliseconds) to seek to  
seektype - seek type IMediaReader.SEEK\_\*

### Returns:

detailed results object of seek or null if failure

---

## writePackets

```
public int writePackets(IMediaStream stream,  
    java.io.OutputStream out,  
    AMFObj wmsObjAudio,  
    AMFObj wmsObjVideo,  
    AMFObj wmsObjData,  
    PlaylistCursor flvCursor,  
    PlaylistWriteControl control,  
    PlaylistReaderWriteResults results,  
    long[] sizes,  
    FastPlaySettings fastPlaySettings,  
    boolean isForceTCZero)
```

From current location in file write packets to output

### 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

---

(continued from last page)

## writePackets

```
public int writePackets(java.util.List packetList,
    PlaylistCursor flvCursor,
    PlaylistWriteControl control,
    PlaylistReaderWriteResults results,
    long[] sizes,
    FastPlaySettings fastPlaySettings)
```

Write packets to the packetList as AMFPackets

### Parameters:

packetList - list to which to write packets  
 flvCursor - current cursor that keeps track of timecodes and flags indicating position in media file  
 control - write control flags. Mainly used to control when it should stop writing  
 results - write results returned from this routine  
 sizes - array of values that keep track of number of bytes written and number of packets for each data type (audio, video, data). See IMediaStreamPlay.PLAYSIZES\_\*  
 fastPlaySettings - current fastPlay settings for the stream

### Returns:

number of bytes written

## writeGeneratedKeyFrame

```
public int writeGeneratedKeyFrame(IMediaStream stream,
    java.io.OutputStream out,
    AMFObj wmsObjAudio,
    AMFObj wmsObjVideo,
    AMFObj wmsObjData,
    PlaylistCursor flvCursor,
    PlaylistWriteControl control,
    PlaylistReaderWriteResults results,
    long[] sizes,
    FastPlaySettings fastPlaySettings,
    boolean isForceTCZero)
```

From current location in file generate a key frame (enhanced seek) and write it to out

### Parameters:

stream - parent mediaStream  
 out - output stream  
 wmsObjAudio - audio channel amfObj  
 wmsObjVideo - video channel amfObj  
 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

(continued from last page)

## sendZeroLengthPacket

```
public int sendZeroLengthPacket(int type,  
    int timecode,  
    boolean isAbsolute,  
    IMediaStream stream,  
    java.io.OutputStream out,  
    AMFObj wmsObj,  
    PlaylistCursor flvCursor,  
    PlaylistWriteControl control,  
    PlaylistReaderWriteResults results,  
    long[] sizes,  
    boolean isForceTCZero)
```

---

## startPlayback

```
public void startPlayback()
```

Called each time the player being playback (before the buffer it filled)

---

## getDuration

```
public long getDuration()
```

Get duration or time (milliseconds) of the media file

**Returns:**

duration or time (milliseconds) of the media file

---

## getLength

```
public long getLength()
```

Get the stream length in bytes

**Returns:**

stream length in bytes

---

## getPath

```
public String getPath()
```

Get abstract path to the media item

**Returns:**

abstract path to the media item

---

## getMediaExtension

```
public String getMediaExtension()
```

Get media extension

**Returns:**

media extension



(continued from last page)

## getStreamPosition

```
public IMediaReaderStreamPosition getStreamPosition()
```

Get a reference to the current stream position

**Returns:**

stream position

---

## setStreamPosition

```
public void setStreamPosition(IMediaReaderStreamPosition pos)
```

Set the file position within the media file

**Parameters:**

pos - stream position

---

## setProperty

```
public void setProperty(WMSPProperties properties)
```

Set the properties for this media reader

**Parameters:**

properties - properties

---

## com.wowza.wms.stream Interface IMediaReaderActionNotify

public interface **IMediaReaderActionNotify**  
extends

### Method Summary

void	<a href="#"><u>onMediaReaderClose</u></a> ( <a href="#"><u>IMediaReader</u></a> mediaReader, <a href="#"><u>IMediaStream</u></a> stream)
void	<a href="#"><u>onMediaReaderCreate</u></a> ( <a href="#"><u>IMediaReader</u></a> mediaReader)
void	<a href="#"><u>onMediaReaderExtractMetaData</u></a> ( <a href="#"><u>IMediaReader</u></a> mediaReader, <a href="#"><u>IMediaStream</u></a> stream)
void	<a href="#"><u>onMediaReaderInit</u></a> ( <a href="#"><u>IMediaReader</u></a> mediaReader, <a href="#"><u>IMediaStream</u></a> stream)
void	<a href="#"><u>onMediaReaderOpen</u></a> ( <a href="#"><u>IMediaReader</u></a> mediaReader, <a href="#"><u>IMediaStream</u></a> stream)

### Methods

#### onMediaReaderCreate

public void **onMediaReaderCreate**([IMediaReader](#) mediaReader)

#### onMediaReaderInit

public void **onMediaReaderInit**([IMediaReader](#) mediaReader, [IMediaStream](#) stream)

#### onMediaReaderOpen

public void **onMediaReaderOpen**([IMediaReader](#) mediaReader, [IMediaStream](#) stream)

#### onMediaReaderExtractMetaData

public void **onMediaReaderExtractMetaData**([IMediaReader](#) mediaReader, [IMediaStream](#) stream)

(continued from last page)

## onMediaReaderClose

```
public void onMediaReaderClose(IMediaReader mediaReader,  
    IMediaStream stream)
```

---

## com.wowza.wms.stream Interface IMediaReaderStreamPosition

---

public interface **IMediaReaderStreamPosition**  
extends

---

### Method Summary

boolean	<a href="#">isValid()</a>
---------	---------------------------

---

### Methods

#### **isValid**

public boolean **isValid()**

## com.wowza.wms.stream Interface IMediaStream

public interface **IMediaStream**  
extends

IMediaStream: public, generic interface to mediaStream object. Upon creation of a new mediaStream, the client connection's default streamType value will be used to dynamically create a mediaStream object based on the definitions available in the vHosts.mediaStreamMap. All mediaStream implementations implement this interface.

### Field Summary

public static final	<a href="#">AUDIOSAMPLEACCESS</a> Value: <b>2</b>
public static final	<a href="#">READACCESS</a> Value: <b>0</b>
public static final	<a href="#">VIDEOSAMPLEACCESS</a> Value: <b>3</b>
public static final	<a href="#">WRITEACCESS</a> Value: <b>1</b>

### Method Summary

void	<a href="#">addAudioCodecConfigPacket</a> (long timecode, <a href="#">AMFPacket</a> packet) Set audio codec configuration packet (needed for H.264/AAC playback)
void	<a href="#">addAudioData</a> (byte[] data, int offset, int size) Add data to curent audio packet
void	<a href="#">addClientListener</a> ( <a href="#">IMediaStreamActionNotify</a> actionListener) Add client listener.
void	<a href="#">addClientListener</a> ( <a href="#">IMediaStreamActionNotify2</a> actionListener) Add client listener.
void	<a href="#">addDataData</a> (byte[] data, int offset, int size) Add data to curent data packet
void	<a href="#">addVideoCodecConfigPacket</a> (long timecode, <a href="#">AMFPacket</a> packet) Set video codec configuration packet (needed for H.264/AAC playback)
void	<a href="#">addVideoData</a> (byte[] data, int offset, int size) Add data to curent video packet
void	<a href="#">clear</a> () Delete media file pointed to by this mediaStream (be careful)

void	<a href="#"><u>clearFastPlaySettings()</u></a> Clear fastPlay settings
void	<a href="#"><u>clearLoggingValues()</u></a>
void	<a href="#"><u>close()</u></a> Close mediaStream
void	<a href="#"><u>flush()</u></a> Force publishing packets to be flushed from the input buffers to the output buffers
boolean[]	<a href="#"><u>getAccess()</u></a> ( <a href="#"><u>IClient</u></a> client, String name) Get the read/write access to this stream for this client
<a href="#"><u>AMFPacket</u></a>	<a href="#"><u>getAudioCodecConfigPacket()</u></a> (long timecode) Get audio codec configuration packet (needed for H.264/AAC playback)
int	<a href="#"><u>getAudioMissing()</u></a> Get number of audio bytes missing from current audio packet
int	<a href="#"><u>getAudioSize()</u></a> Get the size of the current audio packet that is being streamed from the client to the server
long	<a href="#"><u>getAudioTC()</u></a> Get last absolute audio timecode (milliseconds) sent to mediaStream
int	<a href="#"><u>getBufferTime()</u></a> Get buffer time for mediaStream (milliseconds)
byte[]	<a href="#"><u>getBurstStartStop()</u></a> (boolean isStart) Get the dynamic streaming burst start/stop AMF packet
String	<a href="#"><u>getCacheName()</u></a> not used
<a href="#"><u>IClient</u></a>	<a href="#"><u>getClient()</u></a> Get parent client connection
int	<a href="#"><u>getClientId()</u></a> Get parent client connection (id)
String	<a href="#"><u>getContextStr()</u></a> Returns the stream context string in the form [application]/[appInstance]/[streamName].
int	<a href="#"><u>getDataMissing()</u></a> Get number of data bytes missing from current audio packet
int	<a href="#"><u>getDataSize()</u></a> Get the size of the current data packet that is being streamed from the client to the server
long	<a href="#"><u>getDataTC()</u></a> Get last absolute data timecode (milliseconds) sent to mediaStream
int	<a href="#"><u>getDataType()</u></a> Get the data packet type: (IVHost.CONTENTTYPE_DATA0 or IVHost.CONTENTTYPE_DATA3)
<a href="#"><u>ElapsedTimer</u></a>	<a href="#"><u>getElapsedTime()</u></a> Get the interface to the elapse timer

String	<a href="#"><u>getExt()</u></a> Get media file extension
<a href="#"><u>FastPlaySettings</u></a>	<a href="#"><u>getFastPlaySettings()</u></a> Get current fastPlay settings
int	<a href="#"><u>getHeaderSize()</u></a> Get the last packet header size (debugging)
<a href="#"><u>IHTTPStreamerSession</u></a>	<a href="#"><u>getHTTPStreamerSession()</u></a> Get the HTTPStreamer session associated with this stream
<a href="#"><u>AMFPacket</u></a>	<a href="#"><u>getLastKeyFrame()</u></a> Get most recent video key frame
<a href="#"><u>AMFPacket</u></a>	<a href="#"><u>getLastPacket()</u></a> Get most recent live packet
String	<a href="#"><u>getLiveStreamPacketizer()</u></a> Get the live stream packetizer that this stream is using
ILiveStreamPacketizer	<a href="#"><u>getLiveStreamPacketizer(String name)</u></a> Get the LiveStreamPacketizer interface to a stream by name
String	<a href="#"><u>getLiveStreamPacketizerList()</u></a> Get the comma separated list of LiveStreamPacketizers names being used by this stream (see conf/LiveStreamPacketizers.xml)
String	<a href="#"><u>getLiveStreamRepeater()</u></a> Get the live stream repeater name for the stream
long	<a href="#"><u>getMaxTimecode()</u></a> Get the timecode of the latest received packet
<a href="#"><u>IOPerformanceCounter</u></a>	<a href="#"><u>getMediaIOPerformance()</u></a> Get IO performance counter
<a href="#"><u>IMediaStreamMetaDataProvider</u></a>	<a href="#"><u>getMetaDataProvider()</u></a> Get the metaData provider
String	<a href="#"><u>getName()</u></a> Get stream name
com.wowza.wms.netconnection.INetConnection	<a href="#"><u>getNetConnection()</u></a> Get parent netConnection (future server to server communication)
<a href="#"><u>IMediaStreamPlay</u></a>	<a href="#"><u>getPlayer()</u></a> Get underlying player (IMediaStreamPlay) object
java.util.List	<a href="#"><u>getPlayPackets()</u></a> Get all available live packets
<a href="#"><u>WMSProperties</u></a>	<a href="#"><u>getProperties()</u></a> Get mediaStream properties
int	<a href="#"><u>getPublishAudioCodecId()</u></a> Get the codec id of the most recently published audio packet
int	<a href="#"><u>getPublishVideoCodecId()</u></a> Get the codec id of the most recently published video packet

String	<a href="#"><u>getQueryStr()</u></a> Get play/publish name query string.
int	<a href="#"><u>getReceiveVideoFPS()</u></a> Set frame per seconds for video (not currently implemented)
<a href="#"><u>AMFObj</u></a>	<a href="#"><u>getRespAMFAudioObj()</u></a> Get audio response channel object
<a href="#"><u>AMFObj</u></a>	<a href="#"><u>getRespAMFDataObj()</u></a> Get data response channel object
<a href="#"><u>AMFObj</u></a>	<a href="#"><u>getRespAMFVideoObj()</u></a> Get video response channel object
RTPStream	<a href="#"><u>getRTPStream()</u></a> Get the RTP based stream this stream is associated with
int	<a href="#"><u>getSrc()</u></a> Get the stream id
java.io.File	<a href="#"><u>getStreamFileForRead()</u></a> Get the File object to read from a stream (get stream name, ext and query from stream object)
java.io.File	<a href="#"><u>getStreamFileForRead(String name, String ext, String query)</u></a> Get the File object to read from a stream (specify name, ext and query)
java.io.File	<a href="#"><u>getStreamFileForWrite()</u></a> Get the File object to write to a stream (get stream name, ext and query from stream object)
java.io.File	<a href="#"><u>getStreamFileForWrite(String name, String ext, String query)</u></a> Get the File object to write to a stream (specify name, ext and query)
<a href="#"><u>MediaStreamMap</u></a>	<a href="#"><u>getStreams()</u></a> Get parent mediaStreamMap (owned by applicationInstance)
String	<a href="#"><u>getStreamType()</u></a> Get mediaStream streamType
String	<a href="#"><u>getUniqueStreamIdStr()</u></a> Get a string that uniquely identifies this stream
<a href="#"><u>AMFPacket</u></a>	<a href="#"><u>getVideoCodecConfigPacket(long timecode)</u></a> Get video codec configuration packet (needed for H.264/AAC playback)
int	<a href="#"><u>getVideoMissing()</u></a> Get number of video bytes missing from current audio packet
int	<a href="#"><u>getVideoSize()</u></a> Get the size of the current video packet that is being streamed from the client to the server
long	<a href="#"><u>getVideoTC()</u></a> Get last absolute video timecode (milliseconds) sent to mediaStream
void	<a href="#"><u>handleCallback(RequestFunction function)</u></a> Routes request function to callback handler onStatus, onPlayStatus or [method/handler]
boolean	<a href="#"><u>idle()</u></a> Allow the mediaStream to perform idle work (not currently in use)



long	<a href="#"><u>incrementMediaInBytes</u></a> (long increment) Increment the number of mediaStream bytes received
long	<a href="#"><u>incrementMediaLossBytes</u></a> (long bytes, long count) Increment the number of mediaStream loss bytes sent and number of packets sent
long	<a href="#"><u>incrementMediaOutBytes</u></a> (long bytes, long count) Increment the number of mediaStream bytes sent and number of packets sent
void	<a href="#"><u>init</u></a> ( <a href="#"><u>MediaStreamMap</u></a> parent, int src, <a href="#"><u>WMSProperties</u></a> properties) Initialize the mediaStream object after creation.
void	<a href="#"><u>initLiveStreamRepeating</u></a> (String liveStreamPacketizer, String liveStreamRepeater) Initialize this stream for live stream repeating
boolean	<a href="#"><u>isAppend</u></a> ( ) Is append to media file (only valid if isRecord)
boolean	<a href="#"><u>isClustered</u></a> ( ) not used
boolean	<a href="#"><u>isMediaCasterPlay</u></a> ( ) Is MediaCaster play enabled (if true, will trigger MediaCaster startup)
boolean	<a href="#"><u>isOpen</u></a> ( ) Is mediaStream open
boolean	<a href="#"><u>isPlay</u></a> ( ) Is the stream a play stream (vs a publish stream)
boolean	<a href="#"><u>isPlaying</u></a> ( ) Is mediaStream playing (or paused - false)
boolean	<a href="#"><u>isPublishStreamReady</u></a> (boolean checkAudio, boolean checkVideo) Returns true if the publishing stream contains enough video/audio data to start playback
boolean	<a href="#"><u>isReceiveAudio</u></a> ( ) Is client currently receiving audio.
boolean	<a href="#"><u>isReceiveVideo</u></a> ( ) Is client currently receiving video.
boolean	<a href="#"><u>isRecord</u></a> ( ) Is this stream being recorded to a file
boolean	<a href="#"><u>isSendPlayStopLogEvent</u></a> ( ) Get need to send a log event for stop
boolean	<a href="#"><u>isSendPublishStopLogEvent</u></a> ( ) Get need to send a log event for publishing
boolean	<a href="#"><u>isSendRecordStopLogEvent</u></a> ( ) Get need to send a log event for recording
double	<a href="#"><u>length</u></a> ( ) Get length/duration (seconds) of media file pointed to by mediaStream
void	<a href="#"><u>notifyActionOnMetaData</u></a> ( <a href="#"><u>AMFPacket</u></a> metaDataPacket) Notify client listeners of onMetaData change

void	<a href="#"><u>notifyActionPause</u></a> (boolean isPause, long location) Notify client listeners of pause action
void	<a href="#"><u>notifyActionPauseRaw</u></a> (boolean isPause, long location) Notify client listeners of pauseRaw action
void	<a href="#"><u>notifyActionPlay</u></a> (String streamName, double playStart, double playLen, int playReset) Notify client listeners of play action
void	<a href="#"><u>notifyActionPublish</u></a> (String streamName, boolean isRecord, boolean isAppend) Notify client listeners of publish action
void	<a href="#"><u>notifyActionSeek</u></a> (double location) Notify client listeners of seek action
void	<a href="#"><u>notifyActionStop</u></a> ( ) Notify client listeners of stop action
void	<a href="#"><u>notifyActionUnPublish</u></a> (String streamName, boolean isRecord, boolean isAppend) Notify client listeners of unpublish action
void	<a href="#"><u>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>registerCallback</u></a> (String handlerName, <a href="#"><u>IMediaStreamCallback</u></a> callback) Register a callback handler
void	<a href="#"><u>registerOnPlayStatus</u></a> ( <a href="#"><u>IMediaStreamCallback</u></a> callback) Register onPlayStatus handler
void	<a href="#"><u>registerOnStatus</u></a> ( <a href="#"><u>IMediaStreamCallback</u></a> callback) Register onStatus handler
void	<a href="#"><u>removeClientListener</u></a> ( <a href="#"><u>IMediaStreamActionNotify</u></a> actionListener) Remove client listener.
void	<a href="#"><u>removeClientListener</u></a> ( <a href="#"><u>IMediaStreamActionNotify2</u></a> actionListener) Remove client listener.
void	<a href="#"><u>send</u></a> (String handlerName) Call client side NetStream method/handler with no parameters
void	<a href="#"><u>send</u></a> (String handlerName, Object[] params) Call client side NetStream method/handler
int	<a href="#"><u>sendControlBytes</u></a> (int controlType, java.io.OutputStream out) Send playback control bytes.
void	<a href="#"><u>sendDirect</u></a> (String handlerName) Call client side NetStream method/handler and send event to underlying stream (will record event)
void	<a href="#"><u>sendDirect</u></a> (String handlerName, Object[] params) Call client side NetStream method/handler and send event to underlying stream (will record event)

int	<a href="#"><u>sendLivePlaySeek</u></a> ( java.io.OutputStream out, String name, long timecode) Send onStatus(NetStream.Seek.Notify) event
int	<a href="#"><u>sendLivePlayStart</u></a> ( java.io.OutputStream out, String name, long timecode, long timecodeOffset) Send onStatus(NetStream.Play.Start) event
int	<a href="#"><u>sendLivePlaySwitch</u></a> ( java.io.OutputStream out, String name, long timecode) Send onStatus(NetStream.Play.Transition) event
int	<a href="#"><u>sendPauseNotify</u></a> (long timecode, String name) Send onStatus(NetStream.Pause.Notify) event
int	<a href="#"><u>sendPauseNotify</u></a> ( java.io.OutputStream out, long timecode, String name) Send onStatus(NetStream.Pause.Notify) event.
int	<a href="#"><u>sendPlayReset</u></a> ( java.io.OutputStream out, String name) Send onStatus(NetStream.Play.Reset) event.
int	<a href="#"><u>sendPlayReset</u></a> (String name) Send onStatus(NetStream.Play.Reset) event
int	<a href="#"><u>sendPlaySeek</u></a> (long location, long seekLocation, String name) Send onStatus(NetStream.Seek.Notify) event.
int	<a href="#"><u>sendPlaySeek</u></a> ( java.io.OutputStream out, long location, long seekLocation, String name) Send onStatus(NetStream.Seek.Notify) event.
int	<a href="#"><u>sendPlaySeek</u></a> ( java.io.OutputStream out, long location, long seekLocation, String name, java.util.List seekTypes) Send onStatus(NetStream.Seek.Notify) event.
int	<a href="#"><u>sendPlayStart</u></a> ( java.io.OutputStream out, String name, boolean isSwitch, boolean isLive, long timecode, java.util.List seekTypes) Send onStatus(NetStream.Play.Start) event (it is not a typo, this also send play start but with different control codes for media switch).
int	<a href="#"><u>sendPlayStart</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>sendPlayStart</u></a> ( java.io.OutputStream out, String name, boolean isSwitch, long timecode, java.util.List seekTypes) Send onStatus(NetStream.Play.Start) event (it is not a typo, this also send play start but with different control codes for media switch).
int	<a href="#"><u>sendPlayStart</u></a> (String name, long timecode) Send onStatus(NetStream.Play.Start) event
int	<a href="#"><u>sendPlayStatus</u></a> (long timecode, int statusType, double duration, double bytesSent) Send onPlayStatus(NetStream.Play.Switch, NetStream.Play.Complete, NetStream.Play.Stop) event
int	<a href="#"><u>sendPlayStatus</u></a> ( java.io.OutputStream out, long timecode, int statusType, double duration, double bytesSent) Send onPlayStatus(NetStream.Play.Switch, NetStream.Play.Complete, NetStream.Play.Stop) event.

int	<a href="#"><code>sendPlayStop</code></a> (long location, String name) Send onStatus(NetStream.Play.Stop) event
int	<a href="#"><code>sendPlayStop</code></a> (java.io.OutputStream out, long location, String name) Send onStatus(NetStream.Play.Stop) event.
int	<a href="#"><code>sendPlaySwitch</code></a> (java.io.OutputStream out, String name, boolean isSwitch, long timecode) Send onStatus(NetStream.Play.Start) event (it is not a typo, this also send play start but with different control codes for media switch).
int	<a href="#"><code>sendPlaySwitch</code></a> (String name, long timecode) Send onStatus(NetStream.Play.Start) event (it is not a typo, this also send play start but with different control codes for media switch).
int	<a href="#"><code>sendStreamNotFound</code></a> (java.io.OutputStream out, String name) Send onStatus(NetStream.Play.StreamNotFound) event.
int	<a href="#"><code>sendStreamNotFound</code></a> (String name) Send onStatus(NetStream.Play.StreamNotFound) event
int	<a href="#"><code>sendUnpauseNotify</code></a> (long location, String name) Send onStatus(NetStream.Unpause.Notify) event
int	<a href="#"><code>sendUnpauseNotify</code></a> (java.io.OutputStream out, long location, String name) Send onStatus(NetStream.Unpause.Notify) event.
int	<a href="#"><code>sendUnpauseNotify</code></a> (java.io.OutputStream out, long location, String name, java.util.List seekTypes) Send onStatus(NetStream.Unpause.Notify) event
int	<a href="#"><code>sendVODPlaySwitch</code></a> (java.io.OutputStream out, String name, long timecode) Send onStatus(NetStream.Play.Transition) event
void	<a href="#"><code>setAppend</code></a> (boolean isAppend) Set is append to media file (only valid if isRecord)
void	<a href="#"><code>setAudioSize</code></a> (int audioSize) Set the size of the current audio packet that is being streamed from the client to the server
void	<a href="#"><code>setAudioTC</code></a> (long audioTC) Set last absolute audio timecode (milliseconds) sent to mediaStream
void	<a href="#"><code>setAudioTC</code></a> (long audioTC, boolean isAbsolute) Set last absolute audio timecode (milliseconds) sent to mediaStream
void	<a href="#"><code>setBufferTime</code></a> (int bufferTime) Set buffer time for mediaStream (milliseconds)
void	<a href="#"><code>setClient</code></a> (IClient client) Set parent client connection
void	<a href="#"><code>setClustered</code></a> (boolean isClustered) not used
void	<a href="#"><code>setDataSize</code></a> (int dataSize) Set the size of the current data packet that is being streamed from the client to the server
void	<a href="#"><code>setDataTC</code></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>setExt</u></a> (String ext) Set media file extension
void	<a href="#"><u>setFastPlaySettings</u></a> ( <a href="#"><u>FastPlaySettings</u></a> fastPlaySettings) Set fastPlay settings
void	<a href="#"><u>setHeaderSize</u></a> (int headerSize) Set the last packet header size (debugging)
void	<a href="#"><u>setHTTPStreamerSession</u></a> ( <a href="#"><u>IHTTPStreamerSession</u></a> httpStreamerSession) Set the HTTPStreamer session associated with this stream
void	<a href="#"><u>setIsPlaying</u></a> (boolean isPlaying) Set is mediaStream playing
void	<a href="#"><u>setLiveStreamPacketizer</u></a> (String liveStreamPacketizer) Set the live stream packetizer that this stream is using
void	<a href="#"><u>setLiveStreamPacketizerList</u></a> (String liveStreamPacketizerList) Set the comma separated list of LiveStreamPacketizers names being used by this stream (see conf/LiveStreamPacketizers.xml)
void	<a href="#"><u>setLiveStreamRepeater</u></a> (String liveStreamRepeater) Set the live stream repeater name for the stream
void	<a href="#"><u>setMediaCasterPlay</u></a> (boolean isMediaCasterPlay) Is MediaCaster play enabled (if true, will trigger MediaCaster startup)
void	<a href="#"><u>setMetaDataProvider</u></a> ( <a href="#"><u>IMediaStreamMetaDataProvider</u></a> metaDataProvider) Set the metaData provider
void	<a href="#"><u>setName</u></a> (String name) Set stream name
void	<a href="#"><u>setName</u></a> (String name, String ext) Set stream name and extension.
void	<a href="#"><u>setName</u></a> (String name, String oldName, String ext, String queryStr, double playStart, double playLen, int playTransition) Set stream name, extension, query string, play start, play len and play reset from play command.
void	<a href="#"><u>setNetConnection</u></a> (com.wowza.wms.netconnection.INetConnection netConnection) Set parent netConnection (future server to server communication)
void	<a href="#"><u>setOpen</u></a> (boolean isOpen) Set mediaStream open
void	<a href="#"><u>setPlay</u></a> (boolean isPlay) Set is the stream a play stream (vs a publish stream)
void	<a href="#"><u>setPlayer</u></a> ( <a href="#"><u>IMediaStreamPlay</u></a> player) Set underlying player (IMediaStreamPlay) object

void	<a href="#"><u>setPublishAudioCodecId</u></a> (int publishAudioCodecId) Set the codec id of the most recently published audio packet
void	<a href="#"><u>setPublishVideoCodecId</u></a> (int publishVideoCodecId) Set the codec id of the most recently published video packet
void	<a href="#"><u>setQueryStr</u></a> (String queryStr) Set play/publish name query string.
void	<a href="#"><u>setReceiveAudio</u></a> (boolean receiveAudio) Set receive audio
void	<a href="#"><u>setReceiveVideo</u></a> (boolean receiveVideo) Set receive video
void	<a href="#"><u>setReceiveVideoFPS</u></a> (int receiveVideoFPS) Set frame per second for video (not currently implemented)
void	<a href="#"><u>setRecord</u></a> (boolean isRecord) Set is the stream being recorded
void	<a href="#"><u>setRTPStream</u></a> (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>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="#"><u>IMediaStreamCallback</u></a> callback) Unregister onPlayStatus handler
void	<a href="#"><u>unregisterOnStatus</u></a> ( <a href="#"><u>IMediaStreamCallback</u></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

### 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)

**Parameters:**

bufferTime - buffer time (milliseconds)

---

#### isPlaying

```
public boolean isPlaying()
```

Is mediaStream playing (or paused - false)

**Returns:**

is playing

---

#### setIsPlaying

```
public void setIsPlaying(boolean isPlaying)
```

Set is mediaStream playing

**Parameters:**

isPlaying



## getAudioSize

```
public int getAudioSize()
```

Get the size of the current audio packet that is being streamed from the client to the server

**Returns:**  
size

---

## getVideoSize

```
public int getVideoSize()
```

Get the size of the current video packet that is being streamed from the client to the server

**Returns:**  
size

---

## getDataSize

```
public int getDataSize()
```

Get the size of the current data packet that is being streamed from the client to the server

**Returns:**  
size

---

## setAudioSize

```
public void setAudioSize(int audioSize)
```

Set the size of the current audio packet that is being streamed from the client to the server

**Parameters:**  
audioSize

---

## setVideoSize

```
public void setVideoSize(int videoSize)
```

Set the size of the current video packet that is being streamed from the client to the server

**Parameters:**  
videoSize

---

## setDataSize

```
public void setDataSize(int dataSize)
```

Set the size of the current data packet that is being streamed from the client to the server

**Parameters:**  
dataSize

---

## getSrc

```
public int getSrc()
```

---

(continued from last page)

Get the stream id

**Returns:**

stream id

---

**setSrc**

```
public void setSrc(int src)
```

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

(continued from last page)

queryStr - query string  
playStart - play start  
playLen - play len  
playTransition - play transition (see MediaBase.PLAYTRANSITION\_\*)

---

## 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)

## 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)

---

## 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 curent 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 curent 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 curent 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

---

**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.

---

**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

---

**send**

```
public void send(String handlerName,  
    Object[] params)
```

Call client side NetStream method/handler

**Parameters:**

---

(continued from last page)

handlerName - handler name

params - variable list of parameters. All parameters will be wrapped in AMFDataObj.

---

## 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

---

## 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()
```

---

(continued from last page)

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:

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

(continued from last page)

## 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:**

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( )
```

(continued from last page)

Trim mediaStream. This method will remove live packets that are older than the live buffer size.

---

## handleCallback

```
public void handleCallback(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

(continued from last page)

## 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

---

## addClientListener

```
public void addClientListener(IMediaStreamActionNotify2 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

---

## 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

---

## 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

### 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()
```



(continued from last page)

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

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)

---

## 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()
```

---

(continued from last page)

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:**

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

(continued from last page)

## 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:**

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

(continued from last page)

**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)

---

**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( )
```

(continued from last page)

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:**

liveStreamPacketizerList - comma separated list of LiveStreamPacketizers names

---

## getLiveStreamPacketizer

```
public ILiveStreamPacketizer getLiveStreamPacketizer(String name)
```

Get the LiveStreamPacketizer interface to a stream by name

**Parameters:**

name - LiveStreamPacketizer name

**Returns:**

LiveStreamPacketizer 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)
```

---



(continued from last page)

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

---

## 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

---

(continued from last page)

**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

---

## com.wowza.wms.stream Interface IMediaStreamActionNotify

All Subinterfaces:

[IMediaStreamActionNotify2](#)

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](#)

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 IMediaStreamCallback

public interface **IMediaStreamCallback**  
extends

IMediaStreamCallback: callback interface used by IMediaStream registerCallback, registerOnStatus, registerOnPlayStatus

Method Summary	
void	<a href="#">onCallback</a> ( <a href="#">IMediaStream</a> stream, <a href="#">RequestFunction</a> function, <a href="#">AMFDataList</a> params) Triggered on callback event

## Methods

### onCallback

```
public void onCallback(IMediaStream stream,  
    RequestFunction function,  
    AMFDataList params)
```

Triggered on callback event

**Parameters:**

- stream - mediaStream
- function - function that triggered callback
- params - function parameters

## com.wowza.wms.stream Interface IMediaStreamFileMapper

public interface **IMediaStreamFileMapper**  
extends

Interface for mapping a IMediaStream to the underlying file system. Implement this interface and set your class using `IApplicationInstance.setStreamFileMapper(IMediaStreamFileMapper streamFileMapper)`. Your class will then be called each time a stream needs to be mapped to the underlying file system.

### Method Summary

java.io.File	<a href="#"><code>streamToFileForRead(IMediaStream stream)</code></a> Get the File object to read from a stream (get stream name, ext and query from stream object)
java.io.File	<a href="#"><code>streamToFileForRead(IMediaStream stream, String name, String ext, String query)</code></a> Get the File object to read from a stream (specify name, ext and query)
java.io.File	<a href="#"><code>streamToFileForWrite(IMediaStream stream)</code></a> Get the File object to write to a stream (get stream name, ext and query from stream object)
java.io.File	<a href="#"><code>streamToFileForWrite(IMediaStream stream, String name, String ext, String query)</code></a> Get the File object to write to a stream (specify name, ext and query)

### Methods

#### **streamToFileForRead**

```
public java.io.File streamToFileForRead(IMediaStream stream,  
    String name,  
    String ext,  
    String query)
```

Get the File object to read from a stream (specify name, ext and query)

**Parameters:**

stream - stream  
name - stream name  
ext - stream prefix (Ex. mp4:)  
query - query part of stream name (Ex. mystream?param1=value1)

**Returns:**

resultant File object

#### **streamToFileForRead**

```
public java.io.File streamToFileForRead(IMediaStream stream)
```

Get the File object to read from a stream (get stream name, ext and query from stream object)

**Parameters:**



---

(continued from last page)

stream - stream

**Returns:**

resultant File object

---

## streamToFileForWrite

```
public java.io.File streamToFileForWrite(IMediaStream stream,  
    String name,  
    String ext,  
    String query)
```

Get the File object to write to a stream (specify name, ext and query)

**Parameters:**

stream - stream

name - stream name

ext - stream prefix (Ex. mp4:)

query - query part of stream name (Ex. mystream?param1=value1)

**Returns:**

resultant File object

---

## streamToFileForWrite

```
public java.io.File streamToFileForWrite(IMediaStream stream)
```

Get the File object to write to a stream (get stream name, ext and query from stream object)

**Parameters:**

stream - stream

**Returns:**

resultant File object

---

## com.wowza.wms.stream Interface IMediaStreamMediaCaster

---

public interface **IMediaStreamMediaCaster**  
extends

---

### 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

---

**Method Summary**

void	<a href="#"><u>onStreamStart</u></a> (java.util.List metaDataList, long timecode)
------	---

---

**Methods****onStreamStart**

```
public void onStreamStart(java.util.List metaDataList,  
    long timecode)
```

com.wowza.wms.stream

# Interface IMediaStreamNameAliasProvider

All Subinterfaces:  
[IMediaStreamNameAliasProvider2](#)

public interface **IMediaStreamNameAliasProvider**  
extends

Method Summary	
String	<a href="#">resolvePlayAlias</a> ( <a href="#">IApplicationInstance</a> appInstance, String name)
String	<a href="#">resolveStreamAlias</a> ( <a href="#">IApplicationInstance</a> appInstance, String name)

## Methods

### resolvePlayAlias

```
public String resolvePlayAlias(IApplicationInstance appInstance,  
                               String name)
```

### resolveStreamAlias

```
public String resolveStreamAlias(IApplicationInstance appInstance,  
                                  String name)
```

## com.wowza.wms.stream Interface IMediaStreamNameAliasProvider2

All Superinterfaces:

[IMediaStreamNameAliasProvider](#)

public interface **IMediaStreamNameAliasProvider2**

extends [IMediaStreamNameAliasProvider](#)

### Method Summary

String	<a href="#">resolvePlayAlias</a> ( <a href="#">IApplicationInstance</a> appInstance, String name, <a href="#">IClient</a> client)
String	<a href="#">resolvePlayAlias</a> ( <a href="#">IApplicationInstance</a> appInstance, String name, <a href="#">IHTTPStreamerSession</a> httpSession)
String	<a href="#">resolvePlayAlias</a> ( <a href="#">IApplicationInstance</a> appInstance, String name, <a href="#">ILiveStreamPacketizer</a> liveStreamPacketizer)
String	<a href="#">resolvePlayAlias</a> ( <a href="#">IApplicationInstance</a> appInstance, String name, <a href="#">RTPSession</a> rtpSession)
String	<a href="#">resolveStreamAlias</a> ( <a href="#">IApplicationInstance</a> appInstance, String name, <a href="#">IMediaCaster</a> mediaCaster)

Methods inherited from interface [com.wowza.wms.stream.IMediaStreamNameAliasProvider](#)

[resolvePlayAlias](#), [resolveStreamAlias](#)

### Methods

#### **resolvePlayAlias**

```
public String resolvePlayAlias(IApplicationInstance appInstance,  
    String name,  
    IClient client)
```

#### **resolvePlayAlias**

```
public String resolvePlayAlias(IApplicationInstance appInstance,  
    String name,  
    IHTTPStreamerSession httpSession)
```

(continued from last page)

## resolvePlayAlias

```
public String resolvePlayAlias(IApplicationInstance appInstance,  
    String name,  
    RTPSession rtpSession)
```

---

## resolvePlayAlias

```
public String resolvePlayAlias(IApplicationInstance appInstance,  
    String name,  
    ILiveStreamPacketizer liveStreamPacketizer)
```

---

## resolveStreamAlias

```
public String resolveStreamAlias(IApplicationInstance appInstance,  
    String name,  
    IMediaCaster mediaCaster)
```

## 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> ( <a href="#">IMediaStream</a> parent, <a href="#">MediaStreamMap</a> streams) Initialize mediaStreamPlayer
void	<a href="#">initLiveStreamRepeating</a> (String liveStreamPacketizer, String liveStreamRepeater) Initialize this stream for live stream repeating
void	<a href="#">interruptPlay()</a> Interrupt play to perform operation.
double	<a href="#">length()</a> Get stream length/seconds (seconds)
void	<a href="#">pause</a> (int pauseType, long timecode) pause mediaPlayStream
void	<a href="#">pauseRaw</a> (int pauseType, long timecode) pauseRaw mediaPlayStream
boolean	<a href="#">play()</a> Return true if there are packets to play
int	<a href="#">play</a> (java.io.OutputStream out, <a href="#">AMFObj</a> wmsObjAudio, <a href="#">AMFObj</a> wmsObjVideo, <a href="#">AMFObj</a> wmsObjData, long[] sizes) Write new packets or play packets
void	<a href="#">reset</a> (boolean isReset) Reset mediaStreamPlayer
void	<a href="#">resetNoLookup()</a> Reset but do not lookup current position in live stream
void	<a href="#">seek</a> (int location) seek mediaPlayStream

void	<a href="#"><code>setBufferTime</code></a> (int bufferTime) Set buffer time
void	<a href="#"><code>setName</code></a> (String name, String oldName, String ext, String queryStr, double playStart, double playLen, int playTransition) Set mediaPlayStream name, extension, query string, play start, play len, play reset
void	<a href="#"><code>shutdown</code></a> () Shutdown mediaStreamPlayer
long	<a href="#"><code>size</code></a> () Get stream media file size
void	<a href="#"><code>startPlay</code></a> () Start playing stream
void	<a href="#"><code>stopName</code></a> (String name, String oldName, String ext, String queryStr, double playStart, double playLen, int playTransition) Stop stream name
void	<a href="#"><code>switchName</code></a> (String name, String oldName, String ext, String queryStr, double playStart, double playLen, int playTransition) Switch to stream name
void	<a href="#"><code>updateLoggingValues</code></a> () Update internal logging values

## Fields

### PAUSE\_PAUSE

public static final int **PAUSE\_PAUSE**

Pause type: pause  
Constant value: **1**

### PAUSE\_TOGGLE

public static final int **PAUSE\_TOGGLE**

Pause type: toggle  
Constant value: **-1**

### PAUSE\_PLAY

public static final int **PAUSE\_PLAY**

Pause type: play  
Constant value: **0**

### PLAYSTATUSTYPE\_SWITCH

public static final int **PLAYSTATUSTYPE\_SWITCH**

onPlayStatus type: switch  
Constant value: **1**

(continued from last page)

---

## 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**

---

## PLAYSIZES\_LOSS\_BYTES

```
public static final int PLAYSIZES_LOSS_BYTES
```

---

(continued from last page)

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

---

### play

```
public int play(java.io.OutputStream out,  
               AMFObj wmsObjAudio,  
               AMFObj wmsObjVideo,  
               AMFObj wmsObjData,  
               long[] sizes)
```

(continued from last page)

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:**

(continued from last page)

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)
```

Stop stream name

### Parameters:

name - stream name  
oldName - old stream name  
ext - stream extension  
queryStr - query string  
playStart - play start

(continued from last page)

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

---

## 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)
```

---

(continued from last page)

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

---

## 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 using this interface to persist .flv data captured from the Flash client. These classes are referenced in [\[install-dir\]/conf/MediaWriters.xml](#).

### Example IMediaWriter implementation: MediaWriterFLVBasic

This is a basic IMediaWriter implementation that can handle record and append.

```
import java.io.*;
import java.nio.ByteBuffer;
import java.util.*;

import com.wowza.util.*;
import com.wowza.wms.stream.*;
import com.wowza.wms.amf.AMFData;
import com.wowza.wms.logging.*;

public class MediaWriterFLV implements IMediaWriter
{
    private IMediaStream parent = null;
    private MediaWriterItem mediaWriterItem = null;
    private long[] currentTCs = new long[3];
    private long duration = 0;
    private Map extraMetadata = new HashMap();
    private boolean versionFile = false;

    public void setMediaWriterItem(MediaWriterItem mediaWriterItem)
    {
        this.mediaWriterItem = mediaWriterItem;
    }

    public void setParent(IMediaStream parent)
    {
        this.parent = parent;
    }

    public void writePackets(List audioPackets, List videoPackets,
                             List dataPackets, List audioTCs, List videoTCs, List dataTCs, List
dataTypes,
                             boolean isFirst, boolean isLast)
    {
        File newFile = this.parent.getStreamFile();

        boolean localAppend = this.parent.isAppend();

        if (isFirst)
        {
            long startTC = 0;
            if (newFile.exists())
            {
                if (localAppend)
                    startTC = FLVUtils.getLastTC(newFile);
                else
                {
                    if (versionFile)
                        FileUtils.versionFile(newFile);
                    else
                    {
                        try
```

```

        {
            newFile.delete();
        }
        catch (Exception e)
        {
        }
    }
}
else
    localAppend = false;

this.currentTCs[FLVUtils.FLV_TCINDEXAUDIO] = startTC;
this.currentTCs[FLVUtils.FLV_TCINDEXVIDEO] = startTC;
this.currentTCs[FLVUtils.FLV_TCINDEXDATA] = startTC;
}
else
    localAppend = true;

try
{
    if (newFile.getParentFile() == null)

WMSLoggerFactory.getLogger(MediaWriterFLV.class).warn("MediaWriterFLV: File path does not
exist: "+newFile.getPath());
        else if (!newFile.getParentFile().exists())

WMSLoggerFactory.getLogger(MediaWriterFLV.class).warn("MediaWriterFLV: Folder does not exist:
"+newFile.getParentFile().getPath());
        else if (newFile.exists() && !newFile.canWrite())

WMSLoggerFactory.getLogger(MediaWriterFLV.class).warn("MediaWriterFLV: Cannot write to file
(permission error): "+newFile.getPath());

        FileOutputStream ds = new FileOutputStream(newFile, localAppend);

        if (isFirst)
        {
            if (!localAppend)
            {
                FLVUtils.writeHeader(ds, 0.0, extraMetadata);

                boolean writeZeroPacket = true;
                while(true)
                {
                    if (audioPackets.size() == 0)
                        break;

                    ByteBuffer data =
(ByteBuffer)audioPackets.get(0);

                    long tcA = ((Long)audioTCs.get(0)).longValue();

```

```

        if (tcA == 0 && data.limit() == 0)
            writeZeroPacket = false;

        break;
    }

    if (writeZeroPacket)
    {
        FLVUtils.writeChunk(ds, null, 0,

this.currentTCs[FLVUtils.FLV_TCINDEXAUDIO],

                                                                    (byte) 0x08); //
write zero length audio block
    }
}

    FLVUtils.writePackets(ds, audioPackets, videoPackets, dataPackets,
        audioTCs, videoTCs, dataTCs, dataTypes, currentTCs);

    ds.flush();
    ds.close();
}
catch (Exception e)
{
    WMSLoggerFactory.getLogger(MediaWriterFLV.class).error(
        "MediaWriterFLV: Error writing to file:
"+newFile.getPath()+": "+e.toString());
    e.printStackTrace();
}

    if (isLast)
    {
        duration = Math.max(Math.max(currentTCs[FLVUtils.FLV_TCINDEXAUDIO],
            currentTCs[FLVUtils.FLV_TCINDEXVIDEO]),
            currentTCs[FLVUtils.FLV_TCINDEXDATA]);
        double durationSecs = ((double)duration) / 1000.0;

        FLVUtils.writeDuration(newFile, durationSecs);
    }
}

public Map getExtraMetadata()
{
    return extraMetadata;
}

public void setExtraMetadata(Map extraMetadata)
{
    this.extraMetadata = extraMetadata;
}

```

```
    }

    public boolean isVersionFile()
    {
        return versionFile;
    }

    public void setVersionFile(boolean versionFile)
    {
        this.versionFile = versionFile;
    }

    public void putMetaData(String name, AMFData value)
    {
        this.extraMetadata.put(name, value);
    }
}
```

To use this class, edit [install-dir]/conf/MediaWriter and replace the definition for the **flv** MediaWriter:

```
<MediaWriter>
  <Name>flv</Name>
  <Description>FLV Media Writer</Description>
  <FileExtension>flv</FileExtension>
  <ClassBase>com.wowza.wms.plugin.mediawriter.flv.MediaWriterFLVBasic</ClassBase>
</MediaWriter>
```

## Example IMediaWriter implementation: MediaWriterFLVMetadata

This example illustrates how to write custom metadata into the recorded flv file on the fly.

```
public class MediaWriterFLVMetadata implements IMediaWriter
{
    private IMediaStream parent = null;
    private MediaWriterItem mediaWriterItem = null;
    private long[] currentTCs = new long[3];
    private long duration = 0;
    private File tmpFile = null;
    private Map extraMetadata = new HashMap();
    private boolean versionFile = false;

    public void setMediaWriterItem(MediaWriterItem mediaWriterItem)
    {
        this.mediaWriterItem = mediaWriterItem;
    }

    public void setParent(IMediaStream parent)
    {
        this.parent = parent;
    }

    public void writePackets(List audioPackets, List videoPackets,
        List dataPackets, List audioTCs, List videoTCs, List dataTCs,
        boolean isFirst, boolean isLast)
    {
        File newFile = this.parent.getStreamFile();
        try
        {
            if (tmpFile == null)
                tmpFile = File.createTempFile("wowza", "flv");
        }
        catch (Exception e)
        {
            WMSLoggerFactory.getLogger(MediaWriterFLVBasic.class).error(
                "MediaWriterFLVMetadata: Error createTempFile: "+
                tmpFile+" :"+e.toString());
        }

        boolean localAppend = this.parent.isAppend();
        if (isFirst)
        {
            AMFDataArray keyFrames = null;

            long startTC = 0;
            if (newFile.exists())
            {
                if (localAppend)
                {
                    startTC = FLVUtils.getLastTC(newFile);
                    keyFrames = getKeyFrames(newFile);
                    copyPacketsToTmpFile(newFile, tmpFile);
                }
            }
        }
    }
}
```

```

        if (versionFile)
            FileUtils.versionFile(newFile);
        else
        {
            try
            {
                newFile.delete();
            }
            catch (Exception e)
            {
            }
        }
    }
    else
        localAppend = false;

    if (keyFrames == null)
        keyFrames = new AMFDataArray();
    extraMetadata.put("keyFrames", keyFrames);

    this.currentTCs[FLVUtils.FLV_TCINDEXAUDIO] = startTC;
    this.currentTCs[FLVUtils.FLV_TCINDEXVIDEO] = startTC;
    this.currentTCs[FLVUtils.FLV_TCINDEXDATA] = startTC;
}
else
    localAppend = true;

    AMFDataArray keyFrames = (AMFDataArray)extraMetadata.get("keyFrames");
    long timecode = this.currentTCs[FLVUtils.FLV_TCINDEXVIDEO];
    int size = videoPackets.size();
    for(int i=0;i<size;i++)
    {
        ByteBuffer data = (ByteBuffer)videoPackets.get(i);
        int firstByte = data.get(0);
        timecode += ((Long)videoTCs.get(i)).longValue();
        if (FLVUtils.getFrameType(firstByte) == FLVUtils.FLV_KFRAME)
        {
            double durationSecs = ((double)timecode) / 1000.0;
            AMFDataObj dataObj = new AMFDataObj();
            dataObj.put("name", new AMFDataItem("keyframe
"+durationSecs));

            dataObj.put("time", new AMFDataItem(durationSecs));
            keyFrames.add(dataObj);
        }
    }

    try
    {
        FileOutputStream ds = new FileOutputStream(tmpFile, localAppend);
        FLVUtils.writePackets(ds, audioPackets, videoPackets, dataPackets,

```



```

        audioTCs, videoTCs, dataTCs, currentTCs);

        ds.flush();
        ds.close();
    }
    catch (Exception e)
    {
        WMSLoggerFactory.getLogger(MediaWriterFLVBasic.class).error(
            "MediaWriterFLVMetadata: Error writing to tmp file:

"+
            newFile.getPath()+" :"+e.toString());
    }

    if (isLast)
    {
        duration = Math.max(Math.max(currentTCs[FLVUtils.FLV_TCINDEXAUDIO],
            currentTCs[FLVUtils.FLV_TCINDEXVIDEO]),
            currentTCs[FLVUtils.FLV_TCINDEXDATA]);
        double durationSecs = ((double)duration) / 1000.0;

        try
        {
            AMFPacket packet = null;
            FileOutputStream ds = new FileOutputStream(newFile);

            FileInputStream di = new FileInputStream(tmpFile);
            FLVUtils.writeHeader(ds, durationSecs, extraMetadata);
            while((packet = FLVUtils.readChunk(di)) != null)
            {
                FLVUtils.writeChunk(ds, packet.getDataBuffer(),
packet.getSize(),
                                packet.getTimecode(),
(byte)packet.getType());
            }
            di.close();

            ds.flush();
            ds.close();

            tmpFile.delete();
        }
        catch (Exception e)
        {
            WMSLoggerFactory.getLogger(MediaWriterFLVBasic.class).error(
                "MediaWriterFLVMetadata: Error tmp writing to

file: "+
                newFile.getPath()+" :"+e.toString());
        }
    }
}

private void copyPacketsToTmpFile(File newFile, File tmpFile)

```

```

{
    AMFDataArray keyFrames = null;
    try
    {
        AMFPacket packet = null;
        FileOutputStream ds = new FileOutputStream(tmpFile);

        FileInputStream di = new FileInputStream(newFile);
        FLVUtils.readHeader(di);
        FLVUtils.readChunk(di); // skip metaData packet
        while((packet = FLVUtils.readChunk(di)) != null)
        {
            FLVUtils.writeChunk(ds, packet.getDataBuffer(),
packet.getSize(),
                                packet.getTimecode(),
(byte)packet.getType());
        }
        di.close();

        ds.flush();
        ds.close();
    }
    catch (Exception e)
    {
        WMSLoggerFactory.getLogger(MediaWriterFLVBasic.class).error(
            "MediaWriterFLVMetadata: Error copyPacketsToTmpFile:
"+
            newFile.getPath()+" :"+e.toString());
    }
}

private AMFDataArray getKeyFrames(File newFile)
{
    AMFDataArray keyFrames = null;
    try
    {
        BufferedInputStream inStream = new BufferedInputStream(new
FileInputStream(newFile));
        FLVUtils.readHeader(inStream);
        AMFPacket packet = FLVUtils.readChunk(inStream);
        if (packet.getType() == IVHost.CONTENTTYPE_DATA0 || packet.getType()
== IVHost.CONTENTTYPE_DATA3)
        {
            byte[] mbytes = packet.getData();
            int moffset = 0;
            if (packet.getType() == IVHost.CONTENTTYPE_DATA3 &&
mbytes.length > 0)
            {
                if (mbytes[0] == 0)
                    moffset = 1;
            }
        }
    }
}

```

```

        AMFDataList dataList = new AMFDataList(mbytes, moffset,
mbytes.length-moffset);
        if (dataList.size() > 1)
        {
            if (dataList.get(1).getType() ==
AMFData.DATA_TYPE_MIXED_ARRAY)
            {
                AMFDataMixedArray metaValues =
                (AMFDataMixedArray)dataList.get(1);
                if (metaValues.containsKey("keyFrames"))
                    keyFrames =
                (AMFDataArray)metaValues.get("keyFrames");
            }
        }
        inStream.close();
    }
    catch (Exception e)
    {
        WMSLoggerFactory.getLogger(MediaWriterFLVBasic.class).error(
            "MediaWriterFLVMetadata: Error getKeyFrames: "+
            newFile.getPath()+" :"+e.toString());
    }

    return keyFrames;
}

public boolean isVersionFile()
{
    return versionFile;
}

public void setVersionFile(boolean versionFile)
{
    this.versionFile = versionFile;
}

public void putMetaData(String name, AMFData value)
{
    this.extraMetadata.put(name, value);
}
}

```

To use this class, edit [install-dir]/conf/MediaWriter and replace the definition for the **flv** MediaWriter:

```

<MediaWriter>
  <Name>flv</Name>
  <Description>FLV Media Writer</Description>
  <FileExtension>flv</FileExtension>
  <ClassBase>com.wowza.wms.plugin.mediawriter.flv.MediaWriterFLVMetadata</ClassBase>
</MediaWriter>

```

## Method Summary

long	<a href="#"><code>getDuration()</code></a> Get the recorded duration of the file in seconds
boolean	<a href="#"><code>isVersionFile()</code></a> Return true if the old file is to be versioned
boolean	<a href="#"><code>isWaitForVideoKeyFrame()</code></a> get wait for key frame
void	<a href="#"><code>putMetaData(String name, AMFData value)</code></a> Add metadata to the metadata packet.
void	<a href="#"><code>setMediaWriterItem(MediaWriterItem mediaWriterItem)</code></a> Set the media write definition
void	<a href="#"><code>setParent(IMediaStream parent)</code></a> Set the parent stream for this media write object
void	<a href="#"><code>setVersionFile(boolean versionFile)</code></a> Set to true if the old file is to be versioned
void	<a href="#"><code>setWaitForVideoKeyFrame(boolean waitForVideoKeyFrame)</code></a> Set to true if you want the recorder to skip opening frames until it hits a key frame
void	<a href="#"><code>writePackets(java.util.List audioPackets, java.util.List videoPackets, java.util.List dataPackets, java.util.List audioTCs, java.util.List videoTCs, java.util.List dataTCs, java.util.List dataTypes, boolean isFirst, boolean isLast)</code></a> Invoked each time a set of packets are ready to be presisted.

## Methods

### writePackets

```

public void writePackets(java.util.List audioPackets,
    java.util.List videoPackets,
    java.util.List dataPackets,
    java.util.List audioTCs,
    java.util.List videoTCs,
    java.util.List dataTCs,
    java.util.List dataTypes,
    boolean isFirst,
    boolean isLast)

```

Invoked each time a set of packets are ready to be presisted.

(continued from last page)

**Parameters:**

audioPackets - List of audio packets

videoPackets - List of video packets

dataPackets - List of data packets

audioTCs - List of audio timecodes

videoTCs - List of video timecodes

dataTCs - List of data timecodes

dataTypes - list of integer packets types (IVHost.CONTENTTYPE\_DATA0, IVHost.CONTENTTYPE\_DATA3) - if null assumed to be IVHost.CONTENTTYPE\_DATA0

isFirst - true if first packet to be written

isLast - false if last packet to be written

---

**setMediaWriterItem**

```
public void setMediaWriterItem(MediaWriterItem mediaWriterItem)
```

Set the media write definition

**Parameters:**

mediaWriterItem - media write definition

---

**setParent**

```
public void setParent(IMediaStream parent)
```

Set the parent stream for this media write object

**Parameters:**

parent

---

**isVersionFile**

```
public boolean isVersionFile()
```

Return true if the old file is to be versioned

**Returns:**

true if the old file is to be versioned

---

**setVersionFile**

```
public void setVersionFile(boolean versionFile)
```

Set to true if the old file is to be versioned

**Parameters:**

versionFile

---

**isWaitForVideoKeyFrame**

```
public boolean isWaitForVideoKeyFrame()
```

get wait for key frame

**Returns:**

wait for key frame

(continued from last page)

## setWaitForVideoKeyFrame

```
public void setWaitForVideoKeyFrame(boolean waitForVideoKeyFrame)
```

Set to true if you want the recorder to skip opening frames until it hits a key frame

### Parameters:

waitForVideoKeyFrame - wait for key frame

---

## putMetaData

```
public void putMetaData(String name,  
    AMFData value)
```

Add metadata to the metadata packet. Only metadata added before the first call to writePackets will be included in the file

### Parameters:

name - field name

value - metadata value

---

## getDuration

```
public long getDuration()
```

Get the recorded duration of the file in seconds

### Returns:

recorded duration of the file in seconds

---

## com.wowza.wms.stream Interface IMediaWriterActionNotify

---

public interface **IMediaWriterActionNotify**  
extends

---

### Method Summary

void	<a href="#">onFLVAddMetadata</a> ( <a href="#">IMediaStream</a> stream, java.util.Map extraMetadata)
void	<a href="#">onWriteComplete</a> ( <a href="#">IMediaStream</a> stream, java.io.File file)

---

### Methods

#### onWriteComplete

```
public void onWriteComplete(IMediaStream stream,  
                             java.io.File file)
```

---

#### onFLVAddMetadata

```
public void onFLVAddMetadata(IMediaStream stream,  
                             java.util.Map extraMetadata)
```

## 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="#">liveStreamPacketizers</a>
public static final	<a href="#">MAXSTREAMINDEX</a> Value: <b>65536</b>
protected	<a href="#">mediaStreamListeners</a>
protected	<a href="#">nameGroupId</a>
protected	<a href="#">nameGroups</a>
protected	<a href="#">nextStreamId</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</a> ( <a href="#">IApplicationInstance</a> appInstance) Create empty MediaStreamMap collection
--------	--

### Method Summary

void	<a href="#">addMediaStreamListener</a> ( <a href="#">IMediaStreamNotify</a> mediaStreamListener) Add a media stream listener.
------	--



MediaStreamMapGroup	<a href="#">addNameGroup</a> (MediaStreamMapGroup newGroup)
void	<a href="#">broadcastPlayMessage</a> (IMediaStream stream, long timecode, java.nio.ByteBuffer msg, int objectEncoding) Send a broadcast message to all play stream that are listening to this live published stream.
int	<a href="#">broadcastGetObjectEncoding</a> (IMediaStream stream) Get the minimum object encoding level for the clients playing this stream.
void	<a href="#">clearStreamName</a> (String name) Unregister a published live media stream name.
void	<a href="#">clearStreamName</a> (String name, IMediaStream stream) Unregister a published live media stream name.
<a href="#">IApplicationInstance</a>	<a href="#">getAppInstance</a> () Get the parent applicationInstance.
String	<a href="#">getAppInstanceName</a> () Get the name of the parent applicationInstance.
String	<a href="#">getAppName</a> () Get the name of the parent application.
int	<a href="#">getCount</a> () Get the total number of streams stored in the mediaStreamMap
ILiveStreamPacketizer	<a href="#">getLiveStreamPacketizer</a> (String streamName, String packetizerName, boolean doCreate) Get a live stream packetizer by name and packetizer id
Object	<a href="#">getLiveStreamPacketizerLock</a> () Get the lock to the live stream packetizer system
java.util.Set	<a href="#">getNameGroups</a> ()
java.util.Set	<a href="#">getNameGroups</a> (String streamName)
java.util.Set	<a href="#">getNameGroupStreamNames</a> (String streamName)
int	<a href="#">getNextStreamIndex</a> () Reserve a clientless stream id for a new media stream.
int	<a href="#">getNextStreamIndex</a> (IClient client) Reserve a stream for a client connection.
int	<a href="#">getNextStreamIndex</a> (com.wowza.wms.netconnection.INetConnection netConnection) Reserve a stream for a netConnection connection.
java.util.List	<a href="#">getPublishStreamNames</a> () Returns a List of published stream names
<a href="#">IMediaStream</a>	<a href="#">getStream</a> (IClient client, int index) Get a media stream reference by stream id.
<a href="#">IMediaStream</a>	<a href="#">getStream</a> (IClient client, int index, boolean doCreate) Get a media stream reference by stream id.

<a href="#">IMediaStream</a>	<a href="#">getStream</a> (com.wowza.wms.netconnection.INetConnection netConnection, int index) Get a media stream object that is owned by a server to server netConnection object (not yet implemented).
<a href="#">IMediaStream</a>	<a href="#">getStream</a> (com.wowza.wms.netconnection.INetConnection netConnection, int index, boolean doCreate) Get a media stream reference by stream id.
<a href="#">IMediaStream</a>	<a href="#">getStream</a> (String name) Get a media stream by stream name.
<a href="#">IMediaStream</a>	<a href="#">getStreamClientless</a> (int index, String streamTypeStr) Get a media stream reference by stream id.
edu.emory.mathcs.backport.java.util.concurrent.locks.WMSReadWriteLock	<a href="#">getStreamListLock</a> () Get the underlying read/write lock associated with the list of streams
edu.emory.mathcs.backport.java.util.concurrent.locks.WMSReadWriteLock	<a href="#">getStreamNameLock</a> () Get the underlying read/write lock associated with the stream names list
java.util.List	<a href="#">getStreams</a> () Returns a list of IMediaStream objects
<a href="#">IVHost</a>	<a href="#">getVHost</a> () Get the parent vHost.
void	<a href="#">notifyMediaStreamCreate</a> ( <a href="#">IMediaStream</a> mediaStream) Notify all media stream listeners that a new media stream object has been created.
void	<a href="#">notifyMediaStreamDestroy</a> ( <a href="#">IMediaStream</a> mediaStream) Notify all media stream listeners that a media stream object is being destroyed.
void	<a href="#">notifyPlayPublish</a> ( <a href="#">IMediaStream</a> stream) Notify all play streams that are listening to this stream that the stream is going into a state of publish (NetStream.Play.PublishNotify).
void	<a href="#">notifyPlayUnpublish</a> ( <a href="#">IMediaStream</a> stream) Notify all play streams that are listening to this stream that the stream is going into a state of unpublished (NetStream.Play.UnpublishNotify).
void	<a href="#">notifyPlayUnpublish</a> (String streamName) Notify all play streams that are listening to this stream name that the stream is going into a state of unpublished (NetStream.Play.UnpublishNotify).
void	<a href="#">removeLiveStreamPacketizer</a> (String streamName) Remove all live stream packetizers for this stream name
ILiveStreamPacketizer	<a href="#">removeLiveStreamPacketizer</a> (String streamName, String packetizerName) Remove live stream packetizer
void	<a href="#">removeMediaStreamListener</a> ( <a href="#">IMediaStreamNotify</a> mediaStreamListener) Remove a media stream listener.
MediaStreamMapGroup	<a href="#">removeNameGroup</a> (int groupId)
MediaStreamMapGroup	<a href="#">removeNameGroup</a> (MediaStreamMapGroup nameGroup)

void	<a href="#">removeStream</a> ( <a href="#">IClient</a> client, int index) Remove a stream associated with a client connection
void	<a href="#">removeStream</a> (com.wowza.wms.netconnection.INetConnection netConnection, int index) Remove a stream associated with a netConnection object
void	<a href="#">removeStream</a> (int index) Remove a clientless media stream
void	<a href="#">setStreamName</a> ( <a href="#">IMediaStream</a> stream, String name) Insert live media stream into the mediaStreamMap by name.
long	<a href="#">streamToIndex</a> ( <a href="#">IMediaStream</a> stream) Get the unique stream identifier for a given stream

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Fields

### MAXSTREAMINDEX

public static final int **MAXSTREAMINDEX**

Constant value: **65536**

### streams

protected java.util.Map **streams**

### streamNames

protected java.util.Map **streamNames**

### streamNameToGroup

protected java.util.Map **streamNameToGroup**

### nameGroups

protected java.util.List **nameGroups**

(continued from last page)

## nameGroupId

protected int **nameGroupId**

## 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**

## Constructors

### MediaStreamMap

public **MediaStreamMap**([IApplicationInstance](#) appInstance)

Create empty MediaStreamMap collection

#### Parameters:

appInstance - parent applicationInstance

## Methods

(continued from last page)

---

## 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

---

## 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)
```

(continued from last page)

Get a media stream reference by stream id. If it does not exist, create a new one with the given id. This is a clientless stream that is served by the backend of the server. If creating a new media stream object, its id must first be obtained by calling [getNextStreamIndex\(\)](#).

**Parameters:**

index - stream id  
streamTypeStr - stream type

**Returns:**

media stream object

---

## getStream

```
public IMediaStream getStream(IClient client,  
    int index,  
    boolean doCreate)
```

Get a media stream reference by stream id. If it does not exist, create a new one with the given id. If creating a new media stream object, its id must first be obtained by calling [getNextStreamIndex\(IClient\)](#).

**Parameters:**

client - client  
index - stream id  
doCreate - true to create if it does not exist

**Returns:**

media stream object

---

## getStream

```
public IMediaStream getStream(com.wowza.wms.netconnection.INetConnection  
    netConnection,  
    int index)
```

Get a media stream object that is owned by a server to server netConnection object (not yet implemented).

**Parameters:**

netConnection - netConnection to remote server  
index - stream id

**Returns:**

media stream object or null if does not exist

---

## getStream

```
public IMediaStream getStream(com.wowza.wms.netconnection.INetConnection  
    netConnection,  
    int index,  
    boolean doCreate)
```

Get a media stream reference by stream id. If it does not exist, create a new one with the given id. If creating a new media stream object, its id must first be obtained by calling [getNextStreamIndex\(INetConnection\)](#).

**Parameters:**

netConnection - netConnection to remote server  
index - stream id  
doCreate - true to create if it does not exist

**Returns:**

media stream object

## getStream

```
public IMediaStream getStream(String name)
```

Get a media stream by stream name. Only published live streams are stored in the mediaStreamMap by name. This method is used to lookup a published live stream by name.

**Parameters:**

name - stream name

**Returns:**

media stream object or null if does not exist

---

## setStreamName

```
public void setStreamName(IMediaStream stream,  
    String name)
```

Insert live media stream into the mediaStreamMap by name.

**Parameters:**

stream - media stream object

name - media stream name

---

## clearStreamName

```
public void clearStreamName(String name)
```

Unregister a published live media stream name.

**Parameters:**

name - stream name

---

## streamToIndex

```
public long streamToIndex(IMediaStream stream)
```

Get the unique stream identifier for a given stream

**Parameters:**

stream - stream

**Returns:**

unique stream identifier

---

## clearStreamName

```
public void clearStreamName(String name,  
    IMediaStream stream)
```

Unregister a published live media stream name.

**Parameters:**

name - stream name

stream - stream

---

(continued from last page)

## getNextStreamIndex

```
public int getNextStreamIndex(com.wowza.wms.netconnection.INetConnection  
netConnection)
```

Reserve a stream for a netConnection connection. Use [getStream\(INetConnection, int, boolean\)](#) to create stream.

**Parameters:**

netConnection

**Returns:**

next stream index

---

## getNextStreamIndex

```
public int getNextStreamIndex(IClient client)
```

Reserve a stream for a client connection. Use [getStream\(IClient, int, boolean\)](#) to create stream.

**Parameters:**

client - parent client

**Returns:**

stream index

---

## getNextStreamIndex

```
public int getNextStreamIndex( )
```

Reserve a clientless stream id for a new media stream. Use this method to obtain a stream id for a new media stream object that is then created with a call to `getClientlessStream()`.

**Returns:**

new reserved stream id

---

## getCount

```
public int getCount( )
```

Get the total number of streams stored in the mediaStreamMap

**Returns:**

total number of streams stored in the mediaStreamMap

---

## removeStream

```
public void removeStream(com.wowza.wms.netconnection.INetConnection netConnection,  
int index)
```

Remove a stream associated with a netConnection object

**Parameters:**

netConnection - net connection

index - stream index

---

## removeStream

```
public void removeStream(IClient client,  
int index)
```



(continued from last page)

Remove a stream associated with a client connection

**Parameters:**

client - client  
index - stream index

---

## removeStream

```
public void removeStream(int index)
```

Remove a clientless media stream

**Parameters:**

index - stream index

---

## getAppName

```
public String getAppName()
```

Get the name of the parent application.

**Returns:**

parent application name.

---

## getAppInstanceName

```
public String getAppInstanceName()
```

Get the name of the parent applicationInstance.

**Returns:**

parent applicationInstance name

---

## 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).

(continued from last page)

**Parameters:**

stream - live published stream that is being unpublished

---

**notifyPlayUnpublish**

```
public void notifyPlayUnpublish(String streamName)
```

Notify all play streams that are listening to this stream name that the stream is going into a state of unpublished (NetStream.Play.UnpublishNotify).

**Parameters:**

streamName - live published stream that is being unpublished

---

**broadcastGetObjectEncoding**

```
public int broadcastGetObjectEncoding(IMediaStream stream)
```

Get the minimum object encoding level for the clients playing this stream.

**Parameters:**

stream - publish stream

**Returns:**

object encoding level (AMF0 or AMF3)

---

**broadcastPlayMessage**

```
public void broadcastPlayMessage(IMediaStream stream,  
    long timecode,  
    java.nio.ByteBuffer msg,  
    int objectEncoding)
```

Send a broadcast message to all play stream that are listening to this live published stream.

**Parameters:**

stream - media stream  
timecode - timecode (milliseconds)  
msg - byte[] of data that will be deserialized as an AMFData object.  
objectEncoding - object encoding (AMF0 or AMF3)

---

**notifyPlayPublish**

```
public void notifyPlayPublish(IMediaStream stream)
```

Notify all play streams that are listening to this stream that the stream is going into a state of publish (NetStream.Play.PublishNotify).

**Parameters:**

stream - live published stream that is being published

---

**addMediaStreamListener**

```
public void addMediaStreamListener(IMediaStreamNotify mediaStreamListener)
```

Add a media stream listener. A media stream listener will receive the following events: onMediaStreamCreate, onMediaStreamDestroy.

**Parameters:**

mediaStreamListener - media stream listener

## removeMediaStreamListener

```
public void removeMediaStreamListener(IMediaStreamNotify mediaStreamListener)
```

Remove a media stream listener.

**Parameters:**

mediaStreamListener - media stream listener

---

## notifyMediaStreamCreate

```
public void notifyMediaStreamCreate(IMediaStream mediaStream)
```

Notify all media stream listeners that a new media stream object has been created.

**Parameters:**

mediaStream - new media stream object

---

## notifyMediaStreamDestroy

```
public void notifyMediaStreamDestroy(IMediaStream mediaStream)
```

Notify all media stream listeners that a media stream object is being destroyed.

**Parameters:**

mediaStream - media stream object being destroyed

---

## removeLiveStreamPacketizer

```
public void removeLiveStreamPacketizer(String streamName)
```

Remove all live stream packetizers for this stream name

**Parameters:**

streamName - streamName

---

## removeLiveStreamPacketizer

```
public ILiveStreamPacketizer removeLiveStreamPacketizer(String streamName,  
String packetizerName)
```

Remove live stream packetizer

**Parameters:**

streamName - stream name

packetizerName - packetizer id

**Returns:**

live stream packetizer

---

## getLiveStreamPacketizerLock

```
public Object getLiveStreamPacketizerLock()
```

Get the lock to the live stream packetizer system

**Returns:**

lock to the live stream packetizer system

---

---

## getLiveStreamPacketizer

```
public ILiveStreamPacketizer getLiveStreamPacketizer(String streamName,  
    String packetizerName,  
    boolean doCreate)
```

Get a live stream packetizer by name and packetizer id

### Parameters:

streamName - stream name  
packetizerName - packetizer id  
doCreate - create if does not exist

### Returns:

live stream packetizer

---

## addNameGroup

```
public MediaStreamMapGroup addNameGroup(MediaStreamMapGroup newGroup)
```

---

## removeNameGroup

```
public MediaStreamMapGroup removeNameGroup(int groupId)
```

---

## removeNameGroup

```
public MediaStreamMapGroup removeNameGroup(MediaStreamMapGroup nameGroup)
```

---

## getNameGroups

```
public java.util.Set getNameGroups()
```

---

## getNameGroups

```
public java.util.Set getNameGroups(String streamName)
```

---

## getNameGroupStreamNames

```
public java.util.Set getNameGroupStreamNames(String streamName)
```

---

---

Package

**com.wowza.wms.stream.publish**

## com.wowza.wms.stream.publish Interface IPublishingProvider

All Known Implementing Classes:

[PublishingProviderMediaReader](#), [PublishingProviderLive](#)

public interface **IPublishingProvider**  
extends

### Method Summary

void	<a href="#">close()</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)

### Methods

#### play

public boolean **play**([Publisher](#) publisher)

#### close

public void **close**()

#### seek

public boolean **seek**(long timecode,  
int seekType)

#### seek

public boolean **seek**(long timecode)

---

## **setDuration**

```
public void setDuration(long duration)
```

---

## **setRealTimeStartTime**

```
public void setRealTimeStartTime(long realTimeStartTime)
```

---

## com.wowza.wms.stream.publish Interface IStreamActionNotify

---

public interface **IStreamActionNotify**  
extends

---

### Method Summary

void	<a href="#">onPlaylistItemStart</a> ( <a href="#">Stream</a> stream, <a href="#">PlaylistItem</a> playlistItem)
void	<a href="#">onPlaylistItemStop</a> ( <a href="#">Stream</a> stream, <a href="#">PlaylistItem</a> playlistItem)

---

### Methods

#### onPlaylistItemStart

```
public void onPlaylistItemStart(Stream stream,  
    PlaylistItem playlistItem)
```

---

#### onPlaylistItemStop

```
public void onPlaylistItemStop(Stream stream,  
    PlaylistItem playlistItem)
```



## com.wowza.wms.stream.publish Class Playlist

java.lang.Object

└─com.wowza.wms.stream.publish.Playlist

public class **Playlist**  
extends Object

### Constructor Summary

public	<a href="#">Playlist</a> (String sName) Class constructor - A simple structure that maintains a list of playlist items.
--------	--

### Method Summary

void	<a href="#">addItem</a> (String sName, int start, int length) Appends an item to this playlist
String	<a href="#">getName</a> () Returns the name of this playlist as defined in the XML definition file
boolean	<a href="#">getRepeat</a> ()
boolean	<a href="#">open</a> ( <a href="#">Stream</a> s) Opens this playlist on the given stream...
void	<a href="#">setRepeat</a> (boolean repeat)

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Constructors

#### Playlist

public **Playlist**(String sName)

Class constructor - A simple structure that maintains a list of playlist items.

#### Parameters:

sName -- name of playlist - as defined in the XML definition file

### Methods

(continued from last page)

---

## 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

---

## open

```
public boolean open(Stream s)
```

Opens this playlist on the given stream... stopping anything currently playing on that stream and switching over.

**Parameters:**

s -- stream to play on

**Returns:**

- true if successful

---

## getName

```
public String getName()
```

Returns the name of this playlist as defined in the XML definition file

**Returns:**

name of this playlist

## com.wowza.wms.stream.publish Class PlaylistItem

java.lang.Object

└─com.wowza.wms.stream.publish.PlaylistItem

public class **PlaylistItem**  
extends Object

### Constructor Summary

public	<a href="#">PlaylistItem</a> (String name, int start, int length, int index) Class constructor - A simple structure to define parameters associated with a playlist item
--------	---

### Method Summary

int	<a href="#">getIndex()</a>
int	<a href="#">getLength()</a> Number of seconds of track to play
String	<a href="#">getName()</a> Name of playlist stream
int	<a href="#">getStart()</a> Number of seconds into track to start from
void	<a href="#">setIndex</a> (int index)
String	<a href="#">toString()</a>

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructors

### PlaylistItem

```
public PlaylistItem(String name,
                    int start,
                    int length,
                    int index)
```

Class constructor - A simple structure to define parameters associated with a playlist item

#### Parameters:

name - - name of the playlist stream

start - - number of seconds into the track to start from (-2 means live stream)

(continued from last page)

length -- number of seconds of track to play

index -- index in the playlist

## Methods

### getName

```
public String getName()
```

Name of playlist stream

**Returns:**- name of stream

---

### getStart

```
public int getStart()
```

Number of seconds into track to start from

**Returns:**- number of seconds

---

### getLength

```
public int getLength()
```

Number of seconds of track to play

**Returns:**- number of seconds

---

### getIndex

```
public int getIndex()
```

---

### setIndex

```
public void setIndex(int index)
```

---

### toString

```
public String toString()
```

## com.wowza.wms.stream.publish Class Publisher

java.lang.Object

└─com.wowza.wms.stream.publish.Publisher

---

```
public class Publisher  
extends Object
```

Publisher: clientless stream publisher. This class can be used to publish raw video, audio and metadata packets to the Wowza Pro server. Here is a quick snippet of code that illustrates how to use it.

This code below will publish data the stream named "myStream". It will be streamed to the default virtual host and available at the rtmp address rtmp://[server-ip-address]/streamtest.

```
IVHost vhost = VHostSingleton.getInstance(VHost.VHOST_DEFAULT);
Publisher publisher = Publisher.createInstance(vhost, "streamtest");

publisher.setFileExtension("flv");
publisher.setStreamType("live");

publisher.publish("myStream", "live");

// sit in a loop adding data
boolean done = false;
while(true)
{
    AMFPacket amfPacket;

    // read packet from audio, video, data source
    // amfPacket = readPacketFromSomewhere();

    switch (amfPacket.getType())
    {
        case IVHost.CONTENTTYPE_AUDIO:
            publisher.addAudioData(amfPacket.getData(), amfPacket.getSize(),
amfPacket.getTimecode());
            break;
        case IVHost.CONTENTTYPE_VIDEO:
            publisher.addVideoData(amfPacket.getData(), amfPacket.getSize(),
amfPacket.getTimecode());
            break;
        case IVHost.CONTENTTYPE_DATA:
            publisher.addDataData(amfPacket.getData(), amfPacket.getSize(),
amfPacket.getTimecode());
            break;
    }
    if (done)
        break;
}

publisher.unpublish();
publisher.close();
```

Basic packet format:

Audio:

AAC

[1-byte header]

[1-byte codec config indicator (0 - audio data, 1 - codec config packet)]

[n-bytes audio content or codec config data]

All others

[1-byte header]

[n-bytes audio content]

Below is the bit

layout of the header byte of data (table goes from least significant bit to most significant bit):

1 bit Number of channels:

- 0 mono
- 1 stereo

1 bit Sample size:

- 0 8 bits per sample
- 1 16 bits per sample

2 bits Sample rate:

- 0 special or 8KHz
- 1 11KHz
- 2 22KHz
- 3 44KHz

4 bits Audio type:

- 0 PCM (big endian)
- 1 PCM (swf - ADPCM)
- 2 MP3
- 3 PCM (little endian)
- 4 Nelly Moser ASAO 16KHz Mono
- 5 Nelly Moser ASAO 8KHz Mono
- 6 Nelly Moser ASAO
- 7 G.711 ALaw
- 8 G.711 MULaw
- 9 Reserved
- a AAC
- b Speex
- f MP3 8Khz

Note: For AAC the codec config data is generally a two byte packet that describes the stream. It must

be published first. Here is the basic code to fill in the codec config data.

```
AACFrame frame = new AACFrame();
```

```

int sampleRate = 22100;
int channels = 2;
frame.setSampleRate(sampleRate);
frame.setRateIndex(AACUtils.sampleRateToIndex(sampleRate));
frame.setChannels(channels);
frame.setChannelIndex(AACUtils.channelCountToIndex(sampleRate));
byte[] codecConfig = new byte[2];
AACUtils.encodeAACCodecConfig(frame, codecConfig, 0);

```

Note: For AAC the header byte is always 0xaf

Note: For Speex the audio data must be encoded as 16000Hz wide band

Video:

H.264

```

[1-byte header]
[1-byte codec config indicator (1 - video data, 0 - codec config packet)]
[3-byte time difference between dts and pts in milliseconds]
[n-bytes video content or codec config data]

```

All others

```

[1-byte header]
[n-bytes audio content]

```

Below is the bit layout of the header byte of data (table goes from least significant bit to most significant bit):

4 bits Video type:

2	Sorenson Spark (H.263)
3	Screen
4	On2 VP6
5	On2 VP6A
6	Screen2
7	H.264

2 bit Frame type:

1	K frame (key frame)
2	P frame
3	B frame

Note: H.264 codec config data is the same as the AVCc packet in a QuickTime container.

Note: All timecode data is in milliseconds

## Method Summary



void	<a href="#">addAudioData</a> (byte[] data, int offset, int len, long timecode) Add audio data
void	<a href="#">addAudioData</a> (byte[] data, int len, long timecode) Add audio data
void	<a href="#">addAudioData</a> (byte[] data, long timecode) Add audio data
void	<a href="#">addDataData</a> (byte[] data, int offset, int len, long timecode) Add metadata
void	<a href="#">addDataData</a> (byte[] data, int len, long timecode) Add metadata
void	<a href="#">addDataData</a> (byte[] data, long timecode) Add metadata
void	<a href="#">addVideoData</a> (byte[] data, int offset, int len, long timecode) Add video data
void	<a href="#">addVideoData</a> (byte[] data, int len, long timecode) Add video data
void	<a href="#">addVideoData</a> (byte[] data, long timecode) Add video data
void	<a href="#">close</a> () Close the publisher
static <a href="#">Publisher</a>	<a href="#">createInstance</a> ( <a href="#">IApplicationInstance</a> appInstance)
static <a href="#">Publisher</a>	<a href="#">createInstance</a> ( <a href="#">IVHost</a> vhost, String applicationName)
static <a href="#">Publisher</a>	<a href="#">createInstance</a> ( <a href="#">IVHost</a> vhost, String applicationName, String appInstanceName)
void	<a href="#">flush</a> () Flush the packets from the input buffer to the output buffer
<a href="#">IApplicationInstance</a>	<a href="#">getAppInstance</a> ()
String	<a href="#">getFileExtension</a> () Get the file extension (default flv)
long	<a href="#">getMaxTimecode</a> () Highest timecode written through this publisher (milliseconds).
<a href="#">IMediaStream</a>	<a href="#">getStream</a> () Get the media stream object
String	<a href="#">getStreamType</a> ()
void	<a href="#">publish</a> (String streamName) Publish a stream (null to stop publishing)

void	<a href="#">publish</a> (String streamName, String howToPublish) Start publishing a stream (streamName = null to stop).
void	<a href="#">setFileExtension</a> (String fileExtension) Set the file extension
void	<a href="#">setStream</a> ( <a href="#">IMediaStream</a> stream) Set the media stream object
void	<a href="#">setStreamType</a> (String streamType) Set the stream type (default live)
void	<a href="#">unpublish</a> ()

#### Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## Methods

### createInstance

```
public static Publisher createInstance(IVHost vhost,
    String applicationName)
```

### createInstance

```
public static Publisher createInstance(IVHost vhost,
    String applicationName,
    String appInstanceName)
```

### createInstance

```
public static Publisher createInstance(IApplicationInstance appInstance)
```

### getStream

```
public IMediaStream getStream()
```

Get the media stream object

#### Returns:

media stream object

### setStream

```
public void setStream(IMediaStream stream)
```

Set the media stream object

(continued from last page)

**Parameters:**

stream - media stream object

---

**getStreamType**

```
public String getStreamType()
```

---

**setStreamType**

```
public void setStreamType(String streamType)
```

Set the stream type (default live)

**Parameters:**

streamType - stream type

---

**publish**

```
public void publish(String streamName)
```

Publish a stream (null to stop publishing)

**Parameters:**

streamName - stream name

---

**getFileExtension**

```
public String getFileExtension()
```

Get the file extension (default flv)

**Returns:**

file extension

---

**setFileExtension**

```
public void setFileExtension(String fileExtension)
```

Set the file extension

**Parameters:**

fileExtension - file extension

---

**getAppInstance**

```
public IApplicationInstance getAppInstance()
```

---

**unpublish**

```
public void unpublish()
```

## publish

```
public void publish(String streamName,  
    String howToPublish)
```

Start publishing a stream (streamName = null to stop). Valid howToPublish values are (live, record, append)

### Parameters:

streamName - stream name

howToPublish - publish method (live, record, append)

---

## addVideoData

```
public void addVideoData(byte[] data,  
    long timecode)
```

Add video data

### Parameters:

data - data

timecode - absolute timecode (milliseconds)

---

## addVideoData

```
public void addVideoData(byte[] data,  
    int len,  
    long timecode)
```

Add video data

### Parameters:

data - data

len - data length

timecode - absolute timecode (milliseconds)

---

## 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)

---

## addAudioData

```
public void addAudioData(byte[] data,  
    long timecode)
```

Add audio data

### Parameters:

data - data

---

(continued from last page)

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)

---

## addAudioData

```
public void addAudioData(byte[] data,  
    int offset,  
    int len,  
    long timecode)
```

Add audio data

### Parameters:

data - data  
len - data length  
offset - offset  
timecode - absolute timecode (milliseconds)

---

## addDataData

```
public void addDataData(byte[] data,  
    long timecode)
```

Add metadata

### Parameters:

data - data  
timecode - absolute timecode (milliseconds)

---

## addDataData

```
public void addDataData(byte[] data,  
    int len,  
    long timecode)
```

Add metadata

### Parameters:

data - data  
len - data length  
timecode - absolute timecode (milliseconds)

---

## addDataData

```
public void addDataData(byte[] data,  
    int offset,  
    int len,  
    long timecode)
```

(continued from last page)

Add metadata

**Parameters:**

data - data  
offset - offset  
len - data length  
timecode - absolute timecode (milliseconds)

---

**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)

## com.wowza.wms.stream.publish Class PublishingProviderLive

java.lang.Object

└─com.wowza.wms.stream.publish.PublishingProviderLive

All Implemented Interfaces:

[IPublishingProvider](#)

public class **PublishingProviderLive**  
extends Object  
implements [IPublishingProvider](#)

### 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> ()
boolean	<a href="#">play</a> ( <a href="#">Publisher</a> publisher)
boolean	<a href="#">seek</a> (long timecode)
boolean	<a href="#">seek</a> (long timecode, int seekType)
void	<a href="#">setDuration</a> (long duration)
void	<a href="#">setRealTimeStartTime</a> (long realTimeStartTime)

#### Methods inherited from class 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](#), [play](#), [seek](#), [seek](#), [setDuration](#), [setRealTimeStartTime](#)

### Constructors

(continued from last page)

## 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)
```



com.wowza.wms.stream.publish  
Class PublishingProviderMediaReader

java.lang.Object  
└─com.wowza.wms.stream.publish.PublishingProviderMediaReader

All Implemented Interfaces:  
[IPublishingProvider](#)

public class **PublishingProviderMediaReader**  
extends Object  
implements [IPublishingProvider](#)

Constructor Summary	
public	<a href="#">PublishingProviderMediaReader</a> ( <a href="#">Publisher</a> publisher, long timeoffset, String streamName)

Method Summary	
void	<a href="#">close</a> ()
long	<a href="#">getDuration</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)

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface <a href="#">com.wowza.wms.stream.publish.IPublishingProvider</a>
<a href="#">close</a> , <a href="#">play</a> , <a href="#">seek</a> , <a href="#">seek</a> , <a href="#">setDuration</a> , <a href="#">setRealTimeStartTime</a>

Constructors

(continued from last page)

## PublishingProviderMediaReader

```
public PublishingProviderMediaReader(Publisher publisher,  
                                     long timeoffset,  
                                     String streamName)
```

## Methods

### seek

```
public boolean seek(long timecode)
```

### seek

```
public boolean seek(long timecode,  
                    int seekType)
```

### play

```
public boolean play(Publisher publisher)
```

### close

```
public void close()
```

### getDuration

```
public long getDuration()
```

### setDuration

```
public void setDuration(long duration)
```

### setRealTimeStartTime

```
public void setRealTimeStartTime(long realTimeStartTime)
```

## com.wowza.wms.stream.publish Class Stream

java.lang.Object

└─com.wowza.wms.stream.publish.Stream

All Implemented Interfaces:

Runnable

public class **Stream**  
extends Object  
implements Runnable

### Constructor Summary

public	<a href="#">Stream()</a>
--------	--------------------------

### Method Summary

void	<a href="#">addListener(IStreamActionNotify listener)</a> Add a listener
boolean	<a href="#">addToPlaylist(int index, String name, int start, int length)</a> Inserts a media source item to this playlist at a particular index, without interruption.
boolean	<a href="#">addToPlaylist(String existing, String name, int start, int length)</a> Insert item into playlist just after the first item in the playlist with a given name.
void	<a href="#">close()</a> Call this method when you have finished with the playlist object.
static <a href="#">Stream</a>	<a href="#">createInstance(IApplicationInstance appInstance, String sName)</a> Use this to create a named Stream on an application instance.
static <a href="#">Stream</a>	<a href="#">createInstance(IVHost vhost, String applicationName, String sName)</a> Use this to create a named Stream on the default instance (_definst_) of an application on a particular VHost.
static <a href="#">Stream</a>	<a href="#">createInstance(IVHost vhost, String applicationName, String appInstanceName, String sName)</a> Use this to create a named Stream on the non-default instance of an application on a particular VHost.
<a href="#">PlaylistItem</a>	<a href="#">getCurrentItem()</a> Get the currently playing playlist item
Object	<a href="#">getLock()</a> Get the synchronization lock for this interface.
String	<a href="#">getName()</a> Returns the name of the playlist stream - the client would play this stream by this name.

java.util.List	<a href="#">getPlaylist()</a> Get the current playlist
int	<a href="#">getPollingInterval()</a> Get the polling interval (milliseconds)
<a href="#">Publisher</a>	<a href="#">getPublisher()</a>
boolean	<a href="#">getRepeat()</a> Use this to determine if the playlist is auto-repeating
boolean	<a href="#">isSwitchLog()</a> Log when a playlist switch occurs
void	<a href="#">next()</a>
void	<a href="#">next(int n)</a>
void	<a href="#">play(int n)</a>
boolean	<a href="#">play(String sPlaylist)</a> Add a media item to the playlist as defined by an XML file ..
boolean	<a href="#">play(String name, int start, int length, boolean reset)</a> Adds a media source item to this playlist -
void	<a href="#">previous()</a>
void	<a href="#">previous(int n)</a>
boolean	<a href="#">removeFromPlaylist(String name)</a> Remove all items matching the given stream name from the playlist.
void	<a href="#">removeListener(IStreamActionNotify listener)</a> Remove a listener
void	<a href="#">run()</a> Overridden from class Runnable ....
void	<a href="#">setPollingInterval(int pollingInterval)</a> Set the polling interval (milliseconds)
void	<a href="#">setRepeat(boolean repeat)</a> Use this to make the playlist repeat or not...
void	<a href="#">setSwitchLog(boolean switchLog)</a> Log when a playlist switch occurs

**Methods inherited from class java.lang.Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

**Methods inherited from interface java.lang.Runnable**

run

---

## Constructors

### Stream

```
public Stream()
```

---

## Methods

### createInstance

```
public static Stream createInstance(IVHost vhost,  
    String applicationName,  
    String sName)
```

Use this to create a named Stream on the default instance (\_definst\_) of an application on a particular VHost.

**Parameters:**

vhost - - Virtual Host  
applicationName - - Application name  
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

---

(continued from last page)

## createInstance

```
public static Stream createInstance(IApplicationInstance appInstance,  
    String sName)
```

Use this to create a named Stream on an application instance.

### Parameters:

appInstance -- Application instance  
sName -- Name of Stream

### Returns:

stream interface

---

## run

```
public void run()
```

Overridden from class Runnable .... do NOT call this directly. This method handles switching between media sources in the playlist.

---

## getPlaylist

```
public java.util.List getPlaylist()
```

Get the current playlist

### 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.

(continued from last page)

## addToPlaylist

```
public boolean addToPlaylist(int index,  
    String name,  
    int start,  
    int length)
```

Inserts a media source item to this playlist at a particular index, without interruption.

### Parameters:

index -- insertion index  
name -- name of media item being inserted  
start -- where to start playing the item. (-2 implies play a live stream)  
length -- how much of the item to play (-1 implies play the entire file or live stream)

### Returns:

- returns true if item was inserted successfully ...otherwise false. this should not interrupt anything that might be currently playing.... will do nothing if existing item does not exist.. if existing == "" insert at head of list

---

## addToPlaylist

```
public boolean addToPlaylist(String existing,  
    String name,  
    int start,  
    int length)
```

Insert item into playlist just after the first item in the playlist with a given name.

### Parameters:

existing - name of playlist item in which to insert the item after  
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(String name)
```

Remove all items matching the given stream name from the playlist. If the item is currently being played, it is not removed.

### Parameters:

name - stream name

### Returns:

true if at least one item from removed

---

## play

```
public boolean play(String sPlaylist)
```

Add a media item to the playlist as defined by an XML file ..

### Parameters:

sPlaylist -- the playlist XML definition file

### Returns:

- returns true if item was added successfully ...otherwise false.

## close

```
public void close()
```

Call this method when you have finished with the playlist object. It will terminate the playlist thread. The playlist object cannot be used after this call.

---

## getName

```
public String getName()
```

Returns the name of the playlist stream - the client would play this stream by this name.

**Returns:**

- the name of the playlist stream

---

## getRepeat

```
public boolean getRepeat()
```

Use this to determine if the playlist is auto-repeating

**Returns:**

- true if repeating otherwise false.

---

## setRepeat

```
public void setRepeat(boolean repeat)
```

Use this to make the playlist repeat or not...

**Parameters:**

repeat - - true to repeat otherwise false

---

## play

```
public void play(int n)
```

---

## next

```
public void next(int n)
```

---

## next

```
public void next()
```

---

## previous

```
public void previous()
```

---



## previous

```
public void previous(int n)
```

---

## getPollingInterval

```
public int getPollingInterval()
```

Get the polling interval (milliseconds)

**Returns:**

polling interval (milliseconds)

---

## setPollingInterval

```
public void setPollingInterval(int pollingInterval)
```

Set the polling interval (milliseconds)

**Parameters:**

pollingInterval - polling interval (milliseconds)

---

## getPublisher

```
public Publisher getPublisher()
```

---

## addListener

```
public void addListener(IStreamActionNotify listener)
```

Add a listener

**Parameters:**

listener - listener

---

## removeListener

```
public void removeListener(IStreamActionNotify listener)
```

Remove a listener

**Parameters:**

listener - listener

---

## isSwitchLog

```
public boolean isSwitchLog()
```

Log when a playlist switch occurs

**Returns:**

log when a playlist switch occurs

---

## setSwitchLog

```
public void setSwitchLog(boolean switchLog)
```

Log when a playlist switch occurs

**Parameters:**

switchLog - log when a playlist switch occurs

---

Package

**com.wowza.wms.util**

## com.wowza.wms.util Class RTPUtils

java.lang.Object

└─com.wowza.wms.util.RTPUtils

public class **RTPUtils**  
extends Object

### Constructor Summary

public	<a href="#">RTPUtils()</a>
--------	----------------------------

### Method Summary

static double[]	<a href="#">decodeRangeHeader</a> (String rangeStr)
static RTPStream	<a href="#">decodeStreamInfo</a> (RTPContext context, String streamId, String streamInfo)
static byte[]	<a href="#">formatH264CodecConfig</a> (byte[] sps, java.util.List ppss, byte[] profileLevel)
static byte[]	<a href="#">formatH264CodecConfigPacket</a> (byte[] sps, java.util.List ppss, byte[] profileLevel)
static void	<a href="#">loadConfigFile</a> (RTPContext rtpContext, String fileURL)
static <a href="#">RTPPushPublishSession</a>	<a href="#">startRTPPull</a> ( <a href="#">IApplicationInstance</a> appInstance, String streamName, boolean streamPacketizer, String ipAddress, int streamPort)
static <a href="#">RTPPushPublishSession</a>	<a href="#">startRTPPull</a> ( <a href="#">IApplicationInstance</a> appInstance, String streamName, boolean streamPacketizer, String ipAddress, int streamPort, boolean isRTPWrapped)
static <a href="#">RTPPushPublishSession</a>	<a href="#">startRTPPull</a> ( <a href="#">IApplicationInstance</a> appInstance, String streamName, boolean streamPacketizer, String ipAddress, int audioPort, int videoPort)
static <a href="#">RTPPushPublishSession</a>	<a href="#">startRTPPull</a> ( <a href="#">IApplicationInstance</a> appInstance, String streamName, boolean streamPacketizer, String ipAddress, int audioPort, int videoPort, boolean isRTPWrapped)
static <a href="#">RTPPushPublishSession</a>	<a href="#">startRTPPull</a> ( <a href="#">IApplicationInstance</a> appInstance, String streamName, <a href="#">RTPDestination</a> rtpDestination)
static void	<a href="#">stopRTPPull</a> ( <a href="#">RTPPushPublishSession</a> rtpPushPublishSession)
static String	<a href="#">updateSDPDestination</a> ( <a href="#">RTPDestination</a> rtpDestination, String sdpData)

static void	<a href="#">writeCodecConfig</a> (RTPTrack rtpTrack, long adjTimecode, byte[] codecConfig)
-------------	--

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

## Constructors

### RTPUtils

```
public RTPUtils()
```

## Methods

### updateSDPDestination

```
public static String updateSDPDestination(RTPDestination rtpDestination,  
    String sdpData)
```

### decodeRangeHeader

```
public static double[] decodeRangeHeader(String rangeStr)
```

### stopRTPPull

```
public static void stopRTPPull(RTPPushPublishSession rtpPushPublishSession)
```

### startRTPPull

```
public static RTPPushPublishSession startRTPPull(IApplicationInstance appInstance,  
    String streamName,  
    boolean streamPacketizer,  
    String ipAddress,  
    int streamPort,  
    boolean isRTPWrapped)
```

### startRTPPull

```
public static RTPPushPublishSession startRTPPull(IApplicationInstance appInstance,  
    String streamName,  
    boolean streamPacketizer,  
    String ipAddress,  
    int streamPort)
```

---

## startRTPPull

```
public static RTPPushPublishSession startRTPPull(IApplicationInstance appInstance,  
        String streamName,  
        boolean streamPacketizer,  
        String ipAddress,  
        int audioPort,  
        int videoPort)
```

---

## startRTPPull

```
public static RTPPushPublishSession startRTPPull(IApplicationInstance appInstance,  
        String streamName,  
        boolean streamPacketizer,  
        String ipAddress,  
        int audioPort,  
        int videoPort,  
        boolean isRTPWrapped)
```

---

## startRTPPull

```
public static RTPPushPublishSession startRTPPull(IApplicationInstance appInstance,  
        String streamName,  
        RTPDestination rtpDestination)
```

---

## writeCodecConfig

```
public static void writeCodecConfig(RTPTrack rtpTrack,  
        long adjTimecode,  
        byte[] codecConfig)
```

---

## loadConfigFile

```
public static void loadConfigFile(RTPContext rtpContext,  
        String fileURL)
```

---

## decodeStreamInfo

```
public static RTPStream decodeStreamInfo(RTPContext context,  
        String streamId,  
        String streamInfo)
```

---

## formatH264CodecConfigPacket

```
public static byte[] formatH264CodecConfigPacket(byte[] sps,  
        java.util.List ppss,  
        byte[] profileLevel)
```

---

## **formatH264CodecConfig**

```
public static byte[] formatH264CodecConfig(byte[] sps,  
      java.util.List ppss,  
      byte[] profileLevel)
```

---

Package

**com.wowza.wms.vhost**



## com.wowza.wms.vhost Class HostPort

java.lang.Object

└─com.wowza.wms.vhost.HostPort

public class **HostPort**  
extends Object

HostPort: data object that describes a socket connection. The address can be defined by ipAddress or by domainName. It can also contain a reference to an SSLFactory class that can be used to create a secure connection to the server.

### Constructor Summary

public	<a href="#">HostPort()</a> Create an empty HostPort object
--------	---

### Method Summary

void	<a href="#">addHttpProvider(IHTTPProvider httpProvider)</a>
void	<a href="#">addHttpProvider(IHTTPProvider2 httpProvider)</a>
void	<a href="#">addHTTPStreamerAdapterID(String ID)</a>
void	<a href="#">configureSocketAcceptor(org.apache.mina.transport.socket.nio.SocketAcceptorConfig socketConfig)</a> Configure a socketAcceptor
java.net.InetAddress	<a href="#">getAddress()</a> Get the ipAddress as an InetAddress object
String	<a href="#">getAddressRawStr()</a>
String	<a href="#">getAddressStr()</a> Get a String representation of the address
HostPortConfig	<a href="#">getConfiguration()</a> Get the socket configuration
java.util.List	<a href="#">getHttpProviders()</a>
java.util.List	<a href="#">getHTTPStreamerAdapterIDs()</a>
int	<a href="#">getPort()</a> Get port
int	<a href="#">getProcessorCount()</a> Get the number of threads to use to service this incoming port

HostPortSSLConfig	<a href="#">getSSLConfig()</a>
String	<a href="#">getSslFactoryClass()</a> Get full class name or SSLFactory class
boolean	<a href="#">isSuspended()</a>
void	<a href="#">setDomainName(String domainName)</a> Set domainName.
void	<a href="#">setIpAddress(String ipAddress)</a> Set ipAddress for object.
void	<a href="#">setPort(int port)</a> Set port
void	<a href="#">setProcessorCount(int processorCount)</a> Set the number of threads to use to service this incoming port
void	<a href="#">setSSLConfig(HostPortSSLConfig sslConfig)</a>
void	<a href="#">setSslFactoryClass(String sslFactoryClass)</a> Set full class name of SSLFactory class
void	<a href="#">setSuspended(boolean isSuspended)</a>
String	<a href="#">toString()</a> Return object as formatted string
String	<a href="#">toString(boolean mBeanSafe)</a> Return object as formatted string

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructors

### HostPort

```
public HostPort()
```

Create an empty HostPort object

## Methods

### setDomainName

```
public void setDomainName(String domainName)
```

Set domainName. A domainName of \* defines a local connection.

#### Parameters:

domainName - domainName like www.mycompany.com or \* for local

## getAddress

```
public java.net.InetAddress getAddress()
```

Get the ipAddress as an InetAddress object

**Returns:**

ipAddress as an InetAddress object. At this point the domainName (if specified) will be resolved.

---

## getAddressStr

```
public String getAddressStr()
```

Get a String representation of the address

**Returns:**

String representation of the address/domainName. If \* it will return [any]

---

## getAddressRawStr

```
public String getAddressRawStr()
```

---

## setIpAddress

```
public void setIpAddress(String ipAddress)
```

Set ipAddress for object. A ipAddress of \* defines a local connection.

**Parameters:**

ipAddress - ipAddress like 127.0.0.1 or \* for local

---

## getPort

```
public int getPort()
```

Get port

**Returns:**

port

---

## setPort

```
public void setPort(int port)
```

Set port

**Parameters:**

port - port

---

## getSslFactoryClass

```
public String getSslFactoryClass()
```

Get full class name or SSLFactory class

---

(continued from last page)

**Returns:**full class name or SSLFactory or "" if not specified

---

**setSslFactoryClass**

```
public void setSslFactoryClass(String sslFactoryClass)
```

Set full class name of SSLFactory class

**Parameters:**sslFactoryClass - full class name or SSLFactory or "" if not specified

---

**getProcessorCount**

```
public int getProcessorCount()
```

Get the number of threads to use to service this incoming port

**Returns:**number of processor threads

---

**setProcessorCount**

```
public void setProcessorCount(int processorCount)
```

Set the number of threads to use to service this incoming port

**Parameters:**processorCount - number of processor threads

---

**toString**

```
public String toString(boolean mBeanSafe)
```

Return object as formatted string

**Parameters:**

mBeanSafe - make the name safe for JMX management interface

**Returns:**formmatted string

---

**toString**

```
public String toString()
```

Return object as formatted string

**Returns:**formmatted string

---

**getConfiguration**

```
public HostPortConfig getConfiguration()
```

Get the socket configuration

**Returns:**

(continued from last page)

socket configuration

---

## configureSocketAcceptor

```
public void  
configureSocketAcceptor(org.apache.mina.transport.socket.nio.SocketAcceptorConfig  
socketConfig)
```

Configure a socketAcceptor

### Parameters:

socketConfig - socket acceptor

---

## getHttpProviders

```
public java.util.List getHttpProviders()
```

---

## addHttpProvider

```
public void addHttpProvider(IHTTPProvider httpProvider)
```

---

## addHttpProvider

```
public void addHttpProvider(IHTTPProvider2 httpProvider)
```

---

## isSuspended

```
public boolean isSuspended()
```

---

## setSuspended

```
public void setSuspended(boolean isSuspended)
```

---

## getHTTPStreamerAdapterIDs

```
public java.util.List getHTTPStreamerAdapterIDs()
```

---

## addHTTPStreamerAdapterID

```
public void addHTTPStreamerAdapterID(String ID)
```

---

## getSSLConfig

```
public HostPortSSLConfig getSSLConfig()
```

---

(continued from last page)

---

## setSSLConfig

```
public void setSSLConfig(HostPortSSLConfig sslConfig)
```

## com.wowza.wms.vhost Class HostPortList

java.lang.Object

└--com.wowza.wms.vhost.HostPortList

```
public class HostPortList
extends Object
```

HostPortList: data object that contains a collection of HostPort objects.

### Constructor Summary

public	<a href="#">HostPortList</a> () Create empty HostPortList
--------	--

### Method Summary

void	<a href="#">add</a> ( <a href="#">HostPort</a> hostPort) Add HostPort object
<a href="#">HostPort</a>	<a href="#">get</a> (int index) Get HostPort object at index, null if out of bounds
int	<a href="#">size</a> () Get number of HostPort objects

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructors

### HostPortList

```
public HostPortList()
```

Create empty HostPortList

## Methods

### add

```
public void add(HostPort hostPort)
```

Add HostPort object

#### Parameters:

hostPort

## size

```
public int size()
```

Get number of HostPort objects

**Returns:**

number of HostPort objects

---

## get

```
public HostPort get(int index)
```

Get HostPort object at index, null if out of bounds

**Parameters:**

index - index

**Returns:**

HostPort object at index, null if out of bounds

---



## com.wowza.wms.vhost Interface IAcceptorNotify

public interface **IAcceptorNotify**  
extends

IAcceptorNotify: listener interface used by IVHost addAcceptorListener

### Method Summary

void	<a href="#"><code>onAcceptorCreate</code></a> ( <a href="#"><code>HostPort</code></a> hostPort, java.util.Map acceptorMap) Triggered when a new acceptor is created
void	<a href="#"><code>onAcceptorDestroy</code></a> ( <a href="#"><code>HostPort</code></a> hostPort, java.util.Map acceptorMap) Triggered when a new acceptor is deleted

### Methods

#### **onAcceptorCreate**

```
public void onAcceptorCreate(HostPort hostPort,  
                             java.util.Map acceptorMap)
```

Triggered when a new acceptor is created

**Parameters:**

hostPort - HostPort object

acceptorMap - Map of acceptors

#### **onAcceptorDestroy**

```
public void onAcceptorDestroy(HostPort hostPort,  
                               java.util.Map acceptorMap)
```

Triggered when a new acceptor is deleted

**Parameters:**

hostPort - HostPort object

acceptorMap - Map of acceptors

## com.wowza.wms.vhost Interface IVHost

public interface **IVHost**  
extends

IVHost: public interface to VHost object.

### Field Summary

public static final	<a href="#">ACCEPTORS_ACCEPTOR</a> Acceptor types: acceptor Value: <b>acceptor</b>
public static final	<a href="#">ACCEPTORS_HANDLERADAPTER</a> Acceptor types: handler Value: <b>handlerAdapter</b>
public static final	<a href="#">CODEC_AUDIO_AAC</a> Value: <b>10</b>
public static final	<a href="#">CODEC_AUDIO_G711_ALAW</a> Value: <b>7</b>
public static final	<a href="#">CODEC_AUDIO_G711_MULAW</a> Value: <b>8</b>
public static final	<a href="#">CODEC_AUDIO_MP3</a> Value: <b>2</b>
public static final	<a href="#">CODEC_AUDIO_MP3_8</a> Value: <b>15</b>
public static final	<a href="#">CODEC_AUDIO_NELLYMOSER</a> Value: <b>6</b>
public static final	<a href="#">CODEC_AUDIO_NELLYMOSER_16MONO</a> Value: <b>4</b>
public static final	<a href="#">CODEC_AUDIO_NELLYMOSER_8MONO</a> Value: <b>5</b>
public static final	<a href="#">CODEC_AUDIO_PCM_BE</a> Value: <b>0</b>
public static final	<a href="#">CODEC_AUDIO_PCM_LE</a> Value: <b>3</b>

public static final	<a href="#">CODEC_AUDIO_PCM_SWF</a> Value: <b>1</b>
public static final	<a href="#">CODEC_AUDIO_RESERVED</a> Value: <b>9</b>
public static final	<a href="#">CODEC_AUDIO_SPEEX</a> Value: <b>11</b>
public static final	<a href="#">CODEC_AUDIO_UNKNOWN</a> Value: <b>-1</b>
public static final	<a href="#">CODEC_AUDIO_VORBIS</a> Value: <b>9</b>
public static final	<a href="#">CODEC_STREAM_MP2T</a> Value: <b>0</b>
public static final	<a href="#">CODEC_STREAM_UNKNOWN</a> Value: <b>-1</b>
public static final	<a href="#">CODEC_VIDEO_H264</a> Value: <b>7</b>
public static final	<a href="#">CODEC_VIDEO_SCREEN</a> Value: <b>3</b>
public static final	<a href="#">CODEC_VIDEO_SCREEN2</a> Value: <b>6</b>
public static final	<a href="#">CODEC_VIDEO_SPARK</a> Value: <b>2</b>
public static final	<a href="#">CODEC_VIDEO_UNKNOWN</a> Value: <b>-1</b>
public static final	<a href="#">CODEC_VIDEO_VP6</a> Value: <b>4</b>
public static final	<a href="#">CODEC_VIDEO_VP6A</a> Value: <b>5</b>
public static final	<a href="#">CODEC_VIDEO_VP8</a> Value: <b>8</b>
public static final	<a href="#">CONTENTTYPE_AUDIO</a> AMF Content type: audio packet Value: <b>8</b>

public static final	<a href="#">CONTENTTYPE_BUFFERSIZE</a> AMF Content type: set buffer size Value: <b>4</b>
public static final	<a href="#">CONTENTTYPE_DATA</a> AMF Content type: data packet Value: <b>18</b>
public static final	<a href="#">CONTENTTYPE_DATA0</a> AMF Content type: data packet (AMF0) Value: <b>18</b>
public static final	<a href="#">CONTENTTYPE_DATA3</a> AMF Content type: data packet (AMF3) Value: <b>15</b>
public static final	<a href="#">CONTENTTYPE_FUNCTION</a> AMF Content type: function data (AMF0) Value: <b>20</b>
public static final	<a href="#">CONTENTTYPE_FUNCTION0</a> AMF Content type: function data (AMF0) Value: <b>20</b>
public static final	<a href="#">CONTENTTYPE_FUNCTION3</a> AMF Content type: function data (AMF3) Value: <b>17</b>
public static final	<a href="#">CONTENTTYPE_MEDIACHUNK</a> AMF Content type: media chunk Value: <b>22</b>
public static final	<a href="#">CONTENTTYPE_PLAYCALLBACK</a> AMF Content type: play callback Value: <b>127</b>
public static final	<a href="#">CONTENTTYPE_SETCHUNKSIZE</a> AMF Content type: set packet chunk size Value: <b>1</b>
public static final	<a href="#">CONTENTTYPE_SHAREDOBJECTS</a> AMF Content type: shared object packet (AMF0) Value: <b>19</b>
public static final	<a href="#">CONTENTTYPE_SHAREDOBJECTS0</a> AMF Content type: shared object packet (AMF0) Value: <b>19</b>
public static final	<a href="#">CONTENTTYPE_SHAREDOBJECTS3</a> AMF Content type: shared object packet (AMF3) Value: <b>16</b>
public static final	<a href="#">CONTENTTYPE_UNKNOWN</a> AMF Content type: unknown Value: <b>0</b>

public static final	<a href="#">CONTENTTYPE_VIDEO</a> AMF Content type: video packet Value: <b>9</b>
public static final	<a href="#">CONTENTTYPE_WATCHDOG</a> AMF Content type: watch dog Value: <b>3</b>
public static final	<a href="#">COUNTER_HTTPCUPERTINO</a> Value: <b>2</b>
public static final	<a href="#">COUNTER_HTTPSANJOSE</a> Value: <b>4</b>
public static final	<a href="#">COUNTER_HTTPSMOOTH</a> Value: <b>3</b>
public static final	<a href="#">COUNTER_HTTPWEBM</a> Value: <b>4</b>
public static final	<a href="#">COUNTER_RTMP</a> Value: <b>0</b>
public static final	<a href="#">COUNTER_RTP</a> Value: <b>1</b>
public static final	<a href="#">COUNTER_TOTAL</a> Value: <b>6</b>
public static final	<a href="#">FILEFORMAT_FLV</a> Value: <b>1</b>
public static final	<a href="#">FILEFORMAT_MP4</a> Value: <b>2</b>
public static final	<a href="#">FILEFORMAT_UNKNOWN</a> Value: <b>-1</b>
public static final	<a href="#">VHOST_DEFAULT</a> Value: <b>_defaultVHost_</b>

## Method Summary

void	<a href="#">addAcceptorListener</a> ( <a href="#">IAcceptorNotify</a> acceptorListener) Add acceptor listener.
void	<a href="#">addApplicationListener</a> ( <a href="#">IApplicationNotify</a> applicationListener) Add application listener.
void	<a href="#">addIdleWorkerListener</a> ( <a href="#">IIdleWorkerNotify</a> idleWorkerListener) Add idleWorker listener.

void	<a href="#"><u>addStartupStream</u></a> (StartupStream startupStream) Add a stream to the list of streams to start and virtual host startup
boolean	<a href="#"><u>applicationExists</u></a> (String name) Return true if an application folder exists for this application name
void	<a href="#"><u>closeHostPort</u></a> (HostPort hostPort, boolean isSuspend) Close an individual HostPort
boolean	<a href="#"><u>createApplication</u></a> (String sName, String sStreamType, String sContentLoc) Method to create a new application
<a href="#"><u>IApplication</u></a>	<a href="#"><u>getApplication</u></a> (String applicationName) Get application by name.
java.util.List	<a href="#"><u>getApplicationFolderNames</u></a> () Get a list of application folder names
edu.emory.mathcs.backport.java.util.concurrent.locks.WMSReadWriteLock	<a href="#"><u>getApplicationLock</u></a> () Get the object used in synchronized statements to lock and application.
java.util.List	<a href="#"><u>getApplicationNames</u></a> () Get a list of application names
int	<a href="#"><u>getApplicationTimeout</u></a> () Get application time out (milliseconds).
AuthenticationList	<a href="#"><u>getAuthenticationList</u></a> () Get the list of available authentication methods
<a href="#"><u>IClient</u></a>	<a href="#"><u>getClient</u></a> (int clientId) Get client by client id.
<a href="#"><u>IClient</u></a>	<a href="#"><u>getClient</u></a> (int clientId, boolean create) Get client by client id and create if does not exist.
int	<a href="#"><u>getClientCount</u></a> () Get number of clients connected to this vHost.
int	<a href="#"><u>getClientIdleFrequency</u></a> () Get default client idle frequency (milliseconds)
int	<a href="#"><u>getClientTimeout</u></a> () Get client timeout.
<a href="#"><u>ConnectionCounter</u></a>	<a href="#"><u>getConnectionCounter</u></a> () Get vHost connection counter.
ConnectionCounterSimple	<a href="#"><u>getConnectionCounter</u></a> (int counterIndex) Get vHost connection counter for a specific technology (see IVHost.COUNTER_*)
int	<a href="#"><u>getConnectionLimit</u></a> () Get vHost connection limit.
int	<a href="#"><u>getCoreHandlerPoolSize</u></a> () Get the handler core thread pool size.

int	<a href="#"><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.
java.util.Properties	<a href="#"><code>getDynamicLogProperties()</code></a> Get the dynamic log properties defined at the vhost level in conf/log4j.properties
int	<a href="#"><code>getFileIOPoolSize()</code></a> Get the default file io pool size.
<a href="#"><code>ThreadPool</code></a>	<a href="#"><code>getHandlerThreadPool()</code></a> Get the VHost handler thread pool.
String	<a href="#"><code>getHomePath()</code></a> Get vHost configuration path.
<a href="#"><code>HostPortList</code></a>	<a href="#"><code>getHostPortsList()</code></a> Get list of host port definitions for vHost.
<a href="#"><code>IHTTPStreamerAdapter</code></a>	<a href="#"><code>getHTTPStreamerAdapter(String ID)</code></a> Get an HTTPStreamerAdapter by ID
java.util.List	<a href="#"><code>getHTTPStreamerAdapterIDs()</code></a> Get a list of HTTPStreamerAdapter IDs
HTTPStreamerContext	<a href="#"><code>getHTTPStreamerContext()</code></a> Get the HTTPStreamer (Cupertino Streaming and Silverlight Smooth Streaming) host context
HTTPStreamerList	<a href="#"><code>getHTTPStreamerList()</code></a> Get the list of HTTPStreamers
int	<a href="#"><code>getIdleCheckFrequency()</code></a> Get idle check frequency (milliseconds)
int	<a href="#"><code>getIdleMinimumWaitTime()</code></a> Get the minimum time (milliseconds) the idle worker thread will sleep before generating idle events
int	<a href="#"><code>getIdleWorkerCount()</code></a> Get number of threads used to generate idle events
IdleWorkersUtil	<a href="#"><code>getIdleWorkers()</code></a> Get the idle worker utility
<a href="#"><code>IOPerformanceCounter</code></a>	<a href="#"><code>getIoPerformanceCounter()</code></a> Get vHost IO performance counter.
<a href="#"><code>IOPerformanceCounter</code></a>	<a href="#"><code>getIoPerformanceCounter(int counterIndex)</code></a> Get vHost IO performance counter for a specific technology (see IVHost.COUNTER_*)
IOScheduler	<a href="#"><code>getIOScheduler()</code></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

int	<a href="#"><u>getMaximumPendingWriteBytes</u></a> ( ) Get maximum number a bytes a client connection can have waiting to be sent before the connection is terminated.
int	<a href="#"><u>getMaximumSetBufferTime</u></a> ( ) Get maximum number of milliseconds allowed for the NetStream.setBufferTime(secs) call.
<a href="#"><u>MediaCasterList</u></a>	<a href="#"><u>getMediaCasterList</u></a> ( ) Get the list of media caster definitions (MediaCaster.xml)
<a href="#"><u>MediaCasterSettings</u></a>	<a href="#"><u>getMediaCasterSettings</u></a> ( ) Get the media caster settings
MediaReaderList	<a href="#"><u>getMediaReaders</u></a> ( ) Get the media readers attached to vHost (MediaReaders.xml).
MediaWriterList	<a href="#"><u>getMediaWriters</u></a> ( ) Get the media writers attached to vHost (MediaWriters.xml).
java.util.Map	<a href="#"><u>getMp3TagMap</u></a> ( ) Get MP3 tag map attached to vHost (MP3Tags.xml).
String	<a href="#"><u>getName</u></a> ( ) Get vHost name
HostPortConfig	<a href="#"><u>getNetConnectionHostPortConfig</u></a> ( ) Get the socket configuration for server to server connections
int	<a href="#"><u>getNetConnectionIdleFrequency</u></a> ( ) Get server to server idle frequency (milliseconds)
int	<a href="#"><u>getNetConnectionProcessorCount</u></a> ( ) Get net connection processor count.
int	<a href="#"><u>getNextNetConnectionId</u></a> ( ) Get next connection id.
int	<a href="#"><u>getPingTimeout</u></a> ( ) Get ping timeout (milliseconds)
<a href="#"><u>WMSProperties</u></a>	<a href="#"><u>getProperties</u></a> ( ) Get properties attached to this vHost.
String	<a href="#"><u>getProperty</u></a> (String key) Get virtual host property.
RTPContext	<a href="#"><u>getRTPContext</u></a> ( ) Get the RTP (real time protocol) virtual host context
com.wowza.wms.rtp.transport.RTPUDPDatagramConfig	<a href="#"><u>getRTPDatagramConfigIncoming</u></a> ( ) Get the RTP Datagram Socket configuration
com.wowza.wms.rtp.transport.RTPUDPDatagramConfig	<a href="#"><u>getRTPDatagramConfigOutgoing</u></a> ( ) Get the RTP Datagram Socket configuration
int	<a href="#"><u>getRTPIidleFrequency</u></a> ( ) Get default RTP idle frequency (milliseconds)



java.util.List	<a href="#"><u>getStartupStreams()</u></a> Get the list of streams to start at virtual host startup
<a href="#"><u>StreamList</u></a>	<a href="#"><u>getStreamTypes()</u></a> Get default stream type.
<a href="#"><u>ThreadPool</u></a>	<a href="#"><u>getThreadPool()</u></a> Get the VHost handler thread pool.
String	<a href="#"><u>getTimeRunning()</u></a> Get the time vHost has been running.
double	<a href="#"><u>getTimeRunningSeconds()</u></a> Get time running in seconds
<a href="#"><u>ThreadPool</u></a>	<a href="#"><u>getTransportThreadPool()</u></a> Get the VHost transport thread pool.
com.wowza.wms.rtp.transport.UDPTransportManager	<a href="#"><u>getUDPTransportManager()</u></a> Get the UDP transport manager.
int	<a href="#"><u>getValidationFrequency()</u></a> Get time between validation pings (milliseconds)
void	<a href="#"><u>init()</u></a> (String basePath) Initialize vHost.
boolean	<a href="#"><u>isApplicationLoaded()</u></a> (String applicationName) Return true is the application is loaded
boolean	<a href="#"><u>isShuttingDown()</u></a> Is the VHost shutting down
boolean	<a href="#"><u>isStartStarupStreams()</u></a> Returns true if the startup streams are to start and vhost startup
boolean	<a href="#"><u>isSuspended()</u></a> Returns true is all HostPorts connected to this VHost are suspended
void	<a href="#"><u>killClient()</u></a> (int clientId) Remove client from vHost and send disconnect message.
void	<a href="#"><u>killRTSPSession()</u></a> (String rtspSessionId) Kill an RTSP connection by the RTSP session id
void	<a href="#"><u>putHTTPStreamerAdapter()</u></a> (String ID, <a href="#"><u>IHTTPStreamerAdapter</u></a> adapter) Add an HTTPStreamerAdapter
String	<a href="#"><u>readVHostConfig()</u></a> (String sName) Method to read xml config file..
void	<a href="#"><u>removeAcceptorListener()</u></a> ( <a href="#"><u>IAcceptorNotify</u></a> acceptorListener) Remove acceptor listener.
boolean	<a href="#"><u>removeApplication()</u></a> (String sName) Method to remove an application
void	<a href="#"><u>removeApplicationListener()</u></a> ( <a href="#"><u>IApplicationNotify</u></a> applicationListener) Remove applation listener.

void	<a href="#"><u>removeClient</u></a> (int clientId) Remove client from vHost.
void	<a href="#"><u>removeIdleWorkerListener</u></a> (IIIdleWorkerNotify idleWorkerListener) Remove idleWorker listener
void	<a href="#"><u>reparentClient</u></a> (IClient client) Move a client object to a new vhost.
void	<a href="#"><u>setAdminInterfaceHostPort</u></a> (HostPort adminInterfaceHostPort) Set admin interface host port (not used)
void	<a href="#"><u>setApplicationTimeout</u></a> (int applicationTimeout) Set application time out (milliseconds).
void	<a href="#"><u>setClientIdleFrequency</u></a> (int clientIdleFrequency) Set default client idle frequency (milliseconds)
void	<a href="#"><u>setClientTimeout</u></a> (int clientTimeout) Set client timeout.
void	<a href="#"><u>setCoreHandlerPoolSize</u></a> (int corePoolSize) Set the handler core thread pool size.
void	<a href="#"><u>setCoreTransportPoolSize</u></a> (int corePoolSize) Set the transport core thread pool size.
void	<a href="#"><u>setDynamicLogProperties</u></a> (java.util.Properties dynamicLogProperties) Set the dynamic log properties set at the vhost level
void	<a href="#"><u>setFileIOPoolSize</u></a> (int fileIOPoolSize) Set default file io thread pool size.
void	<a href="#"><u>setIdleCheckFrequency</u></a> (int idleCheckFrequency) Set idle check frequency (milliseconds)
void	<a href="#"><u>setIdleMinimumWaitTime</u></a> (int idleMinimumWaitTime) Set the minimum time (milliseconds) the idle worker thread will sleep before generating idle events
void	<a href="#"><u>setIdleWorkerCount</u></a> (int idleWorkerCount) Set number of threads used to generate idle events
void	<a href="#"><u>setKeepAliveTimeout</u></a> (int keepAliveTimeout) Set the RTMPT connection keep alive timeout
void	<a href="#"><u>setMaximumPendingWriteBytes</u></a> (int maximumPendingWriteBytes) Set maximum number a bytes a client connection can have waiting to be sent before the connection is terminated.
void	<a href="#"><u>setMaximumSetBufferTime</u></a> (int maximumSetBufferTime) Set maximum number of milliseconds allowed for the NetStream.setBufferTime(secs) call.
void	<a href="#"><u>setNetConnectionIdleFrequency</u></a> (int netConnectionIdleFrequency) Set the server to server idle frequency (milliseconds)
void	<a href="#"><u>setNetConnectionProcessorCount</u></a> (int netConnectionProcessorCount) Set net connection processor count.

void	<a href="#"><u>setPingTimeout</u></a> (int pingTimeout) Set ping timeout (milliseconds)
void	<a href="#"><u>setRTPIidleFrequency</u></a> (int rtpIdleFrequency) Set default RTP idle frequency (milliseconds)
void	<a href="#"><u>setShuttingDown</u></a> (boolean shuttingDown) Set VHost shutting down flag
void	<a href="#"><u>setStartStarupStreams</u></a> (boolean startStarupStreams) Set to true to startup startup stream as vhost startup
void	<a href="#"><u>setValidationFrequency</u></a> (int validationFrequency) Set time between validation pings (milliseconds)
void	<a href="#"><u>shutdown</u></a> ( ) Shutdown.
void	<a href="#"><u>shutdownApplication</u></a> (String appName) Shutdown an application by name.
boolean	<a href="#"><u>startApplicationInstance</u></a> (String appName) Start an application instance.
boolean	<a href="#"><u>startApplicationInstance</u></a> (String appName, String appInstanceName) Start an application instance.
void	<a href="#"><u>startStartupStreams</u></a> ( ) Method to start startup streams
void	<a href="#"><u>stopStartupStreams</u></a> ( ) Method to stop startup streams
void	<a href="#"><u>suspendAllHostPorts</u></a> ( ) Suspend all HostPorts from accepting new connections.
boolean	<a href="#"><u>touchApplicationInstance</u></a> (String appName) Start an application instance if it is not already started then touch it so it stays loaded for at least 3 seconds.
boolean	<a href="#"><u>touchApplicationInstance</u></a> (String appName, String appInstanceName) Start an application instance if it is not already started then touch it so it stays loaded for at least 3 seconds.
void	<a href="#"><u>unbindAllHostPorts</u></a> ( ) Unbind all HostPorts and drop all connections
void	<a href="#"><u>updateLoggingDuration</u></a> ( ) Internal: update the internal logging values.
boolean	<a href="#"><u>writeVHostConfig</u></a> (String sName, String data) Method to write xml config file..

## Fields

(continued from last page)

---

## VHOST\_DEFAULT

```
public static final java.lang.String VHOST_DEFAULT
```

Constant value: `_defaultVHost_`

---

## COUNTER\_RTMP

```
public static final int COUNTER_RTMP
```

Constant value: `0`

---

## COUNTER\_RTP

```
public static final int COUNTER_RTP
```

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: `4`

---

## COUNTER\_TOTAL

```
public static final int COUNTER_TOTAL
```

Constant value: `6`

---

(continued from last page)

---

## CODEC\_VIDEO\_UNKNOWN

```
public static final int CODEC_VIDEO_UNKNOWN
```

Constant value: **-1**

---

## CODEC\_VIDEO\_SPARK

```
public static final int CODEC_VIDEO_SPARK
```

Constant value: **2**

---

## CODEC\_VIDEO\_SCREEN

```
public static final int CODEC_VIDEO_SCREEN
```

Constant value: **3**

---

## CODEC\_VIDEO\_VP6

```
public static final int CODEC_VIDEO_VP6
```

Constant value: **4**

---

## CODEC\_VIDEO\_VP6A

```
public static final int CODEC_VIDEO_VP6A
```

Constant value: **5**

---

## CODEC\_VIDEO\_SCREEN2

```
public static final int CODEC_VIDEO_SCREEN2
```

Constant value: **6**

---

## CODEC\_VIDEO\_H264

```
public static final int CODEC_VIDEO_H264
```

Constant value: **7**

---

## CODEC\_VIDEO\_VP8

```
public static final int CODEC_VIDEO_VP8
```

Constant value: **8**

---

## CODEC\_AUDIO\_UNKNOWN

```
public static final int CODEC_AUDIO_UNKNOWN
```

(continued from last page)

Constant value: **-1**

---

## CODEC\_AUDIO\_PCM\_BE

```
public static final int CODEC_AUDIO_PCM_BE
```

Constant value: **0**

---

## CODEC\_AUDIO\_PCM\_SWF

```
public static final int CODEC_AUDIO_PCM_SWF
```

Constant value: **1**

---

## CODEC\_AUDIO\_MP3

```
public static final int CODEC_AUDIO_MP3
```

Constant value: **2**

---

## CODEC\_AUDIO\_PCM\_LE

```
public static final int CODEC_AUDIO_PCM_LE
```

Constant value: **3**

---

## CODEC\_AUDIO\_NELLYMOSER\_16MONO

```
public static final int CODEC_AUDIO_NELLYMOSER_16MONO
```

Constant value: **4**

---

## CODEC\_AUDIO\_NELLYMOSER\_8MONO

```
public static final int CODEC_AUDIO_NELLYMOSER_8MONO
```

Constant value: **5**

---

## CODEC\_AUDIO\_NELLYMOSER

```
public static final int CODEC_AUDIO_NELLYMOSER
```

Constant value: **6**

---

## CODEC\_AUDIO\_G711\_ALAW

```
public static final int CODEC_AUDIO_G711_ALAW
```

Constant value: **7**

---

## CODEC\_AUDIO\_G711\_MULAW

```
public static final int CODEC_AUDIO_G711_MULAW
```

Constant value: **8**

---

## CODEC\_AUDIO\_RESERVED

```
public static final int CODEC_AUDIO_RESERVED
```

Constant value: **9**

---

## CODEC\_AUDIO\_VORBIS

```
public static final int CODEC_AUDIO_VORBIS
```

Constant value: **9**

---

## CODEC\_AUDIO\_AAC

```
public static final int CODEC_AUDIO_AAC
```

Constant value: **10**

---

## CODEC\_AUDIO\_SPEEX

```
public static final int CODEC_AUDIO_SPEEX
```

Constant value: **11**

---

## CODEC\_AUDIO\_MP3\_8

```
public static final int CODEC_AUDIO_MP3_8
```

Constant value: **15**

---

## CODEC\_STREAM\_UNKNOWN

```
public static final int CODEC_STREAM_UNKNOWN
```

Constant value: **-1**

---

## CODEC\_STREAM\_MP2T

```
public static final int CODEC_STREAM_MP2T
```

Constant value: **0**

---

(continued from last page)

---

## FILEFORMAT\_UNKNOWN

```
public static final int FILEFORMAT_UNKNOWN
```

Constant value: **-1**

---

## FILEFORMAT\_FLV

```
public static final int FILEFORMAT_FLV
```

Constant value: **1**

---

## FILEFORMAT\_MP4

```
public static final int FILEFORMAT_MP4
```

Constant value: **2**

---

## CONTENTTYPE\_UNKNOWN

```
public static final byte CONTENTTYPE_UNKNOWN
```

AMF Content type: unknown  
Constant value: **0**

---

## CONTENTTYPE\_SETCHUNKSIZE

```
public static final byte CONTENTTYPE_SETCHUNKSIZE
```

AMF Content type: set packet chunk size  
Constant value: **1**

---

## CONTENTTYPE\_WATCHDOG

```
public static final byte CONTENTTYPE_WATCHDOG
```

AMF Content type: watch dog  
Constant value: **3**

---

## CONTENTTYPE\_BUFFERSIZE

```
public static final byte CONTENTTYPE_BUFFERSIZE
```

AMF Content type: set buffer size  
Constant value: **4**

---

## CONTENTTYPE\_AUDIO

```
public static final byte CONTENTTYPE_AUDIO
```

AMF Content type: audio packet  
Constant value: **8**

---

## CONTENTTYPE\_VIDEO

```
public static final byte CONTENTTYPE_VIDEO
```

---



(continued from last page)

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**

---

## 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.

---

(continued from last page)

**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.

**Parameters:**

clientId - client id

---

## killRTSPSession

```
public void killRTSPSession(String rtspSessionId)
```

Kill an RTSP connection by the RTSP session id

**Parameters:**

rtspSessionId - RTSP session id

---

## getApplication

```
public IApplication getApplication(String applicationName)
```

Get application by name.

**Parameters:**

applicationName - application name

**Returns:**

application

---

## isApplicationLoaded

```
public boolean isApplicationLoaded(String applicationName)
```

Return true is the application is loaded

**Parameters:**

applicationName - application name

**Returns:**

true if application is loaded

---

## getHomePath

```
public String getHomePath()
```

Get vHost configuration path.

---

---

(continued from last page)

**Returns:**  
configuration path

---

## getProperties

public [WMSProperties](#) **getProperties()**

Get properties attached to this vHost.

**Returns:**  
properties attached to this vHost

---

## getCoreTransportPoolSize

public int **getCoreTransportPoolSize()**

Get the transport core thread pool size.

**Returns:**  
default core thread pool size

---

## setCoreTransportPoolSize

public void **setCoreTransportPoolSize**(int corePoolSize)

Set the transport core thread pool size.

**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

---

(continued from last page)

---

## 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()
```

---

(continued from last page)

Get vHost name

**Returns:**

vHost name

---

## setAdminInterfaceHostPort

```
public void setAdminInterfaceHostPort(HostPort adminInterfaceHostPort)
```

Set admin interface host port (not used)

**Parameters:**

adminInterfaceHostPort - admin interface host port

---

## getClientCount

```
public int getClientCount()
```

Get number of clients connected to this vHost.

**Returns:**

number of clients connected to this vHost

---

## getNetConnectionProcessorCount

```
public int getNetConnectionProcessorCount()
```

Get net connection processor count. Number of threads used for server to server communication (not finished).

**Returns:**

net connection processor count

---

## setNetConnectionProcessorCount

```
public void setNetConnectionProcessorCount(int netConnectionProcessorCount)
```

Set net connection processor count. Number of threads used for server to server communication (not finished).

**Parameters:**

netConnectionProcessorCount - net connection processor count

---

## addApplicationListener

```
public void addApplicationListener(IApplicationNotify applicationListener)
```

Add application listener. An application listener will receive the following events: onApplicationCreate, onApplicationDestroy.

**Parameters:**

applicationListener - application listener

---

## removeApplicationListener

```
public void removeApplicationListener(IApplicationNotify applicationListener)
```

Remove application listener.

**Parameters:**

applicationListener - application listener

## addAcceptorListener

```
public void addAcceptorListener(IAcceptorNotify acceptorListener)
```

Add acceptor listener. Acceptor listeners will receive the following events: onAcceptorCreate, onAcceptorDestroy.

**Parameters:**

acceptorListener - acceptor listener

---

## removeAcceptorListener

```
public void removeAcceptorListener(IAcceptorNotify acceptorListener)
```

Remove acceptor listener.

**Parameters:**

acceptorListener - acceptor listener

---

## getIOScheduler

```
public IOScheduler getIOScheduler()
```

Get IO scheduler for vHost. IO scheduler is used to schedule reads from the disk to increase server throughput for static flv serving.

**Returns:**

IO scheduler for vHost

---

## getIoPerformanceCounter

```
public IoPerformanceCounter getIoPerformanceCounter()
```

Get vHost IO performance counter.

**Returns:**

io performance counter

---

## getIoPerformanceCounter

```
public IoPerformanceCounter getIoPerformanceCounter(int counterIndex)
```

Get vHost IO performance counter for a specific technology (see IVHost.COUNTER\_\*)

**Parameters:**

counterIndex - counter index (see IVHost.COUNTER\_\*)

**Returns:**

io performance counter

---

## getConnectionCounter

```
public ConnectionCounter getConnectionCounter()
```

Get vHost connection counter.

**Returns:**

connection counter

---



---

## getConnectionCounter

```
public ConnectionCounterSimple getConnectionCounter(int counterIndex)
```

Get vHost connection counter for a specific technology (see IVHost.COUNTER\_\*)

**Parameters:**

counterIndex - counter index (see IVHost.COUNTER\_\*)

**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

---

## getMp3TagMap

```
public java.util.Map getMp3TagMap()
```

Get MP3 tag map attached to vHost (MP3Tags.xml).

**Returns:**

MP3 tag map attached to vHost

---

## updateLoggingDuration

```
public void updateLoggingDuration()
```

Internal: update the internal logging values.

---

## getApplicationTimeout

```
public int getApplicationTimeout()
```

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)

---

(continued from last page)

**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

---

**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.

---

(continued from last page)

**Parameters:**

maximumSetBufferTime - maximum number of milliseconds allowed for the NetStream.setBufferTime(secs) call

---

## getApplicationLock

```
public edu.emory.mathcs.backport.java.util.concurrent.locks.WMSReadWriteLock  
getApplicationLock( )
```

Get the object used in synchronized statements to lock and application.

**Returns:**

object used in synchronized statements to lock and application

---

## getClientIdleFrequency

```
public int getClientIdleFrequency( )
```

Get default client idle frequency (milliseconds)

**Returns:**

default client idle frequency (milliseconds)

---

## setClientIdleFrequency

```
public void setClientIdleFrequency(int clientIdleFrequency)
```

Set default client idle frequency (milliseconds)

**Parameters:**

clientIdleFrequency - default client idle frequency (milliseconds)

---

## getRTPIdeFrequency

```
public int getRTPIdeFrequency( )
```

Get default RTP idle frequency (milliseconds)

**Returns:**

default RTP idle frequency (milliseconds)

---

## setRTPIdeFrequency

```
public void setRTPIdeFrequency(int rtpIdleFrequency)
```

Set default RTP idle frequency (milliseconds)

**Parameters:**

rtpIdleFrequency - RTP default client idle frequency (milliseconds)

---

## getNetConnectionIdleFrequency

```
public int getNetConnectionIdleFrequency( )
```

Get server to server idle frequency (milliseconds)

**Returns:**

server to server idle frequency (milliseconds)

---

---

## setNetConnectionIdleFrequency

```
public void setNetConnectionIdleFrequency(int netConnectionIdleFrequency)
```

Set the server to server idle frequency (milliseconds)

**Parameters:**

netConnectionIdleFrequency - server to server idle frequency (milliseconds)

---

## getIdleCheckFrequency

```
public int getIdleCheckFrequency( )
```

Get idle check frequency (milliseconds)

**Returns:**

idle check frequency (milliseconds)

---

## setIdleCheckFrequency

```
public void setIdleCheckFrequency(int idleCheckFrequency)
```

Set idle check frequency (milliseconds)

**Parameters:**

idleCheckFrequency - idle check frequency (milliseconds)

---

## getIdleWorkerCount

```
public int getIdleWorkerCount( )
```

Get number of threads used to generate idle events

**Returns:**

number of threads used to generate idle events

---

## setIdleWorkerCount

```
public void setIdleWorkerCount(int idleWorkerCount)
```

Set number of threads used to generate idle events

**Parameters:**

idleWorkerCount - number of threads used to generate idle events

---

## getKeepAliveTimeout

```
public int getKeepAliveTimeout( )
```

Get the RTMPT connection keep alive timeout

**Returns:**

RTMPT connection keep alive timeout

---

## setKeepAliveTimeout

```
public void setKeepAliveTimeout(int keepAliveTimeout)
```

---

(continued from last page)

Set the RTMPT connection keep alive timeout

**Parameters:**

keepAliveTimeout - RTMPT connection keep alive timeout

---

## addIdleWorkerListener

```
public void addIdleWorkerListener(IIIdleWorkerNotify idleWorkerListener)
```

Add idleWorker listener. An idleWorker listener will receive the following events: onIdleWorkerCreate, onIdleWorkerDestroy.

**Parameters:**

idleWorkerListener - idleWorker listener

---

## removeIdleWorkerListener

```
public void removeIdleWorkerListener(IIIdleWorkerNotify idleWorkerListener)
```

Remove idleWorker listener

**Parameters:**

idleWorkerListener - idleWorker listener

---

## getNetConnectionHostPortConfig

```
public HostPortConfig getNetConnectionHostPortConfig()
```

Get the socket configuration for server to server connections

**Returns:**

socket configuration for server to server connections

---

## getMediaCasterSettings

```
public MediaCasterSettings getMediaCasterSettings()
```

Get the media caster settings

**Returns:**

media caster settings

---

## getMediaCasterList

```
public MediaCasterList getMediaCasterList()
```

Get the list of media caster definitions (MediaCaster.xml)

**Returns:**

list of media caster definitions

---

## getApplicationNames

```
public java.util.List getApplicationNames()
```

Get a list of application names

**Returns:**

list of application names

## getApplicationFolderNames

```
public java.util.List getApplicationFolderNames()
```

Get a list of application folder names

**Returns:**

list of application folder names

---

## applicationExists

```
public boolean applicationExists(String name)
```

Return true if an application folder exists for this application name

**Parameters:**

name - application name

**Returns:**

true if an application folder exists for this application name

---

## getRTPContext

```
public RTPContext getRTPContext()
```

Get the RTP (real time protocol) virtual host context

**Returns:**

RTP (real time protocol) virtual host context

---

## getHTTPStreamerContext

```
public HTTPStreamerContext getHTTPStreamerContext()
```

Get the HTTPStreamer (Cupertino Streaming and Silverlight Smooth Streaming) host context

**Returns:**

HTTPStreamer (Cupertino Streaming and Silverlight Smooth Streaming) host context

---

## getRTPDatagramConfigIncoming

```
public com.wowza.wms.rtp.transport.RTPUDPDatagramConfig getRTPDatagramConfigIncoming()
```

Get the RTP Datagram Socket configuration

**Returns:**

RTP Datagram Socket configuration

---

## getRTPDatagramConfigOutgoing

```
public com.wowza.wms.rtp.transport.RTPUDPDatagramConfig getRTPDatagramConfigOutgoing()
```

Get the RTP Datagram Socket configuration

**Returns:**

RTP Datagram Socket configuration

---

## getAuthenticationList

```
public AuthenticationList getAuthenticationList()
```

Get the list of available authentication methods

**Returns:**

list of available authentication methods

---

## getIdleMinimumWaitTime

```
public int getIdleMinimumWaitTime()
```

Get the minimum time (milliseconds) the idle worker thread will sleep before generating idle events

**Returns:**

minimum time (milliseconds) the idle worker thread will sleep before generating idle events

---

## setIdleMinimumWaitTime

```
public void setIdleMinimumWaitTime(int idleMinimumWaitTime)
```

Set the minimum time (milliseconds) the idle worker thread will sleep before generating idle events

**Parameters:**

idleMinimumWaitTime - minimum time (milliseconds) the idle worker thread will sleep before generating idle events

---

## getIdleWorkers

```
public IdleWorkersUtil getIdleWorkers()
```

Get the idle worker utility

**Returns:**

idle worker utility

---

## isShuttingDown

```
public boolean isShuttingDown()
```

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)
```

---



(continued from last page)

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

---

## 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

(continued from last page)

---

## isSuspended

```
public boolean isSuspended()
```

Returns true if 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:**

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

---

---

## getStartupStreams

```
public java.util.List getStartupStreams()
```

Get the list of streams to start at virtual host startup

**Returns:**

list of streams to start at virtual host startup

---

## addStartupStream

```
public void addStartupStream(StartupStream startupStream)
```

Add a stream to the list of streams to start and virtual host startup

**Parameters:**

startupStream - startup stream

---

## isStartStarupStreams

```
public boolean isStartStarupStreams()
```

Returns true if the startup streams are to start and vhost startup

**Returns:**

true if the startup streams are to start and vhost startup

---

## setStartStarupStreams

```
public void setStartStarupStreams(boolean startStarupStreams)
```

Set to true to startup startup stream as vhost startup

**Parameters:**

startStarupStreams - true if the startup streams are to start and vhost startup

---

## startStartupStreams

```
public void startStartupStreams()
```

Method to start startup streams

---

## stopStartupStreams

```
public void stopStartupStreams()
```

Method to stop startup streams

---

## createApplication

```
public boolean createApplication(String sName,  
    String sStreamType,  
    String sContentLoc)
```

Method to create a new application

---

(continued from last page)

## **removeApplication**

```
public boolean removeApplication(String sName)
```

Method to remove an application

---

## **readVHostConfig**

```
public String readVHostConfig(String sName)
```

Method to read xml config file..

---

## **writeVHostConfig**

```
public boolean writeVHostConfig(String sName,  
                                String data)
```

Method to write xml config file..

## com.wowza.wms.vhost Interface IVHostItemNotify

public interface **IVHostItemNotify**  
extends

IVHostItemNotify: listener interface used by VHostList addVHostItemListener

VHostList is the list vHost definitions in VHosts.xml not the vHosts themselves. With this listener interface you can monitor the vHost definition changes.

### Method Summary

void	<a href="#">onVHostItemCreate</a> ( <a href="#">VHostItem</a> vhostItem) Triggered when vHostItem created
void	<a href="#">onVHostItemDestroy</a> ( <a href="#">VHostItem</a> vhostItem) Triggered when vHostItem destroyed
void	<a href="#">onVHostItemUpdate</a> ( <a href="#">VHostItem</a> vhostItem) Triggered when vHostItem updated

### Methods

#### onVHostItemCreate

public void **onVHostItemCreate**([VHostItem](#) vhostItem)

Triggered when vHostItem created

**Parameters:**

vhostItem - vhostItem

#### onVHostItemUpdate

public void **onVHostItemUpdate**([VHostItem](#) vhostItem)

Triggered when vHostItem updated

**Parameters:**

vhostItem - vhostItem

#### onVHostItemDestroy

public void **onVHostItemDestroy**([VHostItem](#) vhostItem)

Triggered when vHostItem destroyed

**Parameters:**

vhostItem - vhostItem

## com.wowza.wms.vhost Interface IVHostNotify

public interface **IVHostNotify**  
extends

IVHostNotify: listener interface used by VHostSingleton addVHostListener

### Method Summary

void	<a href="#">onVHostClientConnect</a> ( <a href="#">IVHost</a> vhost, <a href="#">IClient</a> inClient, <a href="#">RequestFunction</a> function, <a href="#">AMFDataList</a> params) Triggered before a client connects to this virtual host.
void	<a href="#">onVHostCreate</a> ( <a href="#">IVHost</a> vhost) Triggered when vHost created
void	<a href="#">onVHostInit</a> ( <a href="#">IVHost</a> vhost) Triggered when vHost initialized
void	<a href="#">onVHostShutdownComplete</a> ( <a href="#">IVHost</a> vhost) Triggered at the end of vhost shutdown
void	<a href="#">onVHostShutdownStart</a> ( <a href="#">IVHost</a> vhost) Triggered at the beginning of vhost shutdown

### Methods

#### onVHostCreate

public void **onVHostCreate**([IVHost](#) vhost)

Triggered when vHost created

**Parameters:**

vhost

#### onVHostInit

public void **onVHostInit**([IVHost](#) vhost)

Triggered when vHost initialized

**Parameters:**

vhost - vhost

#### onVHostShutdownStart

public void **onVHostShutdownStart**([IVHost](#) vhost)

Triggered at the beginning of vhost shutdown

(continued from last page)

**Parameters:**vhost - vhost

---

**onVHostShutdownComplete**

```
public void onVHostShutdownComplete(IVHost vhost)
```

Triggered at the end of vhost shutdown

**Parameters:**vhost - vhost

---

**onVHostClientConnect**

```
public void onVHostClientConnect(IVHost vhost,  
    IClient inClient,  
    RequestFunction function,  
    AMFDataList params)
```

Triggered before a client connects to this virtual host. Provides an opportunity to rewrite the information that is being used to connect.

**Parameters:**

vhost  
inClient - client object of the connection  
function - function  
params - parameters



---

## com.wowza.wms.vhost Interface IWorkerThreadClear

---

public interface **IWorkerThreadClear**  
extends

---

### Method Summary

void	<a href="#">clear()</a>
void	<a href="#">reset()</a>

---

### Methods

#### **reset**

public void **reset**()

---

#### **clear**

public void **clear**()

## com.wowza.wms.vhost Class StreamItem

java.lang.Object

└─com.wowza.wms.vhost.StreamItem

public class **StreamItem**  
extends Object

StreamItem: data object that defines a streamType.

### Constructor Summary

public	<a href="#">StreamItem</a> (String name, String baseClass, String playClass) Create a new streamItem
--------	---

### Method Summary

void	<a href="#">clearProperty</a> (String name) Clear property.
String	<a href="#">getBaseClass</a> () Get base class path.
String	<a href="#">getDescription</a> () Get streamType description.
String	<a href="#">getName</a> () Get streamType name.
String	<a href="#">getPlayClass</a> () Get play class path.
<a href="#">WMSProperties</a>	<a href="#">getProperties</a> () Get properties.
String	<a href="#">getProperty</a> (String name) Get property by name.
void	<a href="#">setBaseClass</a> (String baseClass) Set base class path.
void	<a href="#">setDescription</a> (String description) Set streamType description.
void	<a href="#">setName</a> (String name) Set streamType name.
void	<a href="#">setPlayClass</a> (String playClass) Set play class path.
void	<a href="#">setProperty</a> (String name, String value) Set property value.

String	<a href="#">toString()</a>
--------	----------------------------

Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## Constructors

### StreamItem

```
public StreamItem(String name,  
                  String baseClass,  
                  String playClass)
```

Create a new streamItem

**Parameters:**

`name` - streamType name  
`baseClass` - base class path  
`playClass` - play class path

## Methods

### getBaseClass

```
public String getBaseClass()
```

Get base class path.

**Returns:**

base clas path

### setBaseClass

```
public void setBaseClass(String baseClass)
```

Set base class path.

**Parameters:**

`baseClass` - base class path

### getName

```
public String getName()
```

Get streamType name.

**Returns:**

streamType name

### setName

```
public void setName(String name)
```

(continued from last page)

Set streamType name.

**Parameters:**

name - streamType name

---

## getPlayClass

```
public String getPlayClass()
```

Get play class path.

**Returns:**

play class path

---

## setPlayClass

```
public void setPlayClass(String playClass)
```

Set play class path.

**Parameters:**

playClass - play class path

---

## setProperty

```
public void setProperty(String name,  
                        String value)
```

Set property value.

**Parameters:**

name - property name

value - property value

---

## clearProperty

```
public void clearProperty(String name)
```

Clear property.

**Parameters:**

name - property name

---

## getProperty

```
public String getProperty(String name)
```

Get property by name.

**Parameters:**

name - property name

**Returns:**

property value

---

## getProperties

```
public WMSProperties getProperties()
```

---

(continued from last page)

Get properties.

**Returns:**

properties

---

## getDescription

```
public String getDescription()
```

Get streamType description.

**Returns:**

streamType description

---

## setDescription

```
public void setDescription(String description)
```

Set streamType description.

**Parameters:**

description - streamType description

---

## toString

```
public String toString()
```

## com.wowza.wms.vhost Class StreamList

java.lang.Object

└─com.wowza.wms.vhost.StreamList

```
public class StreamList
    extends Object
```

StreamList: collection of StreamItems

### Constructor Summary

public	<a href="#">StreamList()</a> Create empty StreamList
--------	---

### Method Summary

<a href="#">StreamItem</a>	<a href="#">getStreamDef(String name)</a> Get streamItem by streamType name.
java.util.Map	<a href="#">getStreamDefs()</a> Get Map of streamItems (by streamType names).
java.util.List	<a href="#">getStreamTypeNames()</a> Get list of streamType names.

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Constructors

#### StreamList

```
public StreamList()
```

Create empty StreamList

### Methods

#### getStreamDefs

```
public java.util.Map getStreamDefs()
```

Get Map of streamItems (by streamType names).

##### Returns:

Map of streamItems

## getStreamTypeNames

```
public java.util.List getStreamTypeNames()
```

Get list of streamType names. Returns shallow copy of list.

**Returns:**

list of streamType names

---

## getStreamDef

```
public StreamItem getStreamDef(String name)
```

Get streamItem by streamType name.

**Parameters:**

name - streamType name

**Returns:**

streamItem

## com.wowza.wms.vhost Class ThreadPool

java.lang.Object

└─com.wowza.wms.vhost.ThreadPool

public class **ThreadPool**  
extends Object

ThreadPool: class for managing a pool of threads.

### Constructor Summary

public	<a href="#">ThreadPool</a> ( <a href="#">IVHost</a> vhost, String name) Create a new thread pool attached to a vHost.
--------	--

### Method Summary

void	<a href="#">execute</a> (Runnable command) Execute a runnable object.
int	<a href="#">getActiveCount</a> () Get number of active threads.
java.util.concurrent. Executor	<a href="#">getExecutor</a> () Get the underlying Executor pool.
int	<a href="#">getQueueSize</a> () Get the number of command objects in the LinkedBlockingQueue.
void	<a href="#">init</a> (int corePoolSize) Initialize threadPool.
void	<a href="#">terminate</a> () Terminate all threads and cleanup threadPool.

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructors

### ThreadPool

public **ThreadPool**([IVHost](#) vhost,  
[String](#) name)

Create a new thread pool attached to a vHost.

#### Parameters:



(continued from last page)

vhost - vHost

## Methods

### init

```
public void init(int corePoolSize)
```

Initialize threadPool.

**Parameters:**

corePoolSize - core pool size

### terminate

```
public void terminate()
```

Terminate all threads and cleanup threadPool.

### execute

```
public void execute(Runnable command)
```

Execute a runnable object. If the threadPool is at the corePool size, the object will be added to a LinkedBlockingQueue in the order it was received.

**Parameters:**

command

### getExecutor

```
public java.util.concurrent.Executor getExecutor()
```

Get the underlying Executor pool.

**Returns:**

Executor pool

### getActiveCount

```
public int getActiveCount()
```

Get number of active threads.

**Returns:**

number of active threads

### getQueueSize

```
public int getQueueSize()
```

Get the number of command objects in the LinkedBlockingQueue.

**Returns:**

number of command objects in the LinkedBlockingQueue

## com.wowza.wms.vhost Class VHostItem

java.lang.Object

└─com.wowza.wms.vhost.VHostItem

public class **VHostItem**  
extends Object

VHostItem: data class that hold definition of a virtual host.

### Constructor Summary

public	<a href="#">VHostItem()</a> Create empty vHostItem
--------	---

### Method Summary

String	<a href="#">getConfigDir()</a> Get configuration path for vHost.
int	<a href="#">getConnectionLimit()</a> Get connection limit of this vHost item.
String	<a href="#">getName()</a> Get vHost name
<a href="#">WMSProperties</a>	<a href="#">getProperties()</a> Get properties
boolean	<a href="#">isVisited()</a> Has this vHostItem been visited during load of VHosts.xml file.
void	<a href="#">reset()</a> Reset vHostItem to empty state
void	<a href="#">setConfigDir(String configDir)</a> Set configuration path for vHost.
void	<a href="#">setConnectionLimit(int connectionLimit)</a> Set connection limit of this vHost item.
void	<a href="#">setName(String name)</a> Set vHost name
void	<a href="#">setProperties(WMSProperties properties)</a> Set properties
void	<a href="#">setVisited(boolean visited)</a> Set is visited.
String	<a href="#">toString()</a> Format vHost item

**Methods inherited from class** `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

## Constructors

### VHostItem

```
public VHostItem()
```

Create empty vHostItem

## Methods

### reset

```
public void reset()
```

Reset vHostItem to empty state

### getConfigDir

```
public String getConfigDir()
```

Get configuration path for vHost.

**Returns:**

configuration path for vHost

### setConfigDir

```
public void setConfigDir(String configDir)
```

Set configuration path for vHost.

**Parameters:**

configDir - configuration path for vHost

### getName

```
public String getName()
```

Get vHost name

**Returns:**

vHost name

### setName

```
public void setName(String name)
```

Set vHost name

**Parameters:**

(continued from last page)

name - vHost name

---

## getProperties

```
public WMSProperties getProperties()
```

Get properties

**Returns:**

properties

---

## setProperties

```
public void setProperties(WMSProperties properties)
```

Set properties

**Parameters:**

properties - properties

---

## isVisited

```
public boolean isVisited()
```

Has this vHostItem been visited during load of VHosts.xml file. Internally used to track and delete vHost definitions on reload of VHosts.xml.

**Returns:**

Has this vHostItem been visited during load

---

## setVisited

```
public void setVisited(boolean visited)
```

Set is visited.

**Parameters:**

visited - is visited

---

## getConnectionLimit

```
public int getConnectionLimit()
```

Get connection limit of this vHost item.

**Returns:**

connection limit of this vHost item

---

## setConnectionLimit

```
public void setConnectionLimit(int connectionLimit)
```

Set connection limit of this vHost item.

**Parameters:**

connectionLimit - connection limit of this vHost item

---

(continued from last page)

**toString**

```
public String toString()
```

Format vHost item

## com.wowza.wms.vhost Class VHostList

java.lang.Object

└─com.wowza.wms.vhost.VHostList

public class **VHostList**  
extends Object

VHostList: list of VHost items. Result of parsing VHosts.xml at server startup or VHosts.xml reload. This interface can keep track of reloads and carefully mark items for deletion.

### Constructor Summary

public	<a href="#">VHostList()</a> Create empty vHostList
--------	---

### Method Summary

void	<a href="#">addVHostItemListener()</a> ( <a href="#">IVHostItemNotify</a> vHostItemListener) 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()</a> ( <a href="#">VHostItem</a> vhostItem) Notify vHostItem listener of item create.
void	<a href="#">notifyVHostItemDestroy()</a> ( <a href="#">VHostItem</a> vhostItem) Notify vHostItem listener of item destroy.
void	<a href="#">notifyVHostItemUpdate()</a> ( <a href="#">VHostItem</a> vhostItem) Notify vHostItem listener of item update.
void	<a href="#">reloadConfig()</a> Reload VHosts.xml file.
void	<a href="#">removeVHostItemListener()</a> ( <a href="#">IVHostItemNotify</a> vHostItemListener) Remove vHostItem listener.

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

---

## Constructors

### VHostList

```
public VHostList()
```

Create empty vHostList

---

## Methods

### loadConfig

```
public void loadConfig()
```

---

### reloadConfig

```
public void reloadConfig()
```

Reload VHosts.xml file.

---

### getVHostMap

```
public java.util.Map getVHostMap()
```

Get the Map of vHostItem defintions.

**Returns:**

Map of vHostItem defintions

---

### addVHostItemListener

```
public void addVHostItemListener(IVHostItemNotify vHostItemListener)
```

Add vHostItem listener. vHostItem listeners will receive the following events: onVHostItemCreate, onVHostItemUpdate, onVHostItemDestroy.

**Parameters:**

vHostItemListener - vHostItem listener

---

### removeVHostItemListener

```
public void removeVHostItemListener(IVHostItemNotify vHostItemListener)
```

Remove vHostItem listener.

**Parameters:**

vHostItemListener - vHostItem listener

---

### notifyVHostItemCreate

```
public void notifyVHostItemCreate(VHostItem vhostItem)
```

Notify vHostItem listener of item create.

(continued from last page)

**Parameters:**

vhostItem - vHostItem

---

**notifyVHostItemUpdate**

```
public void notifyVHostItemUpdate(VHostItem vhostItem)
```

Notify vHostItem listener of item update.

**Parameters:**

vhostItem - vHostItem

---

**notifyVHostItemDestroy**

```
public void notifyVHostItemDestroy(VHostItem vhostItem)
```

Notify vHostItem listener of item destroy.

**Parameters:**

vhostItem - vHostItem

---

**getVHostNames**

```
public java.util.List getVHostNames()
```

Get a list of vHost names. Creates a copy of list.

**Returns:**

list of vHost names

---

**getVHostItems**

```
public java.util.List getVHostItems()
```

Get a list of vHostItems. Create a shallow copy of list.

**Returns:**

list of vHostItems



## com.wowza.wms.vhost Class VHostSingleton

java.lang.Object

└─com.wowza.wms.vhost.VHostSingleton

public class **VHostSingleton**  
extends Object

VHostSingleton: singleton that provide access to running vHosts.

### Constructor Summary

public	<a href="#">VHostSingleton()</a>
--------	----------------------------------

### Method Summary

static void	<a href="#">addVHostListener</a> ( <a href="#">IVHostNotify</a> vhostListener) Add a vHost listener.
static <a href="#">IVHost</a>	<a href="#">getInstance</a> (String vhostName) Get vHost by name.
static <a href="#">IVHost</a>	<a href="#">getInstance</a> (String vhostName, boolean doCreate) Get vHost by name.
static java.util.List	<a href="#">getVHostNames</a> () Get a list of vHost names.
static void	<a href="#">init</a> (String vhostName, String configHome) Initialize vhost by name.
static void	<a href="#">notifyVHostClientConnect</a> ( <a href="#">IVHost</a> vhost, <a href="#">IClient</a> inClient, <a href="#">RequestFunction</a> function, <a href="#">AMFDataList</a> params) Notify vhost client connect
static void	<a href="#">notifyVHostCreate</a> ( <a href="#">IVHost</a> vhost) Notify vHost listener of create.
static void	<a href="#">notifyVHostInit</a> ( <a href="#">IVHost</a> vhost) Notify vHost listener of ini.
static void	<a href="#">notifyVHostShutdownComplete</a> ( <a href="#">IVHost</a> vhost) Notify vHost listener of destruction.
static void	<a href="#">notifyVHostShutdownStart</a> ( <a href="#">IVHost</a> vhost) Notify vHost listener of destruction.
static void	<a href="#">remove</a> (String vhostName) Remove a vHost from list of running vHosts.
static void	<a href="#">removeApplicationListener</a> ( <a href="#">IVHostNotify</a> vhostListener) Remove vHost listener

static void	<a href="#">setServer</a> ( <a href="#">IServer</a> server) Set a reference to the current running server.
static void	<a href="#">shutdown</a> (String vhostName) Shutdown a vHost by name.

**Methods inherited from class** `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## Constructors

### VHostSingleton

```
public VHostSingleton()
```

## Methods

### getInstance

```
public static IVHost getInstance(String vhostName)
```

Get vHost by name. Do NOT create if does not exist.

**Parameters:**

vhostName - vHost name

**Returns:**

vHost

### getInstance

```
public static IVHost getInstance(String vhostName,  
    boolean doCreate)
```

Get vHost by name. Create if does not exist.

**Parameters:**

vhostName - vhost name

doCreate - if true, create if does not exist

**Returns:**

vHost

### init

```
public static void init(String vhostName,  
    String configHome)
```

Initialize vhost by name.

**Parameters:**

vhostName - vHost name

(continued from last page)

configHome - configuration path, path to VHost.xml

---

## setServer

```
public static void setServer(IServer server)
```

Set a reference to the current running server.

### Parameters:

server - server

---

## shutdown

```
public static void shutdown(String vhostName)
```

Shutdown a vHost by name.

### Parameters:

vhostName - vhost name

---

## getVHostNames

```
public static java.util.List getVHostNames()
```

Get a list of vHost names. Return copy of list.

### Returns:

vHost names

---

## remove

```
public static void remove(String vhostName)
```

Remove a vHost from list of running vHosts.

### Parameters:

vhostName - vhost name

---

## addVHostListener

```
public static void addVHostListener(IVHostNotify vhostListener)
```

Add a vHost listener. A vHost listener receives the following events: onVHostCreate, onVHostInit, onVHostDestroy

### Parameters:

vhostListener - vHost listener

---

## removeApplicationListener

```
public static void removeApplicationListener(IVHostNotify vhostListener)
```

Remove vHost listener

### Parameters:

vhostListener - vHost listener

---

(continued from last page)

## notifyVHostClientConnect

```
public static void notifyVHostClientConnect(IVHost vhost,  
      IClient inClient,  
      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 381
- ABSOLUTE\_TIME\_PATTERN 381
- acceptConnection 275, 292
- ACCEPTORS\_ACCEPTOR 834
- ACCEPTORS\_HANDLERADAPTER 834
- acceptSession 356, 582
- acquire 463, 471, 472, 626, 650
- acquireMediaCaster 478
- actionListeners 581
- activateOptions 392
- add 72, 79, 120, 121, 122, 152, 153, 154, 561, 815
- addAcceptConnectionAttribute 306
- addAcceptorListener 840
- addActionListener 588
- addAfterBlock 53
- addApplicationInstanceListener 203
- addApplicationListener 839
- addAudioCodecConfigPacket 718
- addAudioData 704, 788, 789
- addBeforeBlock 53
- addBody 558
- addBouncyCastleSecurityProvider 85
- addBytes 558
- addClient 626, 650
- addClientListener 221, 710
- addConnectionListener 274
- addData 194, 363, 551
- addDataData 704, 789
- addDataEx 197
- addDifference 71, 79
- addHttpProvider 813
- addHTTPStreamerAdapterID 813
- addHTTPStreamerSession 245
- addIdleWorkerListener 846
- addIOPerformance 353, 586
- addIOPerformance2 353, 586
- addItem 778
- addListener 801
- addLiveStreamPacketizerListener 253
- addMediaCasterListener 225
- addMediaReaderListener 250
- addMediaStreamListener 223, 770
- addMediaWriterListener 248
- addMember 182
- addMetaDataListener 482
- addModuleListener 229
- addNameGroup 772
- addObject 136
- addPlayStreamByName 227
- addPublisher 243
- addQualityLevel 364
- addRTPIncomingDatagramPortAll 246
- addRTPIncomingDatagramPortRange 246
- addRTPSession 239
- addRTSPStream 582
- addServerListener 599, 614
- addSharedObjectListener 223, 638
- addSlotListener 630, 654
- addStartupStream 852
- addStreamDomainStr 358
- addStreamDomainStrs 358
- addStreamSrcToMediaCaster 473
- addString 136
- addToPlaylist 798, 799
- addTrait 136
- addVHostItemListener 871
- addVHostListener 875
- addVideoCodecConfigPacket 718
- addVideoData 703, 788
- adjustDataTimecode 67
- adjustFirstPacketTCs 43
- allocateBuffer 61, 64
- AMF3Utils 97
- AMF\_LEVEL0 108
- AMF\_LEVEL3 108
- AMFData 109
- AMFDataArray 118, 119
- AMFDataByteArray 129, 130
- AMFDataContextDeserialize 135
- AMFDataContextSerialize 137, 138
- AMFDataItem 143, 144
- AMFDataList 151, 152
- AMFDataMixedArray 163, 164
- AMFDataObj 171, 172
- AMFDataTrait 181
- AMFObj 186

AMFPacket 192, 193  
AMFUtils 11  
append 393  
appendParamsToUrl 89  
appInstance 541, 579, 764  
appInstanceShutdown 579  
APPLICATION\_KEY 380  
applicationExists 847  
assembleQueryStr 49  
audioCodecToString 37  
AUDIOSAMPLE\_ACCESS\_ALL 288  
AUDIOSAMPLE\_ACCESS\_NONE 288  
AUDIOSAMPLEACCESS 687  
authenticateHandler 320  
authenticateHTTPProvider 267  
authenticateHTTPProviderHandler 320  
authenticatePlayHandler 579  
authenticatePublishHandler 579  
authenticateRTSP 268  
AuthenticateUsernamePasswordProviderBase 263  
authenticationMethod 321  
AUTHMETHOD\_PLAY 578  
AUTHMETHOD\_PUBLISH 578  
AUTHMETHOD\_UNKNOWN 577

## B

badStreams 542  
binarySearch 61, 64  
booleanValue 145  
broadcastPlayMessage 770  
broadcastGetObjectEncoding 770  
broadcastMsg 226  
BufferUtils 14  
byteArrayToInt 17, 18  
byteArrayToLong 16, 17  
byteArrayToShort 18, 19  
byteArrayToString 15  
byteValue 145

## C

calcTotalPacketSize 197  
call 294, 295  
CALLBACK\_PARAM1 505

CALLBACK\_PARAM10 506  
CALLBACK\_PARAM2 505  
CALLBACK\_PARAM3 505  
CALLBACK\_PARAM4 505  
CALLBACK\_PARAM5 505  
CALLBACK\_PARAM6 505  
CALLBACK\_PARAM7 505  
CALLBACK\_PARAM8 506  
CALLBACK\_PARAM9 506  
canHandle 321, 327, 341  
CAT\_ALL 430  
CAT\_application 421  
CAT\_cupertino 421  
CAT\_rtsp 421  
CAT\_sanjose 421  
CAT\_server 420  
CAT\_session 421  
CAT\_smoothstreaming 421  
CAT\_stream 421  
CAT\_vhost 420  
CAT\_webm 421  
checkAndSetPlayLogged 353  
cleanUp 392  
clear 71, 562, 631, 649, 715, 857  
clearFastPlaySettings 297, 706  
clearIntData 135  
clearLoggingValues 354, 588, 715  
clearProperty 452, 860  
clearStreamName 767  
clearStreamSrcToMediaCaster 473  
client 263  
clone 71, 182, 193, 663  
cloneProperties 258  
close 7, 392, 626, 655, 669, 706, 743, 774, 790, 792, 794, 800  
closeHostPort 849  
closeStream 525  
CODEC\_AUDIO\_AAC 831  
CODEC\_AUDIO\_G711\_ALAW 830  
CODEC\_AUDIO\_G711\_MULAW 831  
CODEC\_AUDIO\_MP3 830  
CODEC\_AUDIO\_MP3\_8 831  
CODEC\_AUDIO\_NELLYMOSER 830  
CODEC\_AUDIO\_NELLYMOSER\_16MONO 830  
CODEC\_AUDIO\_NELLYMOSER\_8MONO 830  
CODEC\_AUDIO\_PCM\_BE 830

CODEC\_AUDIO\_PCM\_LE 830  
CODEC\_AUDIO\_PCM\_SWF 830  
CODEC\_AUDIO\_RESERVED 831  
CODEC\_AUDIO\_SPEEX 831  
CODEC\_AUDIO\_UNKNOWN 829  
CODEC\_AUDIO\_VORBIS 831  
CODEC\_STREAM\_MP2T 831  
CODEC\_STREAM\_UNKNOWN 831  
CODEC\_VIDEO\_H264 829  
CODEC\_VIDEO\_SCREEN 829  
CODEC\_VIDEO\_SCREEN2 829  
CODEC\_VIDEO\_SPARK 829  
CODEC\_VIDEO\_UNKNOWN 828  
CODEC\_VIDEO\_VP6 829  
CODEC\_VIDEO\_VP6A 829  
CODEC\_VIDEO\_VP8 829  
CODES\_HREF 381  
compress 132  
CONFIGURATOR\_CLASS\_KEY 380  
configureSocketAcceptor 813  
ConnectionCounter 274  
connectionHolder 580  
containsHeader 334  
containsHTTPStreamer 242  
containsKey 172  
containsLiveStreamPacketizer 242  
containsProperty 631, 651  
containsSlot 632, 651  
containsStreamDomainStr 358  
containsStreamNameParts 357  
CONTENTTYPE\_AUDIO 832  
CONTENTTYPE\_BUFFERSIZE 832  
CONTENTTYPE\_DATA 833  
CONTENTTYPE\_DATA0 833  
CONTENTTYPE\_DATA3 833  
CONTENTTYPE\_FUNCTION 833  
CONTENTTYPE\_FUNCTION0 833  
CONTENTTYPE\_FUNCTION3 834  
CONTENTTYPE\_MEDIACHUNK 834  
CONTENTTYPE\_PLAYCALLBACK 834  
CONTENTTYPE\_SETCRUNKSIZE 832  
CONTENTTYPE\_SHAREDOBJECTS 833  
CONTENTTYPE\_SHAREDOBJECTS0 833  
CONTENTTYPE\_SHAREDOBJECTS3 833  
CONTENTTYPE\_UNKNOWN 832

CONTENTTYPE\_VIDEO 832  
CONTENTTYPE\_WATCHDOG 832  
convertParams 11  
COOKIEDATEFORMAT 48  
cookieStr 578  
copyFile 29  
copyFile2 29  
COUNTER\_HTTPCUPERTINO 828  
COUNTER\_HTTPSANJOSE 828  
COUNTER\_HTTPSMOOTH 828  
COUNTER\_HTTPWEBM 828  
COUNTER\_RTMP 828  
COUNTER\_RTP 828  
COUNTER\_TOTAL 828  
createApplication 852  
createBroadcastMessage 557  
createConnectMessage 557  
createContextDeserialize 112  
createContextSerialize 112  
createDefaultMessage 557  
createEnhancedSeekMessage 558  
createInstance 68, 786, 797  
createPlayStatusMessage 557  
createSeekMessage 557  
createSOMessage 558  
createStream 523  
CTRL\_playlist\_node 425

## D

DATA\_TYPE\_AMF3 106  
DATA\_TYPE\_AMF3\_ARRAY 108  
DATA\_TYPE\_AMF3\_BOOLEAN\_FALSE 107  
DATA\_TYPE\_AMF3\_BOOLEAN\_TRUE 107  
DATA\_TYPE\_AMF3\_BYTEARRAY 108  
DATA\_TYPE\_AMF3\_DATE 108  
DATA\_TYPE\_AMF3\_INTEGER 107  
DATA\_TYPE\_AMF3\_NULL 107  
DATA\_TYPE\_AMF3\_NUMBER 107  
DATA\_TYPE\_AMF3\_OBJECT 108  
DATA\_TYPE\_AMF3\_STRING 107  
DATA\_TYPE\_AMF3\_UNDEFINED 107  
DATA\_TYPE\_AMF3\_XML\_LEGACY 108  
DATA\_TYPE\_AMF3\_XML\_TOP 108  
DATA\_TYPE\_ARRAY 105

DATA\_TYPE\_AS\_OBJECT 106  
DATA\_TYPE\_BOOLEAN 104  
DATA\_TYPE\_BYTEARRAY 106  
DATA\_TYPE\_CUSTOM\_CLASS 106  
DATA\_TYPE\_DATE 105  
DATA\_TYPE\_INTEGER 106  
DATA\_TYPE\_LONG\_STRING 106  
DATA\_TYPE\_MIXED\_ARRAY 105  
DATA\_TYPE\_MOVIE\_CLIP 105  
DATA\_TYPE\_NULL 105  
DATA\_TYPE\_NUMBER 104  
DATA\_TYPE\_OBJECT 105  
DATA\_TYPE\_OBJECT\_END 105  
DATA\_TYPE\_RECORDSET 106  
DATA\_TYPE\_REFERENCE\_OBJECT 105  
DATA\_TYPE\_STRING 104  
DATA\_TYPE\_UNDEFINED 105  
DATA\_TYPE\_UNKNOWN 104  
DATA\_TYPE\_XML 106  
DATA\_TYPE\_XML\_TOP 107  
datas 369  
DATE\_AND\_TIME\_FORMAT 381  
DATE\_AND\_TIME\_PATTERN 381  
dateValue 145  
debug 401, 402  
debugRTSPSession 581  
DebugUtils 22  
DECODE\_OBJ\_REF 170  
DECODE\_TRAITS 171  
DECODE\_TRAITS\_EXT 170  
DECODE\_TRAITS\_REF 170  
DECODE\_UNDEFINED 170  
decodeHexString 15  
decodeRangeHeader 805  
decodeStorageDir 241, 611  
decodeStreamInfo 806  
decodeValue 89  
decompress 133  
decrement 276  
DEFAULT\_APPINSTANCE\_NAME 218  
DEFAULT\_APPLICATION\_NAME 201  
DEFAULT\_CONFIGURATION\_FILE 380  
DEFAULT\_CONFIGURATION\_KEY 380  
DEFAULT\_PORT 392  
DEFAULT\_RANDOMACCESSREADER 667  
DEFAULT\_REPOSITORY\_NAME 379  
DEFAULT\_XML\_CONFIGURATION\_FILE 380  
defaultLocale 84  
defaultTimeZone 84  
deleteDirectory 29  
deleteSlot 627, 651  
deleteStream 524  
deserialize 114, 126, 132, 146, 158, 164, 179  
deserializeDate 98  
deserializeInnerObject 112  
deserializeInt 98  
deserializeString 98  
destroyInstance 68  
disconnect 276, 629, 638, 654  
doBreak 555  
doCRC32 21  
doHTTPAuthentication 322  
doSet 72, 79  
doubleValue 145  
doWatchdog 439, 463, 470, 612  
dummy 77  
  
E  
  
elapsedTime 581  
ElapsedTimer 26  
encodeHexString 15  
encodeValue 89  
encrypt 56  
end 86  
error 400  
EVT\_ALL 430  
EVT\_announce 424  
EVT\_app\_start 424  
EVT\_app\_stop 424  
EVT\_comment 424  
EVT\_connect 422  
EVT\_connect\_burst 422  
EVT\_connect\_pending 422  
EVT\_create 423  
EVT\_describe 425  
EVT\_destroy 423  
EVT\_disconnect 422  
EVT\_pause 422  
EVT\_play 422



- EVT\_publish 422
- EVT\_record 423
- EVT\_recordstop 423
- EVT\_seek 423
- EVT\_server\_start 424
- EVT\_server\_stop 424
- EVT\_setbuffertime 422
- EVT\_setstreamtype 423
- EVT\_stop 423
- EVT\_unpause 423
- EVT\_unpublish 422
- EVT\_vhost\_start 424
- EVT\_vhost\_stop 424
- execute 865
- exists 8, 637
- expandEnvironmentVariables 84
  
- F
  
- FastPlaySettings 659
- fatal 401
- FCPublish 529
- FCSubscribe 530
- fcSubscribe 311
- FCUnPublish 529
- FCUnpublish 529
- FCUnSubscribe 531
- FCUnsubscribe 530
- fcUnSubscribe 312
- fcUnSubscribeAll 312
- FD\_ALL 430
- FD\_c\_client\_id 417
- FD\_c\_ip 417
- FD\_c\_proto 417
- FD\_c\_referrer 417
- FD\_c\_user\_agent 417
- FD\_cs\_bytes 418
- FD\_cs\_bytes\_inc 386
- FD\_cs\_stream\_bytes 418
- FD\_cs\_stream\_bytes\_inc 386
- FD\_cs\_uri\_query 419
- FD\_cs\_uri\_stem 419
- FD\_date 416
- FD\_s\_ip 419
- FD\_s\_port 420
- FD\_s\_uri 417
- FD\_sc\_bytes 418
- FD\_sc\_bytes\_inc 386
- FD\_sc\_stream\_bytes 418
- FD\_sc\_stream\_bytes\_inc 386
- FD\_time 416
- FD\_tz 416
- FD\_x\_app 417
- FD\_x\_appinst 417
- FD\_x\_category 416
- FD\_x\_comment 420
- FD\_x\_ctx 416
- FD\_x\_ctx\_override 416
- FD\_x\_duration 419
- FD\_x\_duration\_inc 386
- FD\_x\_event 416
- FD\_x\_file\_ext 419
- FD\_x\_file\_length 418
- FD\_x\_file\_name 419
- FD\_x\_file\_size 418
- FD\_x\_severity 420
- FD\_x\_sname 418
- FD\_x\_sname\_query 419
- FD\_x\_spos 418
- FD\_x\_status 419
- FD\_x\_stream\_id 420
- FD\_x\_suri 420
- FD\_x\_suri\_query 419
- FD\_x\_suri\_stem 420
- FD\_x\_vhost 416
- FILEEXTENSION 623
- FILEFORMAT\_FLV 832
- FILEFORMAT\_MP4 832
- FILEFORMAT\_UNKNOWN 831
- FileUtils 28
- filters 320
- floatValue 145
- flush 627, 638, 655, 718, 790
- FLV\_CHUNKHEADER\_BUFFERSIZE 35
- FLV\_CHUNKHEADER\_FIRSTBYTE 35
- FLV\_CHUNKHEADER\_HEADERSIZE 35
- FLV\_CHUNKHEADER\_ISIZE 35
- FLV\_CHUNKHEADER\_ITIMECODE 35
- FLV\_CHUNKHEADER\_ITYPE 34
- FLV\_CHUNKHEADER\_SECONDBYTE 35

FLV\_CHUNKHEADER\_VALUESIZE 35  
FLV\_DFRAME 36  
FLV\_KFRAME 35  
FLV\_PFRAME 36  
FLV\_TCINDEXAUDIO 36  
FLV\_TCINDEXDATA 36  
FLV\_TCINDEXVIDEO 36  
FLV\_UFRAME 35  
FLVUtils 36  
forceNewLoggerInstance 405  
forceReset 439  
formatBytes 23  
formatBytesShort 23  
formatBytesStruct 24  
formatDeleteCookie 50  
formatH264CodecConfig 807  
formatH264CodecConfigPacket 806  
formatSetCookie 51  
FORWARD 6  
frameTypeToString 37

## G

generateIndex 61, 64  
get 60, 63, 123, 156, 174, 636, 816  
getAbsTimecode 188, 195  
getAccess 656, 716  
getActiveCount 865  
getAdapterName 343  
getAddress 811  
getAddressRawStr 811  
getAddressStr 811  
getAdminAgent 610  
getAdminInterfaceObjectList 599, 614  
getAfterBlocks 53  
getAllAsStrings 260  
getAllowDomains 232  
getAppInstance 202, 289, 344, 353, 437, 512, 584, 769, 787  
getAppInstanceName 769  
getAppInstanceNames 204  
getAppInstanceProperty 546  
getApplication 218, 289, 393, 512, 836  
getApplicationFolderNames 847  
getApplicationInstance 470  
getApplicationInstanceTouchTimeout 250

getApplicationLock 844  
getApplicationNames 846  
getApplicationPath 201  
getApplicationProperty 545  
getApplicationTimeout 229, 842  
getAppName 769  
getAttributes 366, 371  
getAudioCodec 38, 44  
getAudioCodecConfigPacket 717  
getAudioMissing 703  
getAudioSize 689  
getAudioTC 691  
getAuthenticatePlayHandler 584  
getAuthenticatePublishHandler 584  
getAuthenticationList 848  
getAuthenticationMethod 322, 327  
getBaseClass 453, 859  
getBasePath 8  
getBeforeBlocks 53  
getBitrate 370  
getBlockSize 56  
getBoolean 125, 158, 176, 178  
getBuffers 56  
getBufferTime 299, 688  
getBurstStartStop 714  
getByte 124, 158, 176, 178  
getByteAllocation 52  
getByteContainerLevel 189  
getCacheName 712  
getCallbackParamCount 508  
getChecksum 663  
getClassName 180, 182  
getClient 221, 264, 270, 694, 835  
getClientById 220  
getClientCount 220, 839  
getClientCountTotal 220  
getClientID 526  
getClientId 288, 694  
getClientIdGenerator 601, 615  
getClientIdGeneratorRecycleDelaySize 618  
getClientIdGeneratorRecycleSize 617  
getClientIdGeneratorTimeout 617  
getClientIdleFrequency 232, 844  
getClientProperty 546  
getClients 221, 625, 650

getClientsLockObj 239  
getClientTimeout 835  
getClientUpdates 653  
getCodecPrivateData 371  
getCommandInterface 613  
getCommandInterfaceHostPort 598, 613  
getCommittedVirtualMemory 617  
getConfigDir 867  
getConfigPath 202  
getConfiguation 812  
getConnectionCounter 203, 225, 597, 598, 613, 840, 841  
getConnectionHolder 353, 586  
getConnectionLimit 841, 868  
getConnectionTimeout 453  
getConnectionValidator 613  
getConnectLastAttempt 441  
getConnectLastForceReset 441  
getConnectLastSuccess 441  
getConnectTime 297  
getContentLength 330  
getContentType 330  
getContextStr 248, 722  
getCookieStr 356, 590  
getCoreHandlerPoolSize 600, 615, 837  
getCoreTransportPoolSize 599, 614, 837  
getCount 768  
getCryptoPoolActiveCount 616  
getCryptoPoolMaxSize 616  
getCurrent 276  
getCurrentHeapSize 617  
getCurrentItem 798  
getData 196, 550, 641  
getDataBuffer 196  
getDataByteBuffer 550  
getDataMissing 703  
getDdatas 363  
getDataSetSize 364, 689  
getDataSetStr 368  
getDataSetTC 692  
getDataSetTrack 373  
getDataSetTrackCount 373  
getDataSetTracks 373  
getDataSetType 693  
getDate 26, 125, 158, 176, 178  
getDateStarted 204, 225, 298, 598, 613, 841  
getDateString 27  
getDescription 453, 861  
getDirection 660  
getDirecton 7  
getDouble 125, 157, 176, 178  
getDuration 672, 758, 792, 794  
getDynamicLogProperties 601, 615, 849  
getElapsedTime 311, 359, 720  
getEncoding 393  
getExecutor 865  
getExt 715  
getExtent 662  
getExtraData 552  
getFastPlaySettings 296, 706  
getFileExtension 787  
getFileInBytes 75  
getFileInBytesRate 77  
getFileInfo 358  
getFileIOPoolSize 837  
getFileOutBytes 76  
getFileOutBytesRate 77  
getFilePointer 7  
getFirstByte 197  
getFlashVer 289  
getFloat 125, 157, 176, 178  
getFourCC 371  
getFps 659  
getFrameType 37, 38  
getGlobalLogValue 407  
getGUID 616  
getHandlerThreadPool 600, 615, 838  
getHeader 329, 335  
getHeaderNames 329  
getHeaders 335  
getHeaderSize 336, 552, 717  
getHomePath 836  
getHostPortsList 838  
getHttpProviders 813  
getHTTPStreamerAdapter 353, 851  
getHTTPStreamerAdapterIDs 813, 851  
getHTTPStreamerApplicationContext 245  
getHTTPStreamerContext 847  
getHTTPStreamerItem 342  
getHTTPStreamerList 241, 851  
getHTTPStreamerProperties 240

getHTTPStreamerSession 720  
getHTTPStreamerSessionCount 244, 245  
getHTTPStreamerSessionCountsByName 244  
getHTTPStreamerSessions 243, 244, 245  
getId 343  
getId 186  
getIdleCheckFrequency 845  
getIdleFrequency 302, 342, 589  
getIdleHandler 586  
getIdleMinimumWaitTime 848  
getIdleTimeout 438  
getIdleWorkerCount 845  
getIdleWorkers 848  
getIdString 454  
getIndex 780  
getInnerObj 183  
getInputStream 332  
getInstance 406, 610, 874  
getInt 124, 157, 175, 177  
getIntData 135  
getIntHeader 329, 335  
getIOPerformanceCounter 227, 358, 591  
getIoPerformanceCounter 204, 597, 612, 840  
getIOScheduler 840  
getIp 296, 585  
getIpAddress 354  
getJmxRemoteConfig 614  
getKeepAliveTime 454  
getKeepAliveTimeout 845  
getKey 174  
getKeys 174  
getLastAccessed 663  
getLastAuthenticateMethod 584  
getLastClientId 643  
getLastConnectAcceptedStamp 277  
getLastConnectAcceptedStampString 277  
getLastConnectAcceptedTime 277  
getLastConnectAcceptedTimeString 277  
getLastConnectRejectedByReasonStamp 278  
getLastConnectRejectedByReasonStampString 278  
getLastConnectRejectedByReasonTime 278  
getLastConnectRejectedByReasonTimeString 278  
getLastConnectRejectedStamp 277  
getLastConnectRejectedStampString 277  
getLastConnectRejectedTime 277  
getLastConnectRejectedTimeString 278  
getLastDisconnectStamp 279  
getLastDisconnectStampString 279  
getLastDisconnectTime 279  
getLastDisconnectTimeString 279  
getLastKeyFrame 704  
getLastOperation 643  
getLastPacket 705  
getLastStreamId 524  
getLastTC 44  
getLastTouchTime 249  
getLastValidateTime 307  
getLength 672, 780  
getLiveRepeaterCapabilities 311  
getLiveStreamingPacketizer 354  
getLiveStreamPacketizer 464, 527, 719, 720, 772  
getLiveStreamPacketizerControl 246  
getLiveStreamPacketizerList 241, 310, 719, 851  
getLiveStreamPacketizerLock 771  
getLiveStreamPacketizerProperties 240  
getLiveStreamRepeater 464, 720  
getLiveThreads 616  
getLocale 333  
getLock 351, 461, 469, 797  
getLockCount 463, 477  
getLogger 397, 407, 514  
getLoggerObj 405, 407  
getLong 124, 157, 175, 177  
getMACAddress 85  
getMaxHeapSize 617  
getMaximumPendingWriteBytes 230, 310, 843  
getMaximumSetBufferTime 231, 310, 843  
getMaxStorageDirDepth 240  
getMaxTimecode 704, 790  
getMediaCaster 462, 471, 730  
getMediaCasterCount 470  
getMediaCasterDef 437, 457  
getMediaCasterDefs 456  
getMediaCasterHostPortConfig 458  
getMediaCasterId 437, 462  
getMediaCasterList 846  
getMediaCasterNames 456, 470  
getMediaCasterProcessorCount 459  
getMediaCasterProperties 240  
getMediaCasterSettings 846

getMediaCasterStreamItem 439  
getMediaCasterStreams 228  
getMediaCasterType 442  
getMediaCasterValidator 249  
getMediaExtension 8, 672  
getMediaIOPerformance 707  
getMediaIOPerformanceCounter 289  
getMediaName 8  
getMediaReaderProperties 240  
getMediaReaders 841  
getMediaWriterProperties 240  
getMediaWriters 842  
getMember 182  
getMemberCount 182  
getMembers 182  
getMessage 551  
getMessagesInBytes 75  
getMessagesInBytesRate 76  
getMessagesInCount 75  
getMessagesInCountRate 76  
getMessagesLossBytes 76  
getMessagesLossBytesRate 77  
getMessagesLossCount 75  
getMessagesLossCountRate 76  
getMessagesOutBytes 75  
getMessagesOutBytesRate 77  
getMessagesOutCount 75  
getMessagesOutCountRate 76  
getMetadata 669  
getMetadataPacket 565  
getMetaDataProvider 717  
getMethod 330  
getMissing 193, 550  
getModFunctions 228  
getModuleInstance 229  
getModuleList 229  
getMp3TagMap 842  
getMultiplier 659  
getName 202, 218, 364, 388, 452, 629, 642, 654, 690, 778, 780, 800, 838, 859, 867  
getNameGroups 772  
getNameGroupStreamNames 772  
getNetConnection 444, 694  
getNetConnectionHostPortConfig 846  
getNetConnectionIdleFrequency 844  
getNetConnectionProcessorCount 839  
getNextNetConnectionId 835  
getNextStreamIndex 767, 768  
getNodeByTagName 92  
getNodeValue 92  
getNodeValueByTagName 92  
getObject 125, 136, 158, 177, 179  
getObjectEncoding 135, 138, 190, 309  
getObjectNames 636  
getObjectReference 138  
getOrCreate 636  
getOutputStream 334  
getPageUrl 301, 530  
getParam 508  
getParamBoolean 511, 512  
getParamCount 508  
getParamDate 510  
getParamDouble 510, 511  
getParameter 331  
getParameterMap 331  
getParameterNames 331  
getParameterValues 331  
getParamInt 510  
getParamLong 511  
getParamMixedArray 509  
getParamObj 509  
getParamString 509  
getParamType 508  
getParamValue 90  
getParent 632, 656, 743  
getParentStreamIndex 365  
getPassword 269  
getPath 9, 321, 672  
getPeakThreads 616  
getPingRoundTripTime 298  
getPingTimeout 230, 307, 842  
getPlayClass 860  
getPlayer 703  
getPlayerCount 463, 477  
getPlaylist 798  
getPlayPackets 704  
getPlayStreamCount 228  
getPlayStreamCountsByName 228  
getPlayStreams 289  
getPlayStreamsByName 228

getPollingInterval 801  
getPort 389, 394, 811  
getProcessorCount 812  
getProperties 203, 219, 291, 342, 344, 453, 585, 600, 615, 704, 837, 860, 868  
getProperty 259, 452, 628, 651, 835, 860  
getPropertyBoolean 259  
getPropertyDouble 260  
getPropertyInt 259  
getPropertyLong 260  
getPropertyStr 259  
getProtocol 302, 332  
getPublishAudioCodecId 721  
getPublisher 801  
getPublisherCount 243  
getPublishers 243  
getPublishStreamNames 248, 765  
getPublishStreams 290  
getPublishVideoCodecId 721  
getQualityLevels 364  
getQueryStr 300, 356, 587, 714  
getQueryString 330  
getQueueSize 865  
getReceiveVideoFPS 707  
getReconnectWaitTime 441  
getRefCount 626, 650  
getReference 111  
getReferrer 301, 355, 530, 587  
getRemoteAddr 332  
getRemoteHost 332, 393  
getRepeat 777, 800  
getRepeaterOriginUrl 231, 307, 529  
getRepeaterQueryString 231  
getRequestFilters 321, 327  
getRequestURI 330  
getRequestURL 330  
getRespAMFAudioObj 300, 714  
getRespAMFDataObj 300, 714  
getRespAMFVideoObj 300, 714  
getRespFunctions 297  
getResponseAMFObj 300  
getRsoStorageDir 233  
getRsoStoragePath 234  
getRTPAVSyncMethod 237  
getRTPContext 847  
getRTPDatagramConfigIncoming 847  
getRTPDatagramConfigOutgoing 847  
getRTPIdleFrequency 232, 844  
getRTPMaxRTCPWaitTime 238  
getRTPPlayAuthenticationMethod 237  
getRTPProperties 240  
getRTPPublishAuthenticationMethod 237  
getRTPSession 263, 270, 571  
getRTPSessionCount 238, 239  
getRTPSessionCountsByName 238  
getRTPSessions 238  
getRTPStream 309, 718  
getRTPWriteListener 586  
getRTSPBindIpAddress 252  
getRTSPConnectionAddressType 252  
getRTSPConnectionIpAddress 252  
getRTSPMaximumPendingWriteBytes 250  
getRTSPOriginAddressType 253  
getRTSPOriginIpAddress 253  
getRTSPPlayRangeStart 590  
getRTSPPlayRangeStop 590  
getRTSPSessionTimeout 250  
getRTSPStream 583  
getRTSPTunnelingSessionId 590  
getScheme 332  
getSDPData 570  
getSecondByte 197  
getSeq 196  
getServerHostPort 302  
getServerIp 355, 587  
getServerName 332  
getServerPort 332, 355, 587  
getSessionGUID 616  
getSessionId 351, 582  
getSessionProtocol 354  
getSessionTimeout 352  
getSessionType 354  
getSharedObjectReadAccess 236, 303  
getSharedObjects 221  
getSharedObjectWriteAccess 236, 303  
getShort 124, 157, 175, 177  
getSize 186, 193, 550  
getSlot 625, 649  
getSlotNames 625, 649  
getSlots 631, 649

getSlotVersion 642  
getSoVersion 642  
getSrc 187, 194, 551, 689  
getSSLConfig 813  
getSslFactoryClass 811  
getStart 780  
getStartTC 660  
getStartTCOffset 660  
getStartupStreams 852  
getStorageDir 630, 638, 655  
getStream 352, 437, 514, 765, 766, 767, 786  
getStreamArray 466  
getStreamAudioSampleAccess 234, 304  
getStreamBitrate 528  
getStreamClientless 765  
getStreamCount 228  
getStreamDef 863  
getStreamDefs 862  
getStreamExt 357, 464  
getStreamFile 298, 299  
getStreamFileForRead 713  
getStreamFileForWrite 712, 713  
getStreamFileMapper 241  
getStreamIsRunningLock 441  
getStreamKeyDir 233  
getStreamKeyPath 234  
getStreamLastSeq 440  
getStreamLength 528  
getStreamList 467  
getStreamListLock 764  
getStreamManager 470  
getStreamMissingTime 439  
getStreamName 357  
getStreamNameAliasProvider 242  
getStreamNameLock 765  
getStreamNameParts 357  
getStreamNames 477  
getStreamPosition 357, 672  
getStreamProperties 239  
getStreamProperty 547  
getStreamReadAccess 235, 305  
getStreams 219, 695, 765  
getStreamStorageDir 233  
getStreamStoragePath 234  
getStreamTimeout 439  
getStreamTimeoutLastReset 440  
getStreamTimeoutLastTime 440  
getStreamTimeoutReason 440  
getStreamType 219, 291, 451, 527, 705, 787  
getStreamTypeNames 863  
getStreamTypes 834  
getStreamVideoSampleAccess 234, 304  
getStreamWriteAccess 235, 306  
getString 124, 136, 156, 175, 177  
getStringReference 138  
getSubType 365  
getTargetEncoding 138  
getThreadPool 600, 615, 838  
getTime 27  
getTimecode 187, 195, 367, 551, 558  
getTimeRunning 204, 226, 298, 359, 598, 613, 841  
getTimeRunningSeconds 204, 226, 298, 359, 598, 613, 841  
getTimeScale 365  
getTimeSeconds 27  
getTimeString 27  
getTotal 279  
getTotalAccepted 279  
getTotalIOPerformanceCounter 289  
getTotalRejected 280  
getTrait 136, 180  
getTraitReference 139  
getTransportThreadPool 600, 615, 838  
getType 109, 156, 187, 194, 364, 551, 559  
getUDPPortManager 602, 616  
getUDPTransportManager 851  
getUniqueId 461  
getUniqueStreamIdStr 719  
getUnsignedShort 21  
getUri 301, 355, 587  
getUrlStr 365  
getUserAgent 355, 583  
getUserAgents 599, 614  
getValidationFrequency 230, 843  
getValue 114, 126, 131, 146, 160, 180  
getVersion 526, 596, 611, 627, 651  
getVHost 203, 219, 263, 269, 294, 342, 344, 351, 436, 513, 583, 769  
getVHostItems 872  
getVHostList 601, 616  
getVHostMap 871

getVHostNames 872, 875  
getVideoCodec 38, 45  
getVideoCodecConfigPacket 718  
getVideoMissing 703  
getVideoSize 689  
getVideoTC 692  
getWmsNumber 552  
getWriteListener 306  
getXMLPropertyBool 95  
getXMLPropertyDouble 94  
getXMLPropertyExists 94  
getXMLPropertyInt 94  
getXMLPropertyLong 94  
getXMLPropertyStr 93

## H

handleCallback 709  
handleMessage 564  
HOSTNAME\_KEY 380  
HostPort 810  
HostPortList 815  
HTTPRequestToByteArray 48, 49  
HTTPRequestToFile 48  
HTTPProvider2Base 321  
HTTPStreamerUtils 375  
HTTPUtils 48

## I

idle 694  
idleFrequency 581  
idleHandler 580  
idStringToName 454  
IFasterByteArrayOutputStream 55  
incAbsTimecode 188  
incByteContainerLevel 189  
incClientCountTotal 220  
incrementAccept 275  
incrementBytesIn 72, 80  
incrementBytesLoss 74  
incrementBytesOut 74, 79  
incrementFileIn 72  
incrementFileOut 73  
incrementMediaInBytes 708

incrementMediaLossBytes 708  
incrementMediaOutBytes 708  
incrementMessagesIn 72, 73, 79, 80  
incrementMessagesLoss 73, 74, 75  
incrementMessagesOut 73, 74, 79  
incrementReject 275  
incSlotVersion 642  
indexOffset 369  
info 399, 400  
init 6, 265, 321, 327, 342, 344, 436, 462, 641, 668, 688, 740, 834, 865, 874  
initContextLogging 406  
initializeLogging 406  
initLiveStreamRepeating 527, 721, 743  
initStream 524  
interleavePackets 40, 41  
interruptPlay 743  
intToByteArray 19, 20  
intValue 144  
invokePrevious 514, 515  
IOPerformanceCounter 71  
ioPerformanceCounter 580  
IOPerformanceCounterDebug 79  
ip 579  
isAcceptConnection 220, 291  
isAcceptSession 356  
isActive 352, 388, 393  
isAMF0 136, 138  
isAMF3 136, 138  
isAMF3Start 110  
isAnnounce 580, 585  
isAnnounceOrDescribe 586  
isAppend 702  
isAppInstanceLoaded 202  
isApplicationLoaded 836  
isArrayStart 110  
isAudio 196  
isAudioCodecConfig 44  
isByteArrayStart 111  
isByteContainerEmpty 189  
isByteContainerFull 189  
isCDATA 368  
isClient 625, 650  
isClustered 712  
isConnected 291, 580, 586



isDebugEnabled 398  
isDebugLog 77  
isDebugRTSPSession 590  
isDescribe 580, 585  
isDynamic 182  
isDynamicLogContextLoaded 601, 616  
isEnabledFor 398  
isEncrypted 302  
isErrorEnabled 398  
isFileInfo 358  
isFlashMediaLiveEncoder 308  
isFlashVersion10 308  
isFlashVersion90115 308  
isFlashVersionH264Capable 308  
isForceAMF0 559  
isGlobalLogValueSet 407  
isHitEnd 664  
isInfoEnabled 398  
isIntData 135  
isLastSentAbsTimecode 190  
isLiveRepeater 307  
isLoggedConnect 585  
isLongTimecode 189  
isManifestOutput 365  
isMediaCasterPlay 722  
isMember 182  
isMixedArrayStart 111  
isNew 188  
ISO8601\_FORMAT 382  
ISO8601\_PATTERN 382  
isObjectEncodingAMF0 190, 309  
isObjectEncodingAMF3 190, 308  
isObjEnd 111  
isObjStart 110  
isOpen 7, 670, 706  
isPartial 551  
isPending 562  
isPersistent 626, 637, 650  
isPlay 693  
isPlaying 688  
isPlayLogged 353  
isPublishStreamReady 721  
isReceiveAudio 707  
isReceiveVideo 707  
isRecord 693  
isRTPIncomingDatagramPortValid 246  
isSecure 302, 333  
isSendPlayStopLogEvent 716  
isSendPublishStopLogEvent 716  
isSendRecordStopLogEvent 716  
isSendResult 512  
isSession 440  
isSessionValid 579, 584  
isShutdownClient 591  
isShutdownOnRelease 464  
isShuttingDown 848  
isSSL 302  
isStartStarupStreams 852  
isStream 440  
isStreamIsRunning 441  
isSuspended 602, 611, 813, 849  
isSwitchLog 801  
isTimeout 351  
isTimeoutSession 352  
isTraceEnabled 398  
isValid 462, 676  
isValidated 352  
isValidateFMLEConnections 254, 312  
isValidStreamDomainStr 358  
isVersionFile 757  
isVideo 196  
isVideoCodecConfig 44  
isVideoKeyFrame 44, 45  
isVisited 868  
isWaitForVideoKeyFrame 757  
isWarnEnabled 398

## J

JNDI\_CONTEXT\_NAME 381

## K

KEYFILEEXTENSION 375  
keyFileStrToBytes 376  
killClient 836  
killRTSPSession 836

## L

lastAuthenticateMethod 579  
lastModified 8  
length 9, 61, 64, 715, 743  
liveStreamPacketizers 764  
load 639, 655  
loadConfig 871  
loadConfigFile 806  
loadConfigProperties 93  
lock 632, 655  
lockRepeaterStreams 356  
log 398, 399  
LOG4J\_ID\_KEY 380  
LOG4J\_PACKAGE\_NAME 379  
logDebug 518  
logError 518  
loggedConnect 579  
LOGGERNAME\_SERVER 405  
logInfo 518  
logNotifier 609  
LogNotifyCalculateIncremental 386  
logWarn 518  
longToByteArray 20, 21  
longValue 144

**M**

main 610  
makeNewLoggerInstance 406  
MAXSTREAMINDEX 763  
MediaCasterItem 451  
MediaCasterList 456  
MediaCasterSettings 458  
MediaCasterStreamItem 461  
MediaCasterStreamManager 466  
MediaCasterStreamMap 469  
MEDIACASTERTYPE\_LIVEREPEATER 436  
MEDIACASTERTYPE\_RTPLIVE 436  
MEDIACASTERTYPE\_SHOUTCAST 436  
MEDIACASTERTYPE\_UNKNOWN 436  
mediaStreamListeners 764  
MediaStreamMap 764  
members 171  
MILLS\_PER\_HOUR 108  
MISSING 662  
ModuleBase 508  
ModuleClientLogging 518  
ModuleCore 523  
ModuleFastPlay 532  
ModuleMediaCaster 476  
ModuleMediaCasterStreamMonitorAdvanced 542  
ModuleProperties 545  
monitors 541  
msb0baseTime 84  
msb1baseTime 84  
MSG\_STARTSTREAM 345  
MSG\_SWITCHSTREAM 345

**N**

nameGroupId 763  
nameGroups 763  
newValue 86  
newXPathFactory 95  
next 800  
nextStreamId 764  
notifyActionOnMetaData 711  
notifyActionPause 711  
notifyActionPauseRaw 711  
notifyActionPlay 710  
notifyActionPublish 711  
notifyActionSeek 711  
notifyActionStop 712  
notifyActionUnPublish 711  
notifyLiveStreamPacketizerCreate 254  
notifyLiveStreamPacketizerDestroy 254  
notifyLiveStreamPacketizerInit 254  
notifyMediaReaderClose 251  
notifyMediaReaderCreate 251  
notifyMediaReaderExtractMetaData 251  
notifyMediaReaderInit 251  
notifyMediaReaderOpen 251  
notifyMediaStreamCreate 771  
notifyMediaStreamDestroy 771  
notifyMediaWriterOnFLVAddMetadata 249  
notifyMediaWriterOnWriteComplete 249  
notifyPlayPublish 770  
notifyPlayUnpublish 769, 770  
notifySlotDelete 655  
notifySlotSetValue 655  
notifyVHostClientConnect 875

notifyVHostCreate 876  
notifyVHostInit 876  
notifyVHostItemCreate 871  
notifyVHostItemDestroy 872  
notifyVHostItemUpdate 872  
notifyVHostShutdownComplete 876  
notifyVHostShutdownStart 876

## O

onAACEncodeInfo 483  
onAcceptConnection 315  
onAcceptorCreate 817  
onAcceptorDestroy 817  
onAnnounce 568, 588  
onApplicationCreate 256  
onApplicationDestroy 256  
onApplicationInstanceCreate 255  
onApplicationInstanceDestroy 255  
onAppStart 476, 487, 542  
onAppStop 476, 487, 542  
onBind 322, 323  
onCall 488  
onCallback 727  
onClientAccept 313  
onClientConnect 313  
onClientDisconnect 313  
onClientReject 314  
onCodecConfigAAC 480  
onConnect 489  
onConnectAccept 489  
onConnectFailure 448  
onConnectReject 490  
onConnectStart 447  
onConnectSuccess 447  
onCreate 338  
onData 443  
onDescribe 567, 588  
onDestroy 339  
onDisconnect 316, 489  
onFile 66  
onFillChunkEnd 339  
onFillChunkStart 339  
onFLVAddMetadata 759  
onFrameAAC 480  
onFrameMP3 480  
onGetParameter 568, 588  
onHeaderData 481, 483  
onHTTPCupertinoEncryptionKeyCreateLive 491  
onHTTPCupertinoEncryptionKeyCreateVOD 491  
onHTTPCupertinoEncryptionKeyRequest 491  
onHTTPCupertinoStreamingSessionCreate 492  
onHTTPCupertinoStreamingSessionDestroy 492  
onHTTPRequest 323  
onHTTPSanJoseStreamingSessionCreate 493  
onHTTPSanJoseStreamingSessionDestroy 493  
onHTTPSessionCreate 494  
onHTTPSessionDestroy 494  
onHTTPSmoothStreamingSessionCreate 495  
onHTTPSmoothStreamingSessionDestroy 495  
onHTTPStreamerSessionCreate 360  
onHTTPStreamerSessionDestroy 360  
onIndex 339  
onInit 338  
onLog 384, 387  
onMediaCasterCreate 445  
onMediaCasterDestroy 445  
onMediaReaderClose 674  
onMediaReaderCreate 674  
onMediaReaderExtractMetadata 674  
onMediaReaderInit 674  
onMediaReaderOpen 674  
onMediaStreamCreate 735  
onMediaStreamDestroy 735  
onMetadata 481, 483, 726  
onModuleLoad 486  
onModuleUnload 486  
onMP3EncodeInfo 483  
onNewVHost 613  
onOpen 339  
onOptions 568, 588  
onPause 568, 589, 724  
onPauseRaw 726  
onPlay 568, 589, 723  
onPlaylistItemStart 776  
onPlaylistItemStop 776  
onPublish 724  
onRecord 568, 589  
onRedirect 569, 589  
onRegisterPlayer 445

onRejectConnection 315  
onResetMediaCaster 449, 543  
onResponseWriteStart 593  
onResponseWriteStop 593  
onResult 485, 498  
onRTPSessionCreate 496, 566  
onRTPSessionDestroy 496, 566  
onSeek 724  
onServerConfigLoaded 605  
onServerCreate 603  
onServerInit 603  
onServerShutdownComplete 604  
onServerShutdownStart 603  
onSetParameter 568, 588  
onSetSourceStream 446  
onSetup 569, 589  
onSharedObjectConnect 633  
onSharedObjectCreate 633  
onSharedObjectDestroy 633  
onSharedObjectDisconnect 634  
onSlotDelete 644  
onSlotSetValue 644  
onStop 724  
onStreamCreate 497, 542  
onStreamDestroy 497, 542  
onStreamStart 448, 731  
onStreamStop 448  
onTeardown 569, 589  
onTrim 481  
onUnbind 322, 324  
onUnPublish 724  
onUnRegisterPlayer 446  
onValidateMediaCaster 449, 542  
onValidateMediaCastersStart 449, 542  
onValidateMediaCastersStop 449, 543  
onVHostClientConnect 856  
onVHostCreate 855  
onVHostInit 855  
onVHostItemCreate 854  
onVHostItemDestroy 854  
onVHostItemUpdate 854  
onVHostShutdownComplete 856  
onVHostShutdownStart 855  
onWriteComplete 759  
open 6, 669, 778

order 171  
output 562

## P

packetComplete 695  
PARAM1 503  
PARAM10 505  
PARAM2 504  
PARAM3 504  
PARAM4 504  
PARAM5 504  
PARAM6 504  
PARAM7 504  
PARAM8 504  
PARAM9 504  
PARAMMETHODNAME 503  
parseAllowDomains 232  
parseBodyForParams 331  
parseEncryptionKeyFile 376  
parseIdString 454  
parseQueryStr 90  
PASSWORDFILEFORMAT\_CLEAR 265  
PASSWORDFILEFORMAT\_UNKNOWN 265  
pathToAdapter 376  
pathToFileURL 88  
pause 526, 741  
PAUSE\_PAUSE 738  
PAUSE\_PLAY 738  
PAUSE\_TOGGLE 738  
pauseRaw 530, 741  
peekByte 109  
ping 295  
play 525, 740, 741, 774, 792, 794, 798, 799, 800  
play2 525  
PLAYEVENT\_AFTERBUFFERFILL 668  
PLAYEVENT\_AFTERMETADATA 668  
PLAYEVENT\_BEFOREBUFFERFILL 668  
PLAYEVENT\_BEFOREMETADATA 668  
PLAYEVENT\_STARTPLAYBACK 667  
Playlist 777  
PlaylistItem 779  
PLAYSIZES\_AUDIO\_BYTES 739  
PLAYSIZES\_AUDIO\_COUNT 739  
PLAYSIZES\_DATA\_BYTES 739

- PLAYSIZES\_DATA\_COUNT 739
- PLAYSIZES\_LOSS\_BYTES 739
- PLAYSIZES\_LOSS\_COUNT 740
- PLAYSIZES\_SIZE 740
- PLAYSIZES\_VIDEO\_BYTES 739
- PLAYSIZES\_VIDEO\_COUNT 739
- PLAYSTATUSTYPE\_COMPLETE 738
- PLAYSTATUSTYPE\_STOP 739
- PLAYSTATUSTYPE\_SWITCH 738
- PLAYTRANSITION\_APPEND 507
- PLAYTRANSITION\_APPEND\_IMMEDIATE 507
- PLAYTRANSITION\_RESET 507
- PLAYTRANSITION\_RESET\_IMMEDIATE 507
- PLAYTRANSITION\_STOP 507
- PLAYTRANSITION\_SWAP 507
- PLAYTRANSITION\_SWITCH 507
- PLAYTRANSITION\_UNKNOWN 507
- PLAYTRANSITIONSTR\_APPEND 506
- PLAYTRANSITIONSTR\_RESET 506
- PLAYTRANSITIONSTR\_STOP 506
- PLAYTRANSITIONSTR\_SWAP 506
- PLAYTRANSITIONSTR\_SWITCH 506
- PLAYTRANSITIONSTR\_UNKNOWN 506
- previous 800, 801
- properties 320, 579
- PROTO\_HTTPCUPERTINO 429
- PROTO\_HTTPSCUPERTINO 430
- PROTO\_HTTPSMOOTH 429
- PROTO\_HTTPSSMOOTH 430
- PROTO\_HTTPSSTREAMER 430
- PROTO\_HTTPSTREAMER 429
- PROTO\_RTMP 428
- PROTO\_RTMPE 429
- PROTO\_RTMPS 428
- PROTO\_RTMPTE 429
- PROTO\_RTMPPTS 429
- PROTO\_RTSP 429
- publish 524, 708, 787, 788
- PublishingProviderLive 791
- PublishingProviderMediaReader 793
- purge 631, 649
- put 172, 173, 636
- putDataTrack 372
- putFileInfo 358
- putGlobalLogValue 407
- putHTTPStreamerAdapter 851
- putMetaData 758
- putSlot 625, 650
- putStreamNameParts 357
- ## Q
- queryStr 581
- ## R
- read 7
- READ\_ACCESS\_ALL 288
- READ\_ACCESS\_NONE 288
- READACCESS 648, 687
- readAppConfig 205
- readAppInstConfig 246
- readChunk 38
- readChunkHeader 39
- readConfig 602, 617
- readHeader 39
- readPrevChunkHeader 39
- readVHostConfig 853
- readXMLConfig 617
- receiveAudio 527
- RECEIVER\_NAME\_KEY 380
- receiveVideo 528
- redirectConnection 293
- referrer 581
- registerCallback 709
- registerOnPlayStatus 709
- registerOnStatus 709
- registerPlayer 438, 462, 473
- registerPlayRTPSession 239
- rejectConnection 275, 292, 293
- rejectSession 356, 582
- release 464, 473, 626, 650
- releaseMediaCaster 478
- releaseStream 524
- reloadConfig 871
- reloadVHostConfig 596, 611
- remove 119, 152, 174, 175, 473, 637, 875
- removeAcceptorListener 840
- removeActionListener 588

removeAppInstance 205  
removeApplication 852  
removeApplicationInstanceListener 203  
removeApplicationListener 839, 875  
removeAttribute 366, 371  
removeClient 626, 638, 650, 836  
removeClientListener 222, 710  
removeConnectionListener 274  
removeDataTrack 373  
removeFromPlaylist 799  
removeGlobalLogValue 407  
removeGlobalLogValues 406  
removeHTTPStreamerSession 245  
removeIdleWorkerListener 846  
removeListener 801  
removeLiveStreamPacketizer 771  
removeLiveStreamPacketizerListener 253  
removeMediaCasterListener 225  
removeMediaReaderListener 251  
removeMediaStreamListener 223, 771  
removeMediaWriterListener 248  
removeMetaDataListener 482  
removeModuleListener 229  
removeNameGroup 772  
removePlayStreamByName 227  
removePublisher 243  
removeQualityLevel 364  
removeRTPSession 239  
removeRTSPStream 583  
removeServerListener 599, 614  
removeSharedObjectListener 224, 638  
removeSlotListener 630, 655  
removeStream 768, 769  
removeStreamDomainStr 358  
removeVHostItemListener 871  
reparentClient 310, 848  
ReplaceItem 86  
requestFilters 320  
RequestFunction 550  
requiresLayout 393  
reset 55, 463, 740, 857, 867  
resetMDC 406  
resetMediaCasterStream 247  
resetNoLookup 740  
resetStream 477

resetTimecodes 745  
resolvePlayAlias 732, 733, 734  
resolveStreamAlias 732, 734  
ResponseFunction 555, 556  
ResponseFunctions 561  
REVERSE 6  
rewind 669  
RTPPushPublishSession 570  
RTPSession 582  
rtpSession 263  
RTPUtils 805  
rtpWriteListener 580  
rtspPlayRangeStart 582  
rtspPlayRangeStop 582  
rtspTunnelingSessionId 578  
run 798

## S

seek 7, 525, 670, 741, 774, 792, 794  
SEEK\_EXACT 668  
SEEK\_KEYCLOSE 668  
SEEK\_KEYDOWN 668  
SEEK\_KEYUP 668  
SEEKTARGET\_AUDIO 667  
SEEKTARGET\_ENHANCED 667  
SEEKTARGET\_VIDEOKEYFRAME 667  
send 630, 654, 705, 706  
sendClientOnStatusError 515  
sendControlBytes 713  
sendDirect 705  
sendInternal 654  
sendLivePlaySeek 696  
sendLivePlayStart 695  
sendLivePlaySwitch 696  
sendPauseNotify 701  
sendPlayReset 697  
sendPlaySeek 698  
sendPlayStart 698, 699, 700  
sendPlayStatus 700, 701  
sendPlayStop 697  
sendPlaySwitch 699  
sendResult 513, 514  
sendStreamNotFound 695  
sendStreamOnStatusError 515

sendUnpauseNotify 701, 702  
sendVODPlaySwitch 696  
sendZeroLengthPacket 671  
serialize 113, 126, 132, 146, 147, 159, 164, 179, 180  
serializeDate 98  
serializeInt 98  
serializeString 99  
serializeStringNoLength 99  
serializeZeroLengthString 98  
Server 610  
serverIp 581  
serverPort 581  
service 341  
sessionClosed 439  
sessionId 578  
sessionOpened 438  
SESSIONPROTOCOL\_COUNT 351  
SESSIONPROTOCOL\_CUPERTINOSTREAMING 350  
SESSIONPROTOCOL\_SANJOSESTREAMING 350  
SESSIONPROTOCOL\_SMOOTHSTREAMING 350  
SESSIONPROTOCOL\_UNKNOWN 350  
SESSIONPROTOCOL\_WEBMSTREAMING 350  
SESSIONTYPE\_LIVE 350  
SESSIONTYPE\_UNKNOWN 350  
SESSIONTYPE\_VOD 350  
set 122, 123, 155, 156  
setAbsTimecode 196  
setAbsTimecodeLong 188  
setAbsTimecodeShort 188  
setAcceptConnection 220, 310  
setAcceptConnectionDescription 309  
setAcceptConnectionExObj 309  
setAcceptConnectionObj 309  
setAcceptSession 356  
setActive 352  
setAdminInterfaceHostPort 839  
setAllowDomains 232  
setAnnounce 585  
setAppend 702  
setAppInstance 353, 437, 585  
setAppInstanceProperty 546  
setApplication 393  
setApplicationInstanceTouchTimeout 250  
setApplicationProperty 545  
setApplicationTimeout 229, 842  
setAttribute 366, 371  
setAudioSize 689  
setAudioTC 692  
setAuthenticatePlayHandler 584  
setAuthenticatePublishHandler 584  
setAuthenticationMethod 322, 327  
setBandwidthLimit 525  
setBaseClass 453, 859  
setBitrate 371  
setBlockSize 56  
setBody 558  
setBufferSize 61, 64  
setBufferTime 299, 526, 688, 741  
setByteContainerLevel 189  
setCDATA 368  
setChecksum 663  
setClassName 180, 182  
setClient 264, 270, 695  
setClientIdleFrequency 232, 844  
setClientProperty 546  
setClientTimeout 838  
setClustered 712  
setCodecPrivateData 371  
setCommandInterface 614  
setCommandInterfaceHostPort 598, 613  
setConfigDir 867  
setConnected 586  
setConnectionLimit 868  
setConnectionTimeout 453  
setConnectionValidator 274  
setCookieStr 357, 590  
setCoreHandlerPoolSize 600, 615, 837  
setCoreTransportPoolSize 599, 615, 837  
setData 62, 65, 641  
setDataBuffer 194  
setDataSize 62, 65, 689  
setDataStr 368  
setDataTC 693  
setDataType 693  
setDebugLog 77  
setDebugRTSPSession 590  
setDescribe 585  
setDescription 453, 861  
setDirection 660  
setDirecton 8

setDomainName 810  
setDuration 775, 792, 794  
setDynamic 183  
setDynamicLogProperties 601, 615, 849  
setEncoding 393  
setEntrySize 62, 65  
setExt 715  
setExtraData 552  
setFastPlay 533  
setFastPlaySettings 296, 706  
setFileExtension 787  
setFileIOPoolSize 837  
setFlashVer 289  
setForceAMF0 559  
setFourCC 371  
setFps 659  
setHeader 335  
setHeaderSize 552, 717  
setHTTPStreamerAdapter 353  
setHTTPStreamerItem 342  
setHTTPStreamerList 242  
setHTTPStreamerSession 720  
setID 343  
setId 186  
setIdleCheckFrequency 845  
setIdleFrequency 303, 589  
setIdleHandler 586  
setIdleMinimumWaitTime 848  
setIdleWorkerCount 845  
setIndex 780  
setInnerObj 183  
setIntData 135  
setIntHeader 335  
setIOPerformanceCounter 591  
setIoPerformanceCounter 612  
setIp 585  
setIpAddress 354, 811  
setIsPlaying 688  
setKeepAliveTime 454  
setKeepAliveTimeout 845  
setLastAccessed 663  
setLastAuthenticateMethod 584  
setLastClientId 643  
setLastOperation 643  
setLastSentAbsTimecode 190  
setLastValidateTime 307  
setLiveRepeaterCapabilities 311  
setLiveStreamingPacketizer 354  
setLiveStreamPacketizer 464, 526, 720  
setLiveStreamPacketizerControl 247  
setLiveStreamPacketizerList 242, 311, 719  
setLiveStreamRepeater 465, 720  
setLoggedConnect 585  
setLongTimecode 190  
setManifestOutput 365  
setMaximumPendingWriteBytes 230, 843  
setMaximumSetBufferTime 231, 843  
setMaxStorageDirDepth 241  
setMediaCasterDef 438  
setMediaCasterHostPortConfig 459  
setMediaCasterId 437  
setMediaCasterItem 730  
setMediaCasterPlay 722  
setMediaCasterProcessorCount 459  
setMediaCasterType 442  
setMediaCasterValidator 249  
setMediaReaderItem 669  
setMediaWriterItem 757  
setMessageBytes 557  
setMetaDataProvider 717  
setMultiplier 659  
setName 202, 218, 364, 452, 629, 642, 654, 690, 742, 859, 867  
setNetConnection 694  
setNetConnectionIdleFrequency 845  
setNetConnectionProcessorCount 839  
setNew 188  
setObjectEncoding 135, 138, 190, 309  
setOpen 706  
setParent 656, 757  
setParentStreamIndex 365  
setPersistent 627, 637, 651  
setPingTimeout 230, 842  
setPlay 694  
setPlayClass 860  
setPlayer 703  
setPlayLogged 353  
setPollingInterval 801  
setPort 394, 811  
setProcessorCount 812  
setProperties 321, 324, 342, 673, 868



setProperty 258, 452, 628, 629, 651, 652, 860  
setPublishAudioCodecId 721  
setPublishVideoCodecId 721  
setQualityLevels 366  
setQueryStr 356, 587, 714  
setRandomAccessReader 61, 64  
setRandomAccessReaderStartPos 61, 64  
setRealTimeStartTime 775, 792, 794  
setReceiveAudio 707  
setReceiveVideo 707  
setReceiveVideoFPS 707  
setReconnectWaitTime 442  
setRecord 693  
setReferrer 356, 587  
setRemoteHost 393  
setRepeat 778, 800  
setRepeaterOriginUrl 231, 307, 529  
setRepeaterQueryString 231  
setRequestFilters 321, 327  
setResponseCode 336  
setRetAMFNumber 558  
setRsoStorageDir 233  
setRTPAVSyncMethod 237  
setRTPIIdleFrequency 232, 844  
setRTPMaXRTCPWaitTime 238  
setRTPPlayAuthenticationMethod 237  
setRTPPublishAuthenticationMethod 237  
setRTPSession 263, 270, 571  
setRTPStream 718  
setRTSPBindIpAddress 252  
setRTSPConnectionAddressType 253  
setRTSPConnectionIpAddress 252  
setRTSPMaximumPendingWriteBytes 250  
setRTSPOriginAddressType 253  
setRTSPOriginIpAddress 253  
setRTSPPlayRangeStart 590  
setRTSPPlayRangeStop 590  
setRTSPSessionTimeout 250  
setRTSPTunnelingSessionId 590  
setSDPData 570  
setSendPlayStopLogEvent 716  
setSendPublishStopLogEvent 716  
setSendRecordStopLogEvent 716  
setSeq 196  
setServer 875  
setServerIp 355, 587  
setServerPort 355, 587  
setSessionId 351, 582  
setSessionProtocol 355  
setSessionTimeout 352  
setSessionType 354  
setSessionValid 584  
setSharedObjectReadAccess 236, 303  
setSharedObjectWriteAccess 236, 303  
setShutdownClient 310, 591  
setShutdownOnRelease 464  
setShuttingDown 848  
setSize 186, 193, 550  
setSlotValue 652, 653  
setSlotVersion 642  
setSoVersion 642  
setSrc 187, 195, 551, 556, 690  
setSSLConfig 814  
setSslFactoryClass 812  
setStartStarupStreams 852  
setStartTC 660  
setStartTCOffset 660  
setStorageDir 630, 638, 655  
setStream 352, 437, 786  
setStreamAudioSampleAccess 235, 305  
setStreamExt 357, 464  
setStreamFileMapper 241  
setStreamKeyDir 233  
setStreamName 357, 767  
setStreamNameAliasProvider 242  
setStreamPosition 358, 673  
setStreamProperty 546  
setStreamReadAccess 235, 305  
setStreamStorageDir 233  
setStreamTimeout 439  
setStreamType 219, 291, 451, 527, 705, 787  
setStreamVideoSampleAccess 234, 304  
setStreamWriteAccess 236, 306  
setSubType 365  
setSuspended 813  
setSwitchLog 802  
setTargetEncoding 138  
setTimecode 187, 195, 368, 552, 559  
setTimecodes 195  
setTimeoutSession 352

setTimeScale 365  
setType 109, 187, 194, 364, 551, 559  
setUri 355, 587  
setUrlStr 365  
setUserAgent 355, 583  
setUserAgents 599, 614  
setValid 462  
setValidateFMLEConnections 254, 312  
setValidationFrequency 230, 843  
setVersion 627, 651  
setVersionFile 757  
setVHost 263, 269, 342, 351, 583  
setVideoSize 689  
setVideoTC 692  
setVisited 868  
setWaitForVideoKeyFrame 757  
setWmsNumber 552  
SharedObject 648, 649  
SHARED\_OBJECT\_CMD\_CONNECT 623  
SHARED\_OBJECT\_CMD\_CONNECT\_SUCCESS 624  
SHARED\_OBJECT\_CMD\_DELETE 624  
SHARED\_OBJECT\_CMD\_DISCONNECT 623  
SHARED\_OBJECT\_CMD\_ERROR 624  
SHARED\_OBJECT\_CMD\_SEND 624  
SHARED\_OBJECT\_CMD\_SETVALUE 623  
SHARED\_OBJECT\_STATUS\_CHANGE 624  
SHARED\_OBJECT\_STATUS\_CLEAR 624  
SHARED\_OBJECT\_STATUS\_DELETE 624  
SHARED\_OBJECT\_STATUS\_SUCCESS 624  
shortValue 145  
shutdown 201, 218, 352, 438, 463, 470, 583, 691, 740, 834, 875  
shutdownAppInstance 205  
shutdownApplication 849  
shutdownClient 218, 291, 582  
shutdownSession 342  
shutdownStream 477  
size 55, 120, 130, 152, 172, 626, 636, 649, 715, 743, 816  
skipByte 110  
SmoothStreamingDataTrack 363  
SmoothStreamingDataTrackData 367  
SmoothStreamingDataTrackDatas 369  
SmoothStreamingDataTrackQualityLevel 370  
SmoothStreamingDataTracks 372  
splitCookie 50  
splitPragmas 49  
splitQueryStr 49  
stampToString 82  
start 86, 610  
startApplicationInstance 850  
startAudioPacket 719  
startCommandInterface 596, 611  
startDataPacket 719  
startMediaCasterStream 247, 248  
startPlay 743  
startPlayback 672  
startPublishing 712  
startRTTPull 805, 806  
startServer 611  
startStartupStreams 852  
startStream 467  
startVHost 597, 611  
startVHosts 597, 612  
startVideoPacket 719  
STAT\_connect\_application\_not\_available 425  
STAT\_connect\_application\_not\_found 425  
STAT\_connect\_bad\_gateway 426  
STAT\_connect\_internal\_error 426  
STAT\_connect\_license\_limit 426  
STAT\_connect\_pending\_wating 425  
STAT\_connect\_redirect 426  
STAT\_connect\_rejected\_by\_application 425  
STAT\_connect\_rejected\_by\_module 425  
STAT\_connect\_resource\_limit 426  
STAT\_connect\_service\_unavailable 426  
STAT\_connect\_successful 425  
STAT\_connect\_unknown\_protocol 425  
STAT\_general\_internal\_error 428  
STAT\_general\_successful 428  
STAT\_play\_bad\_request 426  
STAT\_play\_internal\_error 427  
STAT\_play\_rejected\_by\_application 427  
STAT\_play\_rejected\_by\_module 427  
STAT\_play\_stream\_not\_found 427  
STAT\_play\_successful 426  
STAT\_play\_unsupported\_media\_type 427  
STAT\_publish\_bad\_request 427  
STAT\_publish\_in\_use 428  
STAT\_publish\_internal\_error 428  
STAT\_publish\_rejected\_by\_application 427  
STAT\_publish\_successful 427

- STAT\_publish\_unsupported\_media\_type 428
  - STAT\_stop\_client\_disconnect 428
  - STAT\_stop\_successful 428
  - statusCodeToStr 51
  - stopAdminAgent 602, 610
  - stopCommandInterface 596, 611
  - stopMediaCasterStream 248
  - stopName 691, 742
  - stopPublishing 712
  - stopRTPPull 805
  - stopServer 610
  - stopStartupStreams 852
  - stopStream 467
  - stopVHost 597, 611
  - stopVHosts 597, 611
  - Stream 797
  - streamCodecToString 36
  - streamExists 467
  - StreamItem 859
  - StreamList 862
  - streamLock 764
  - streamMonitorAudioStartTimeout 541
  - streamMonitorAudioTCNegTolerance 540
  - streamMonitorAudioTCPosTolerance 540
  - streamMonitorAudioTCToleranceEnable 540
  - streamMonitorAudioTimeout 541
  - streamMonitorAVSyncTolerance 541
  - streamMonitorAVSyncToleranceEnable 541
  - streamMonitorDataTCNegTolerance 540
  - streamMonitorDataTCPosTolerance 540
  - streamMonitorDataTCToleranceEnable 540
  - streamMonitorDebug 541
  - streamMonitorResetNameGroups 541
  - streamMonitorStreamStartTimeout 539
  - streamMonitorStreamTimeout 539
  - streamMonitorVideoStartTimeout 541
  - streamMonitorVideoTCNegTolerance 540
  - streamMonitorVideoTCPosTolerance 540
  - streamMonitorVideoTCToleranceEnable 540
  - streamMonitorVideoTimeout 541
  - streamNames 763
  - streamNamesLock 764
  - streamNameToGroup 763
  - streamNameToValidFilename 28
  - streams 578, 763
  - streamsOrder 578
  - streamSrcToMediaCaster 473
  - STREAMTIMEOUTREASON\_GOOD 435
  - STREAMTIMEOUTREASON\_MISSING 435
  - STREAMTIMEOUTREASON\_NORTSPSESSION 435
  - STREAMTIMEOUTREASON\_NOSESSION 435
  - STREAMTIMEOUTREASON\_NOSTREAM 435
  - STREAMTIMEOUTREASON\_NOTIMEOUT 435
  - STREAMTIMEOUTREASON\_NOURL 435
  - STREAMTIMEOUTREASON\_RECONNECTRUNNING 435
  - STREAMTIMEOUTREASON\_UNKNOWN 434
  - streamToFileForRead 728
  - streamToFileForWrite 729
  - streamToIndex 767
  - StringUtils 82
  - suspendAllHostPorts 849
  - suspendAllVHosts 601, 612
  - suspendCommandInterface 610
  - suspendServer 602, 612
  - switchName 691, 742
  - SystemUtils 84
- ## T
- TEMP\_CONSOLE\_APPENDER\_NAME 381
  - TEMP\_LIST\_APPENDER\_NAME 381
  - terminate 865
  - testFlashVersion 308
  - testNextByte 109
  - ThreadPool 864
  - timeCreated 580
  - TIMESTAMP\_RULE\_FORMAT 380
  - toArray 131
  - toByteArray 55
  - toByteBuffer 55, 131
  - toHex 24
  - toLong 24
  - toNTPTime 85
  - toString 56, 127, 133, 146, 160, 164, 180, 187, 195, 260, 454, 551, 780, 812, 861, 868
  - totalIOPerformance2Last 580
  - totalIOPerformanceLast 580
  - touch 249, 291, 351, 583, 664
  - touchApplicationInstance 850

toValidFilename 29  
trait 171  
traverseDirectory 30  
triggerAMF3Switch 114  
trim 708  
trimData 363, 364  
truncatePacket 193  
type 109

## U

UDPAppender 392  
unbindAllHostPorts 849  
unbindAllVHosts 601, 612  
unlock 632, 656  
unpublish 787  
unregisterCallback 709  
unregisterOnPlayStatus 709  
unregisterOnStatus 709  
unregisterPlayer 438, 463, 474  
updateIOPerformance 612  
updateLoggingDuration 614, 715, 842  
updateLoggingValues 354, 588, 715, 743  
updateOnCuePointTimecode 45, 46  
updateSDPDestination 805  
uri 581  
urlDecode 376  
urlEncode 376  
urlToId 89  
URLUtils 88  
userAgent 578  
userExists 269

## V

validateConnection 317  
validateNewConnection 274  
versionFile 29  
vhost 263, 578  
VHOST\_DEFAULT 827  
VHostItem 867  
VHostList 871  
VHostSingleton 874  
videoCodecToString 37  
VIDEOSAMPLE\_ACCESS\_ALL 287

VIDEOSAMPLE\_ACCESS\_NONE 287  
VIDEOSAMPLEACCESS 688

## W

warn 402  
WMSLogger 397  
WMSLoggerFactory 405  
WMSLoggerIDs 430  
WMSProperties 258  
wrap 131  
write 55, 59, 559, 560  
WRITE\_ACCESS\_ALL 288  
WRITE\_ACCESS\_NONE 288  
WRITEACCESS 648, 687  
writeAppConfig 205  
writeAppInstConfig 246  
writeChunk 40  
writeCodecConfig 806  
writeConfig 602, 617  
writeControl 57  
WRITECONTROL\_ENHANCEDSEEK\_START 57  
WRITECONTROL\_ENHANCEDSEEK\_STOP 57  
WRITECONTROL\_MEDIAPACKET 57  
writeDeleteError 631, 653  
writeDuration 43  
writeError 653  
writeGeneratedKeyFrame 671  
writeHeader 40  
writePackets 42, 43, 670, 756  
writeSetValueError 630, 653  
writeShortHeader 39  
writeString 139  
writeVHostConfig 853  
writeXMLConfig 617

## X

XMLUtils 92