

Android Documentation

Document current as of 04/17/2019 12:14 PM.

Class AbstractProtocol

Hierarchy

java.lang.Object

Package

`com.smartdevicelink.protocol`

All Known Subclasses

`com.smartdevicelink.protocol.WiProProtocol`

Quick Jump

Constructors

- `AbstractProtocol( IProtocolListener protocolListener )`

Methods

- `SendMessage(ProtocolMessage msg)`
- `getMtu()`
- `getMtu(SessionType type)`
- `handlePacketReceived(SdIPacket packet)`
- `StartProtocolSession(SessionType sessionType)`
- `StartProtocolService(SessionType sessionType,byte sessionID,boolean isEncrypted)`
- `EndProtocolService(SessionType serviceType,byte sessionID)`
- `EndProtocolSession(SessionType sessionType,byte sessionID,int hashID)`
- `SetHeartbeatSendInterval(int heartbeatSendInterval_ms)`
- `SetHeartbeatReceiveInterval(int heartbeatReceiveInterval_ms)`
- `SendHeartBeat(byte sessionID)`
- `SendHeartBeatACK(byte sessionID)`
- `handleProtocolFrameReceived(SdIPacket packet,WiProProtocol.MessageFrameAssembler assembler)`
- `handlePacketToSend(SdIPacket header)`
- `handleProtocolMessageReceived(ProtocolMessage message)`
- `handleProtocolSessionEndedNACK(SessionType sessionType,byte sessionID,String correlationID)`
- `handleProtocolSessionEnded(SessionType sessionType,byte sessionID,String correlationID)`
- `handleProtocolSessionStarted(SessionType sessionType,byte sessionID,byte version,String correlationID,int hashID,boolean isEncrypted)`
- `handleProtocolSessionNACKed(SessionType sessionType,byte sessionID,byte version,String correlationID,java.util.List<java.lang.String> rejectedParams)`
- `handleProtocolError(String string,Exception ex)`
- `handleProtocolHeartbeat(SessionType sessionType,byte sessionID)`
- `handleProtocolHeartbeatACK(SessionType sessionType,byte sessionID)`
- `handleProtocolServiceDataACK(SessionType sessionType,int dataSize,byte sessionID)`
- `onResetIncomingHeartbeat(SessionType sessionType,byte sessionID)`

Constructors

AbstractProtocol(IProtocolListener protocolListener)

```
public AbstractProtocol(IProtocolListener protocolListener)
```

Parameters

protocolListener

Methods

SendMessage(ProtocolMessage msg)

```
public abstract void SendMessage(ProtocolMessage msg)
```

Parameters

msg

getMtu()

```
public abstract int getMtu()
```

getMtu(SessionType type)

```
public abstract long getMtu(SessionType type)
```

Parameters

type

handlePacketReceived(SdlPacket packet)

```
public abstract void handlePacketReceived(SdlPacket packet)
```

Parameters

packet

StartProtocolSession(SessionType sessionType)

```
public abstract void StartProtocolSession(SessionType sessionType)
```

Parameters

sessionType

StartProtocolService(
SessionType sessionType, byte
sessionID, boolean isEncrypted)

```
public abstract void StartProtocolService(SessionType sessionType,byte  
sessionID,boolean isEncrypted)
```

Parameters

sessionType

sessionID

isEncrypted

EndProtocolService(SessionType serviceType, byte sessionID)

```
public abstract void EndProtocolService(SessionType serviceType,byte sessionID)
```

Parameters

serviceType
sessionID

EndProtocolSession(SessionType sessionType, byte sessionID, int hashID)

```
public abstract void EndProtocolSession(SessionType sessionType,byte sessionID,int hashID)
```

Parameters

sessionType
sessionID
hashID

SetHeartbeatSendInterval(int heartbeatSendInterval_ms)

```
public abstract void SetHeartbeatSendInterval(int  
heartbeatSendInterval_ms)
```

Parameters

heartbeatSendInterval_ms

SetHeartbeatReceiveInterval(int heartbeatReceiveInterval_ms)

```
public abstract void SetHeartbeatReceiveInterval(int heartbeatReceiveInterval_ms)
```

Parameters

heartbeatReceiveInterval_ms

SendHeartBeat(byte sessionId)

```
public abstract void SendHeartBeat(byte sessionId)
```

Parameters

sessionId

SendHeartBeatACK(byte sessionID)

```
public abstract void SendHeartBeatACK(byte sessionID)
```

Parameters

sessionID

handleProtocolFrameReceived(SdlPacket packet, WiProProtocol.MessageFrameAssembler assembler)

```
protected void handleProtocolFrameReceived(SdlPacket packet,  
WiProProtocol.MessageFrameAssembler assembler)
```

Parameters

packet

assembler

handlePacketToSend(SdlPacket header)

SdlPacket should have included payload at this point.

```
protected void handlePacketToSend(SdlPacket header)
```

Parameters

header

handleProtocolMessageReceived(ProtocolMessage message)

```
protected void handleProtocolMessageReceived(ProtocolMessage  
message)
```

Parameters

message

handleProtocolSessionEndedNACK (SessionType sessionType, byte sessionID, String correlationID)

```
protected void handleProtocolSessionEndedNACK(SessionType  
sessionType,byte sessionID,String correlationID)
```

Parameters

sessionType
sessionID
correlationID

handleProtocolSessionEnded(
SessionType sessionType, byte
sessionID, String correlationID)

```
protected void handleProtocolSessionEnded(SessionType sessionType,  
byte sessionID,String correlationID)
```

Parameters

sessionType

sessionID
correlationID

handleProtocolSessionStarted(
SessionType sessionType, byte
sessionID, byte version, String
correlationID, int hashID,
boolean isEncrypted)

```
protected void handleProtocolSessionStarted(SessionType sessionType,  
byte sessionID,byte version,String correlationID,int hashID,boolean  
isEncrypted)
```

Parameters

sessionType
sessionID

version
correlationID
hashID
isEncrypted

handleProtocolSessionNACKed(
SessionType sessionType, byte
sessionID, byte version, String
correlationID,
java.util.List<java.lang.String>
rejectedParams)

```
protected void handleProtocolSessionNACKed(SessionType sessionType,  
byte sessionID,byte version,String correlationID,java.util.List<java.lang.  
String> rejectedParams)
```

Parameters

sessionType
sessionID
version
correlationID
rejectedParams

handleProtocolError(String
string, Exception ex)

```
protected void handleProtocolError(String string,Exception ex)
```


Parameters

string
ex

handleProtocolHeartbeat(
SessionType sessionType, byte
sessionID)

```
protected void handleProtocolHeartbeat(SessionType sessionType,byte  
sessionID)
```

Parameters

sessionType
sessionID

handleProtocolHeartbeatACK(SessionType sessionType, byte sessionID)

```
protected void handleProtocolHeartbeatACK(SessionType sessionType,  
byte sessionID)
```

Parameters

sessionType
sessionID

handleProtocolServiceDataACK(SessionType sessionType, int dataSize, byte sessionID)

```
protected void handleProtocolServiceDataACK(SessionType sessionType  
,int dataSize,byte sessionID)
```

Parameters

sessionType
dataSize
sessionID

onResetIncomingHeartbeat(SessionType sessionType, byte sessionID)

```
protected void onResetIncomingHeartbeat(SessionType sessionType,  
byte sessionID)
```

Parameters

sessionType

sessionID

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class AndroidTools

Hierarchy

java.lang.Object

Package

com.smartdevicelink.util

Quick Jump

Constructors

- [AndroidTools\(\)](#)

Methods

- [isServiceExported\(Context context,ComponentName name\)](#)
- [getSdlEnabledApps\(Context context,String myPackageName\)](#)
- [querySdlAppInfo\(Context context,java.util.Comparator<com.smartdevicelink.util.SdlAppInfo> comparator\)](#)
- [sendExplicitBroadcast\(Context context,Intent intent,java.util.List<android.content.pm.ResolveInfo> apps\)](#)
- [isUSBCableConnected\(Context context\)](#)
- [downloadImage\(String urlStr\)](#)

Constructors

AndroidTools()

```
public AndroidTools()
```

Methods

isServiceExported(Context context, ComponentName name)

Check to see if a component is exported

```
public static boolean isServiceExported(Context context,  
ComponentName name)
```

Parameters

context

object used to retrieve the package manager

name

of the component in question

getSdlEnabledApps(Context context, String myPackageName)

Get all SDL enabled apps. If the package name is null, it will return all apps. However, if the package name is included, the resulting hash map will not include the app with that package name.

```
public static java.util.HashMap getSdlEnabledApps(Context context, String myPackageName)
```

Parameters

context

a context object used to get an instance of the package manager

myPackageName

the package of the requesting app. This should only be included if the app wants to exclude itself from the map

querySdlAppInfo(Context context, java.util.Comparator<com.smartdevice comparator)

Finds all SDL apps via their SdlRouterService manifest entry. It will return the metadata associated with that router service.

```
public static java.util.List querySdlAppInfo(Context context, java.util.  
Comparator<com.smartdevice.link.util.SdlAppInfo> comparator)
```

Parameters

context

a context instance to obtain the package manager

comparator

the Comparator to sort the resulting list. If null is supplied, they will be returned as they are from the system

sendExplicitBroadcast(Context context, Intent intent, java.util.List<android.content.pm.ResolveInfo> apps)

Sends the provided intent to the specified destinations making it an explicit intent, rather than an implicit intent. A direct replacement of `sendBroadcast(Intent)`. As of Android 8.0 (API 26+) implicit broadcasts are no longer sent to broadcast receivers that are declared via the AndroidManifest. If no apps are found to receive the intent, this method will send the broadcast implicitly if no list of apps is provided.

```
public static void sendExplicitBroadcast(Context context, Intent intent,
    java.util.List<android.content.pm.ResolveInfo> apps)
```

Parameters

context
intent - the intent to send explicitly
apps

- the list of apps that this broadcast will be sent to. If null is passed in the intent will be sent to all apps that match the provided intent via a query to the package manager; it will also be sent implicitly to mimic `sendBroadcast()`'s original functionality.

isUSBCableConnected(Context context)

Checks if the usb cable is physically connected or not Note: the intent here is a sticky intent so `registerReceiver` is actually a synchronous call and doesn't register a receiver on each call

```
public static boolean isUSBCableConnected(Context context)
```

Parameters

context

a context instance

downloadImage(String urlStr)

```
public static android.graphics.Bitmap downloadImage(String urlStr)  
throws java.io.IOException
```

Parameters

urlStr

Throws

- `java.io.IOException`

Inherited Methods

From Class	Methods
<code>java.lang.Object</code>	<code>getClass</code> , <code>hashCode</code> , <code>equals</code> , <code>clone</code> , <code>toString</code> , <code>notify</code> , <code>notifyAll</code> , <code>wait</code> , <code>wait</code> , <code>wait</code> , <code>finalize</code>

Class AudioDecoder

Hierarchy

`java.lang.Object`

`com.smartdevicelink.managers.audio.BaseAudioDecoder`

Package

com.smartdevicelink.managers.audio

Overview

The audio decoder to decode a single audio file to PCM.

Quick Jump

Methods

- [start\(\)](#)

Fields

Inherited Fields

From Class	Fields
com.smartdevicelink.managers.audio.BaseAudioDecoder	targetSampleRate , targetSampleType , extractor , decoder , audioSource , contextWeakReference , listener

Methods

start()

Starts the audio decoding asynchronously.

```
public void start()
```

Inherited Methods

From Class	Methods
com.smartdevicelink.managers.audio.BaseAudioDecoder	initMediaComponents , onInputBufferAvailable , onOutputBufferAvailable , onOutputFormatChanged , onMediaCodecError , start , stop
java.lang.Object	getClass , hashCode , equals , clone , toString , notify , notifyAll , wait , wait , wait , finalize

Class `AudioDecoderCompat`

Hierarchy

[java.lang.Object](#)

[com.smartdevicelink.managers.audio.BaseAudioDecoder](#)

Package

com.smartdevicelink.managers.audio

Overview

The audio decoder to decode a single audio file to PCM. This decoder supports phones with api < 21 but uses methods deprecated with api 21.

Quick Jump

Methods

- [start\(\)](#)

Fields

Inherited Fields

From Class	Fields
com.smartdevicelink.managers.audio.BaseAudioDecoder	targetSampleRate , targetSampleType , extractor , decoder , audioSource , contextWeakReference , listener

Methods

start()

Starts the audio decoding asynchronously.

```
public void start()
```

Inherited Methods

From Class	Methods
com.smartdevicelink.managers.audio.BaseAudioDecoder	initMediaComponents, onInputBufferAvailable, onOutputBufferAvailable, onOutputFormatChanged, onMediaCodecError, start, stop
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Interface AudioDecoderListener

Package

`com.smartdevicelink.managers.audio`

Overview

An interface for the audio decoder classes. The caller using the audio decoder will be notified when the decoding is finished or if an error occurred. During decoding the caller receives sample buffers with decoded audio data.

Quick Jump

Methods

- [onAudioDataAvailable\(SampleBuffer sampleBuffer\)](#)
- [onDecoderFinish\(boolean success\)](#)
- [onDecoderError\(Exception e\)](#)

Methods

onAudioDataAvailable(SampleBuffer sampleBuffer)

Notifies that decoded audio data is available.

```
public void onAudioDataAvailable(SampleBuffer sampleBuffer)
```

Parameters

sampleBuffer

The sample buffer holding the decoded audio data.

onDecoderFinish(boolean success)

Notifies that the audio decoding is finished.

```
public void onDecoderFinish(boolean success)
```

Parameters

success

Indicates whether audio decoding was successful or if an error occurred.

onDecoderError(Exception e)

Notifies the caller that an error/exception occurred during audio decoding.

```
public void onDecoderError(Exception e)
```

Parameters

e

The exception storing information about the error.

Class AudioManager

Hierarchy

java.lang.Object

BaseAudioStreamManager

Package

com.smartdevicelink.managers.audio

Overview

The AudioManager class provides methods to start and stop an audio stream to the connected device. Audio files can be pushed to the manager in order to play them on the connected device. The manager uses the Android built-in MediaCodec.

Quick Jump

Constructors

- [AudioStreamManager\(ISdl internalInterface, Context context \)](#)

Methods

- [start\(CompletionListener listener\)](#)
- [dispose\(\)](#)

- `startAudioStream(boolean encrypted,CompletionListener completionListener)`
- `stopAudioStream(CompletionListener completionListener)`
- `pushResource(int resourceId,CompletionListener completionListener)`
- `pushAudioSource(Uri audioSource,CompletionListener completionListener)`
- `onTransportUpdate(java.util.List<TransportRecord> connectedTransports,boolean audioStreamTransportAvail,boolean videoStreamTransportAvail)`

Constructors

AudioStreamManager(ISdl internalInterface, Context context)

Creates a new object of AudioStreamManager

```
public AudioStreamManager(ISdl internalInterface,Context context)
```

Parameters

internalInterface

The internal interface to the connected device.

context

Methods

start(CompletionListener
listener)

```
public void start(CompletionListener listener)
```

Parameters

listener

dispose()

```
public void dispose()
```

startAudioStream(boolean encrypted, CompletionListener completionListener)

Starts the audio service and audio stream to the connected device. The method is non-blocking.

```
public void startAudioStream(boolean encrypted,CompletionListener completionListener)
```

Parameters

encrypted

Specify whether or not the audio stream should be encrypted.

completionListener

stopAudioStream(CompletionListener completionListener)

Stops the audio service and audio stream to the connected device. The method is non-blocking.

```
public void stopAudioStream(CompletionListener completionListener)
```

Parameters

completionListener

pushResource(int resourceId, CompletionListener completionListener)

Pushes the specified resource file to the playback queue. The audio file will be played immediately. If another audio file is currently playing the specified file will stay queued and automatically played when ready.

```
public void pushResource(int resourceId,CompletionListener  
completionListener)
```

Parameters

resourceId

The specified resource file to be played.

completionListener

A completion listener that informs when the audio file is played.

pushAudioSource(Uri audioSource, CompletionListener completionListener)

Pushes the specified audio file to the playback queue. The audio file will be played immediately. If another audio file is currently playing the specified file will stay queued and automatically played when ready.

```
public void pushAudioSource(Uri audioSource,CompletionListener  
completionListener)
```


Parameters

audioSource

The specified audio file to be played.

completionListener

A completion listener that informs when the audio file is played.

onTransportUpdate(
java.util.List<TransportRecord>
connectedTransports, boolean
audioStreamTransportAvail,
boolean
videoStreamTransportAvail)

```
protected void onTransportUpdate(java.util.List<TransportRecord>  
connectedTransports,boolean audioStreamTransportAvail,boolean  
videoStreamTransportAvail)
```

Parameters

connectedTransports
audioStreamTransportAvail
videoStreamTransportAvail

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

AudioStreamManager.SampleType

Quick Jump

Class BTTransport

Hierarchy

java.lang.Object

[com.smartdevicelink.transport.SdlTransport](#)

Package

com.smartdevicelink.transport

Overview

Bluetooth Transport Implementation. This transport advertises its existence to SDL by publishing an SDP record and waiting for an incoming connection from SDL. Connection is verified by checking for the SDL UUID. For more detailed information please refer to the [Bluetooth Transport Guide](#).

Quick Jump

Constructors

- `BTTransport( ITransportListener transportListener )`
- `BTTransport( ITransportListener transportListener, boolean bKeepSocket )`

Methods

- `getBTSocket(BluetoothServerSocket bsSocket) Deprecated`
- `getChannel(BluetoothSocket bsSocket) Deprecated`
- `openConnection()`
- `disconnect()`
- `sendBytesOverTransport(SdIPacket packet)`
- `getTransportType()`
- `getBroadcastComment()`
- `handleTransportDisconnected(String info)`
- `handleTransportError(String message,Exception ex)`

Constructors

BTTransport(ITransportListener transportListener)

```
public BTTransport(ITransportListener transportListener)
```

Parameters

transportListener

BTTransport(ITransportListener
transportListener, boolean
bKeepSocket)

```
public BTTransport(ITransportListener transportListener,boolean  
bKeepSocket)
```

Parameters

transportListener

bKeepSocket

Methods

getBTSocket(
BluetoothServerSocket bsSocket
)

Deprecated

```
public android.bluetooth.BluetoothSocket getBTSocket(  
BluetoothServerSocket bsSocket)
```

Parameters

bsSocket

getChannel(BluetoothSocket
bsSocket)

Deprecated

```
public int getChannel(BluetoothSocket bsSocket)
```

Parameters

bsSocket

openConnection()

```
public void openConnection() throws SdlException
```

Throws

- SdlException

disconnect()

```
public void disconnect()
```

sendBytesOverTransport(SdlPacket packet)

Sends data over the transport. Takes a byte array and transmits data provided starting at the offset and of the provided length to fragment transmission.

```
public boolean sendBytesOverTransport(SdlPacket packet)
```

Parameters

packet

getTransportType()

Overridden abstract method which returns specific type of this transport.

```
public TransportType getTransportType()
```


getBroadcastComment()

```
public java.lang.String getBroadcastComment()
```

handleTransportDisconnected(String info)

```
protected void handleTransportDisconnected(String info)
```

Parameters

info

handleTransportError(String message, Exception ex)

```
protected void handleError(String message,Exception ex)
```

Parameters

message

ex

Inherited Methods

From Class	Methods
com.smartdevicelink.transport.SdlTransport	getIsConnected, handleReceivedPacket, sendBytesOverTransport, sendBytes, handleTransportConnected, handleTransportDisconnected, handleError, openConnection, disconnect, getTransportType, getBroadcastComment
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class BTTransportConfig

Hierarchy

java.lang.Object

BaseTransportConfig

Package

com.smartdevicelink.transport

Overview

Container of Bluetooth transport specific configuration.

Quick Jump

Constructors

- [BTTransportConfig\(\)](#)
- [BTTransportConfig\(boolean shareConnection \)](#)

Methods

- [getTransportType\(\)](#)
- [setKeepSocketActive\(boolean bValue\)](#)
- [getKeepSocketActive\(\)](#)

Constructors

BTTransportConfig()

```
public BTTransportConfig()
```

BTTransportConfig(boolean shareConnection)

```
public BTTransportConfig(boolean shareConnection)
```

Parameters

shareConnection

Methods

getTransportType()

Overridden abstract method which returns specific type of this transport configuration.

```
public TransportType getTransportType()
```

setKeepSocketActive(boolean bValue)

```
public void setKeepSocketActive(boolean bValue)
```

Parameters

bValue

getKeepSocketActive()

```
public boolean getKeepSocketActive()
```

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class BaseAudioDecoder

Hierarchy

java.lang.Object

Package

com.smartdevicelink.managers.audio

All Known Subclasses

com.smartdevicelink.managers.audio.AudioDecoder,
com.smartdevicelink.managers.audio.AudioDecoderCompat

Quick Jump

Fields

- `targetSampleRate`
- `targetSampleType`
- `extractor`
- `decoder`
- `audioSource`
- `contextWeakReference`
- `listener`

Constructors

- `BaseAudioDecoder( Uri audioSource, Context context, int sampleRate, int sampleType, AudioDecoderListener listener )`

Methods

- `initMediaComponents()`
- `onInputBufferAvailable(MediaExtractor extractor,ByteBuffer inputBuffer)`
- `onOutputBufferAvailable(ByteBuffer outputBuffer)`
- `onOutputFormatChanged(MediaFormat mediaFormat)`
- `onMediaCodecError(MediaCodec.CodecException e)`
- `start()`

- `stop()`

Fields

targetSampleRate

```
protected final int targetSampleRate
```

targetSampleType

```
protected final int targetSampleType
```

extractor

```
protected android.media.MediaExtractor extractor
```

decoder

```
protected android.media.MediaCodec decoder
```

audioSource

```
protected final android.net.Uri audioSource
```

contextWeakReference

```
protected final java.lang.ref.WeakReference contextWeakReference
```

listener

```
protected final com.smartdevicelink.managers.audio.  
AudioDecoderListener listener
```

Constructors

BaseAudioDecoder(Uri
audioSource, Context context,
int sampleRate, int sampleType,
AudioDecoderListener listener)

```
public BaseAudioDecoder(Uri audioSource,Context context,int  
sampleRate,int sampleType,AudioDecoderListener listener)
```

Parameters

audioSource

Uri of the audio source to be converted
context the context
sampleRate can be either 8000, 16000, 22050 or 44100
sampleType can be either UNSIGNED_8_BIT, SIGNED_16_BIT, FLOAT
listener listener for event callbacks

Methods

initMediaComponents()

```
protected void initMediaComponents() throws java.lang.Exception
```

Throws

- `java.lang.Exception`

`onInputBufferAvailable(
MediaExtractor extractor,
ByteBuffer inputBuffer)`

```
protected android.media.MediaCodec.BufferInfo onInputBufferAvailable(  
MediaExtractor extractor,ByteBuffer inputBuffer)
```

Parameters

extractor
inputBuffer

`onOutputBufferAvailable(
ByteBuffer outputBuffer)`

```
protected com.smartdevicelink.managers.audio.SampleBuffer  
onOutputBufferAvailable(ByteBuffer outputBuffer)
```

Parameters

outputBuffer

onOutputFormatChanged(MediaFormat mediaFormat)

```
protected void onOutputFormatChanged(MediaFormat mediaFormat)
```

Parameters

mediaFormat

onMediaCodecError(MediaCodec.CodecException e)

```
protected void onMediaCodecError(MediaCodec.CodecException e)
```

Parameters

e

start()

```
public abstract void start()
```

stop()

```
public void stop()
```

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class ByteArrayMessageAssembler

Hierarchy

java.lang.Object

Package

com.smartdevicelink.transport.utl

Quick Jump

Constructors

- [ByteArrayMessageAssembler\(\)](#)

Methods

- `init()`
- `setTransportType(TransportType transportType)`
- `getTransportType()`
- `close()`
- `append(byte[] bytes)`
- `handleMessage(int flags,byte[] packet)`
- `getBytes()`
- `isFinished()`

Constructors

ByteArrayMessageAssembler()

```
public ByteArrayMessageAssembler()
```

Methods

init()

```
public void init()
```


setTransportType(TransportType transportType)

```
public void setTransportType(TransportType transportType)
```

Parameters

```
transportType
```

getTransportType()

```
public TransportType getTransportType()
```

close()

```
public boolean close()
```

append(byte[] bytes)

```
public void append(byte[] bytes)
```

Parameters

bytes

handleMessage(int flags, byte[] packet)

```
public synchronized boolean handleMessage(int flags,byte[] packet)
```

Parameters

flags

packet

getBytes()

```
public byte getBytes()
```

isFinished()

```
public boolean isFinished()
```

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class

ByteArrayMessageSplitter

Hierarchy

java.lang.Object

Package

com.smartdevicelink.transport.utl

Quick Jump

Fields

- [MAX_BINDER_SIZE](#)

Constructors

- [ByteArrayMessageSplitter\(String appId, int what, byte\[\] bytes, int priorityCoef \)](#)
- [ByteArrayMessageSplitter\(Long appId, int what, byte\[\] bytes, int priorityCoef \)](#)

Methods

- [setRouterServiceVersion\(int version\)](#)
- [setTransportRecord\(TransportRecord transportRecord\)](#)
- [isActive\(\)](#)
- [close\(\)](#)

- `nextMessage()`

Fields

MAX_BINDER_SIZE

```
public static final int MAX_BINDER_SIZE
```

Constant Value

250000

Constructors

ByteArrayMessageSplitter(String appld, int what, byte[] bytes, int priorityCoef)

```
public ByteArrayMessageSplitter(String appld,int what,byte[] bytes,int  
priorityCoef)
```

Parameters

appld
what
bytes
priorityCoef

ByteArrayMessageSplitter(Long
appld, int what, byte[] bytes, int
priorityCoef)

```
public ByteArrayMessageSplitter(Long appld,int what,byte[] bytes,int  
priorityCoef)
```

Parameters

appld
what
bytes
priorityCoef

Methods

setRouterServiceVersion(int
version)

```
public void setRouterServiceVersion(int version)
```

Parameters

version

setTransportRecord(
TransportRecord transportRecord
)

```
public void setTransportRecord(TransportRecord transportRecord)
```

Parameters

transportRecord

isActive()


```
public boolean isActive()
```

close()

```
public boolean close()
```

nextMessage()

```
public android.os.Message nextMessage()
```

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class EncoderUtils

Hierarchy

java.lang.Object

Package

com.smartdevice.link.encoder

Quick Jump

Methods

- [getCodecSpecificData\(MediaFormat format\)](#)

Methods

getCodecSpecificData(MediaFormat format)

Extracts codec-specific data from MediaFormat instance Currently, only AVC is supported.

```
public static byte getCodecSpecificData(MediaFormat format)
```

Parameters

format

MediaFormat instance retrieved from MediaCodec

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class FileManager

Hierarchy

java.lang.Object

BaseFileManager

Package

com.smartdevicelink.managers.file

Overview

FileManager

Note: This class must be accessed through the SdlManager. Do not instantiate it by itself.

The SDLFileManager uploads files and keeps track of all the uploaded files names during a session.

We need to add the following struct: SDLFile

It is broken down to these areas:

1. Getters
2. Deletion methods
3. Uploading Files / Artwork

Quick Jump

Constructors

- `FileManager( ISdl internalInterface, Context context )`

Constructors

FileManager(ISdl internalInterface, Context context)

```
public FileManager(ISdl internalInterface, Context context)
```

Parameters

internalInterface
context

Methods

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class HapticInterfaceManager

Hierarchy

java.lang.Object

Package
com.smartdevicelink.haptic

Overview

Created on 9/22/2017. Manages haptic data used to render focusable areas on the HU screen. App developers can over-ride the default logic used to find focusable Views by passing their own data to {@link #setHapticData(List)}

Quick Jump

Constructors

- [HapticInterfaceManager\(ISdl proxy \)](#)

Methods

- [setHapticData\(java.util.List<HapticRect> hapticData\)](#)
- [refreshHapticData\(View root\)](#)

Constructors

HapticInterfaceManager(ISdl proxy)

```
public HapticInterfaceManager(ISdl proxy)
```

Parameters

proxy

Methods

setHapticData(java.util.List<HapticRect> hapticData)

Sets haptic data and sends update to the HU. To be used by app code instead of letting Presentation find the Views and automatically send to HU.

```
public void setHapticData(java.util.List<HapticRect> hapticData)
```

Parameters

hapticData

Rect data indicating "focusable" screen elements or areas

refreshHapticData(View root)

Sends haptic data found by searching for focusable and clickable Views in the view heirarchy to the HU. Should be called by Presentation's OnShowListener.


```
public void refreshHapticData(View root)
```

Parameters

root
the root or parent View

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class HeartbeatMonitor

Hierarchy

java.lang.Object

Package

`com.smartdevicelink.protocol.heartbeat`

All Implemented Interfaces

`com.smartdevicelink.protocol.heartbeat.IHeartbeatMonitor`

Quick Jump

Fields

- `HEARTBEAT_INTERVAL`
- `HEARTBEAT_INTERVAL_MAX`

Constructors

- `HeartbeatMonitor()`

Methods

- `getHeartbeatRunnable()`
- `isHeartbeatReceived()`
- `start()`
- `stop()`
- `getInterval()`
- `setInterval(int value)`
- `getListener()`
- `setListener(IHeartbeatMonitorListener value)`
- `notifyTransportActivity()`
- `heartbeatACKReceived()`

- `heartbeatReceived()`

Fields

HEARTBEAT_INTERVAL

```
public static final int HEARTBEAT_INTERVAL
```

Constant Value

5000

HEARTBEAT_INTERVAL_MAX

```
public static final int HEARTBEAT_INTERVAL_MAX
```

Constant Value

2147483647

Constructors

HeartbeatMonitor()

```
public HeartbeatMonitor()
```

Methods

getHeartbeatRunnable()

```
public java.lang.Runnable getHeartbeatRunnable()
```

isHeartbeatReceived()

```
public boolean isHeartbeatReceived()
```

start()

```
public void start()
```

stop()

```
public void stop()
```

getInterval()

```
public int getInterval()
```

setInterval(int value)

```
public void setInterval(int value)
```

Parameters

value

getListener()

```
public com.smartdevicelink.protocol.heartbeat.  
IHeartbeatMonitorListener getListener()
```

setListener(IHeartbeatMonitorListener value)

```
public void setListener(IHeartbeatMonitorListener value)
```

Parameters

value

notifyTransportActivity()

```
public void notifyTransportActivity()
```

heartbeatACKReceived()

```
public synchronized void heartbeatACKReceived()
```

heartbeatReceived()

```
public void heartbeatReceived()
```

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class HttpUtils

Hierarchy

java.lang.Object

Package

com.smartdevicelink.util

Quick Jump

Constructors

- [HttpUtils\(\)](#)

Methods

- `downloadImage(String urlStr)`

Constructors

HttpUtils()

```
public HttpUtils()
```

Methods

downloadImage(String urlStr)

```
public static android.graphics.Bitmap downloadImage(String urlStr)  
throws java.io.IOException
```

Parameters

urlStr

Throws

- java.io.IOException

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Interface IHeartbeatMonitor

Package

com.smartdevicelink.protocol.heartbeat

All Known Implementing Classes

`com.smartdevicelink.protocol.heartbeat.HeartbeatMonitor`

Quick Jump

Methods

- `start()`
- `stop()`
- `getInterval()`
- `setInterval(int interval)`
- `getListener()`
- `setListener(IHeartbeatMonitorListener listener)`
- `notifyTransportActivity()`
- `heartbeatACKReceived()`
- `heartbeatReceived()`

Methods

`start()`

Starts the monitor. If the monitor is already started, nothing happens.

```
public void start()
```

stop()

Stops the monitor. Does nothing if it is already stopped.

```
public void stop()
```

getInterval()

Returns the heartbeat messages interval.

```
public int getInterval()
```

setInterval(int interval)

Sets the interval for sending heartbeat messages if nothing is sent over transport.

```
public void setInterval(int interval)
```

Parameters

interval

interval in milliseconds (min/max values depend on concrete implementations)

getListener()

Returns the listener.

```
public com.smartdevicelink.protocol.heartbeat.  
IHeartbeatMonitorListener getListener()
```

setListener(IHeartbeatMonitorListener listener)

Sets the heartbeat's listener.

```
public void setListener(IHeartbeatMonitorListener listener)
```

Parameters

listener

the new listener

notifyTransportActivity()

Notifies the monitor about sent/received messages.

```
public void notifyTransportActivity()
```

heartbeatACKReceived()

Notifies the monitor about a received heartbeat ACK message.

```
public void heartbeatACKReceived()
```

heartbeatReceived()

Notifies the monitor about a received heartbeat message.

```
public void heartbeatReceived()
```

Interface IHeartbeatMonitorListener

Package

com.smartdevicelink.protocol.heartbeat

All Known Implementing Classes

com.smartdevicelink.SdlConnection.SdlSession,
com.smartdevicelink.SdlConnection.SdlSession2

Overview

Listener (aka delegate) that actually knows how to work with the outside world.

Quick Jump

Methods

- `sendHeartbeat(IHeartbeatMonitor monitor)`
- `heartbeatTimedOut(IHeartbeatMonitor monitor)`

Methods

sendHeartbeat(IHeartbeatMonitor monitor)

The listener should create and send a heartbeat message.

```
public void sendHeartbeat(IHeartbeatMonitor monitor)
```

Parameters

monitor

the caller

heartbeatTimedOut(IHeartbeatMonitor monitor)

Notifies the listener that the heartbeat timed out.

```
public void heartbeatTimedOut(IHeartbeatMonitor monitor)
```

Parameters

monitor

the caller

Class LockScreenConfig

Hierarchy

java.lang.Object

Package

com.smartdevicelink.managers.lockscreen

Overview

LockScreenConfig

This is set during SdlManager instantiation.

- `enable` - if false, don't worry about the other parameters. You are responsible for creating and managing a lockscreen. If true, also set the `backgroundColor` and `applcon` if you want. If you don't set the `backgroundColor` or `applcon`, it will use the defaults.
- `backgroundColor` - if using the default lockscreen, you can set this to a color of your choosing
- `applcon` - if using the default lockscreen, you can set your own app icon
- `customView` - If you would like to provide your own view, you can pass it in here.

Quick Jump

Constructors

- `LockScreenConfig()`

Methods

- `setEnabled(boolean enable)`
- `isEnabled()`
- `setBackgroundColor(int resourceColor)`
- `getBackgroundColor()`
- `setApplcon(int applconInt)`
- `getApplcon()`
- `setCustomView(int customViewInt)`
- `getCustomView()`
- `showDeviceLogo(boolean deviceLogo)`
- `isDeviceLogoEnabled()`

Constructors

LockScreenConfig()

```
public LockScreenConfig()
```

Methods

setEnabled(boolean enable)

If set to true, SDL will manage the showing and dismissing of the lock screen for you.

If false, you must manage the lock screen

```
public void setEnabled(boolean enable)
```

Parameters

enable

boolean

isEnabled()

Gets whether the lock screen is being managed for you

```
public boolean isEnabled()
```

setBackgroundColor(int resourceColor)

Set the resource int of the background color. Colors should define colors in your Colors.xml file

```
public void setBackgroundColor(int resourceColor)
```

Parameters

resourceColor

resource int of the color

getBackgroundColor()

Gets the int reference to the custom lock screen background color

```
public int getBackgroundColor()
```

setAppIcon(int appIconInt)

int of the drawable icon.

```
public void setAppIcon(int appIconInt)
```

Parameters

appIconInt

the drawable of the icon to be displayed on the lock screen

getAppIcon()

Gets the resource reference of the icon to be displayed on the lock screen

```
public int getAppIcon()
```

setCustomView(int customViewInt)

Sets the reference to the custom layout to be used for the lock screen

If set, the color and icon setters will be ignored

```
public void setCustomView(int customViewInt)
```

Parameters

customViewInt

the layout

getCustomView()

Gets the reference to the custom lockscreen layout to be used

```
public int getCustomView()
```

showDeviceLogo(boolean deviceLogo)

Whether or not to show the device's logo on the default lock screen

The logo will come from the connected hardware, if set by the manufacturer

If using a Custom View, this will be ignored.

```
public void showDeviceLogo(boolean deviceLogo)
```

Parameters

deviceLogo

- boolean

isDeviceLogoEnabled()

Get whether or not the device's Logo is shown on the default lock screen
The logo will come from the connected hardware, if set by the manufacturer

```
public boolean isDeviceLogoEnabled()
```

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class LockScreenManager

Hierarchy

java.lang.Object

BaseSubManager

Package

com.smartdevicelink.managers.lockscreen

Overview

LockscreenManager

Note: This class must be accessed through the SdlManager. Do not instantiate it by itself.

The LockscreenManager handles the logic of showing and hiding the lock screen.

Quick Jump

Fields

- `lockScreenEnabled`
- `deviceLogoEnabled`
- `lockScreenIcon`
- `lockScreenColor`
- `customView`
- `deviceLogo`

Constructors

- `LockScreenManager( LockScreenConfig lockScreenConfig, Context context, ISdl internalInterface )`

Methods

- `start(CompletionListener listener)`
- `dispose()`

- `getLockScreenStatus()`

Fields

lockScreenEnabled

```
protected boolean lockScreenEnabled
```

deviceLogoEnabled

```
protected boolean deviceLogoEnabled
```

lockScreenIcon

```
protected int lockScreenIcon
```

lockScreenColor

```
protected int lockScreenColor
```

customView

```
protected int customView
```

deviceLogo

```
protected android.graphics.Bitmap deviceLogo
```

Constructors

LockScreenManager(LockScreenConfig lockScreenConfig, Context context, ISdl internalInterface)

```
public LockScreenManager(LockScreenConfig lockScreenConfig, Context  
context, ISdl internalInterface)
```

Parameters

lockScreenConfig
context
internalInterface

Methods

start(CompletionListener listener)

```
public void start(CompletionListener listener)
```

Parameters

listener

dispose()

```
public void dispose()
```

getLockScreenStatus()

Step through some logic to determine if we need to show the lock screen or not
This function is usually triggered on some sort of notification.

```
protected synchronized LockScreenStatus getLockScreenStatus()
```

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class LockScreenManager

Hierarchy

java.lang.Object

Package

com.smartdevicelink.proxy

Quick Jump

Constructors

- `LockScreenManager()`

Methods

- `setSessionID(int iVal)`
- `setDriverDistStatus(boolean bVal)`
- `setHMILevel(HMILevel hmiVal)`
- `getLockObj()`
- `downloadLockScreenIcon(String url, LockScreenManager.OnLockScreenIconDownloadedListener l)`
- `getLockScreenIcon()`

Constructors

LockScreenManager()

```
public LockScreenManager()
```

Methods

setSessionID(int iVal)

```
public synchronized void setSessionID(int iVal)
```

Parameters

iVal

setDriverDistStatus(boolean bVal)

```
public synchronized void setDriverDistStatus(boolean bVal)
```

Parameters

bVal


```
setHMILevel( HMILevel hmiVal )
```

```
public synchronized void setHMILevel(HMILevel hmiVal)
```

Parameters

```
hmiVal
```

```
getLockObj()
```

```
public synchronized OnLockScreenStatus getLockObj()
```

```
downloadLockScreenIcon( String  
url,  
LockScreenManager.OnLockScreenIcon  
l )
```

```
public void downloadLockScreenIcon(String url,LockScreenManager.  
OnLockScreenIconDownloadedListener l)
```

Parameters

url
l

getLockScreenIcon()

```
public android.graphics.Bitmap getLockScreenIcon()
```

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Interface LockScreenManager.OnLockScreen

Package

com.smartdevicelink.proxy

Quick Jump

Methods

- [onLockScreenIconDownloaded\(Bitmap icon\)](#)
- [onLockScreenIconDownloadError\(Exception e\)](#)

Methods

onLockScreenIconDownloaded(Bitmap icon)

```
public void onLockScreenIconDownloaded(Bitmap icon)
```

Parameters

icon

onLockScreenIconDownloadError (Exception e)

```
public void onLockScreenIconDownloadError(Exception e)
```

Parameters

e

Class MultiplexBaseTransport

Hierarchy

java.lang.Object

Package

com.smartdevicelink.transport

All Known Subclasses

com.smartdevicelink.transport.MultiplexBluetoothTransport,
com.smartdevicelink.transport.MultiplexTcpTransport,
com.smartdevicelink.transport.MultiplexUsbTransport

Quick Jump

Fields

- `STATE_NONE`
- `STATE_LISTEN`
- `STATE_CONNECTING`
- `STATE_CONNECTED`
- `STATE_ERROR`
- `LOG`
- `DEVICE_NAME`
- `DEVICE_ADDRESS`
- `mState`
- `handler`
- `transportType`
- `transportRecord`
- `currentlyConnectedDevice`
- `connectedDeviceName`
- `connectedDeviceAddress`

Constructors

- `MultiplexBaseTransport( Handler handler, TransportType transportType )`

Methods

- `setState(int state)`
- `getAddress()`
- `getDeviceName()`
- `getTransportRecord()`
- `getState()`
- `isConnected()`
- `stop()`
- `stop(int state)`
- `write(byte[] out,int offset,int count)`

Fields

STATE_NONE

```
public static final int STATE_NONE
```

STATE_LISTEN

```
public static final int STATE_LISTEN
```

Constant Value

1

STATE_CONNECTING

```
public static final int STATE_CONNECTING
```

Constant Value

2

STATE_CONNECTED

```
public static final int STATE_CONNECTED
```

Constant Value

3

STATE_ERROR

```
public static final int STATE_ERROR
```

Constant Value

4

LOG

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

Constant Value

"log"

DEVICE_NAME

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

Constant Value

"device_name"

DEVICE_ADDRESS

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

Constant Value

"device_address"

mState

```
protected int mState
```

handler

```
protected final android.os.Handler handler
```

transportType

```
protected final TransportType transportType
```

transportRecord

```
protected TransportRecord transportRecord
```

currentlyConnectedDevice

```
public static java.lang.String currentlyConnectedDevice
```

connectedDeviceName

```
protected java.lang.String connectedDeviceName
```

connectedDeviceAddress

```
public java.lang.String connectedDeviceAddress
```

Constructors

MultiplexBaseTransport(Handler
handler, TransportType
transportType)

```
protected MultiplexBaseTransport(Handler handler,TransportType  
transportType)
```

Parameters

handler

transportType

Methods

setState(int state)

```
protected synchronized void setState(int state)
```

Parameters

state

getAddress()

```
public java.lang.String getAddress()
```

getDeviceName()

```
public java.lang.String getDeviceName()
```

getTransportRecord()

Should only be called after a connection has been established

```
public TransportRecord getTransportRecord()
```

getState()

Return the current connection state.

```
public synchronized int getState()
```

isConnected()

```
public boolean isConnected()
```

stop()

```
public synchronized void stop()
```

stop(int state)

```
protected abstract void stop(int state)
```

Parameters

state

write(byte[] out, int offset, int count)

```
public abstract void write(byte[] out,int offset,int count)
```

Parameters

out
offset
count

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class

MultiplexBluetoothTransport

Hierarchy

java.lang.Object

[com.smartdevicelink.transport.MultiplexBaseTransport](#)

Package

com.smartdevicelink.transport

Overview

This class does all the work for setting up and managing Bluetooth connections with other devices. It has a thread that listens for incoming connections, a thread for connecting with a device, and a thread for performing data transmissions when connected.

Quick Jump

Fields

- [SHARED_PREFS](#)

Constructors

- `MultiplexBluetoothTransport( Handler handler )`

Methods

- `getBluetoothSerialServerInstance(Handler handler)` **Deprecated**
- `getBluetoothSerialServerInstance(Handler handler,boolean keepSocketAlive)` **Deprecated**
- `getBluetoothSerialServerInstance()` **Deprecated**
- `setStateManually(int state)`
- `getState()`
- `setKeepSocketAlive(boolean keepSocketAlive)`
- `start()`
- `connect(BluetoothDevice device)`
- `connected(BluetoothSocket socket,BluetoothDevice device)`
- `stop()`
- `stop(int stateToTransitionTo)`
- `write(byte[] out,int offset,int count)`
- `getBTSocket(BluetoothServerSocket bsSocket)`
- `getChannel(BluetoothSocket bsSocket)`

Fields

SHARED_PREFS

```
protected static final java.lang.String SHARED_PREFS
```

Constant Value

"sdl.bluetoothprefs"

Inherited Fields

From Class	Fields
<code>com.smartdevicelink.transport.MultiplexBaseTransport</code>	<code>STATE_NONE</code> , <code>STATE_LISTEN</code> , <code>STATE_CONNECTING</code> , <code>STATE_CONNECTED</code> , <code>STATE_ERROR</code> , <code>LOG</code> , <code>DEVICE_NAME</code> , <code>DEVICE_ADDRESS</code> , <code>mState</code> , <code>handler</code> , <code>transportType</code> , <code>transportRecord</code> , <code>currentlyConnectedDevice</code> , <code>connectedDeviceName</code> , <code>connectedDeviceAddress</code>

Constructors

MultiplexBluetoothTransport(Handler handler)

Constructor. Prepares a new BluetoothChat session.

```
public MultiplexBluetoothTransport(Handler handler)
```

Parameters

handler

A Handler to send messages back to the UI Activity

Methods

getBluetoothSerialServerInstance (Handler handler)

Deprecated

This method has been deprecated. It will return an instance of MultiplexBluetoothTransport but **should not** be used.

```
public static synchronized com.smartdevicelink.transport.  
MultiplexBluetoothTransport getBluetoothSerialServerInstance(Handler  
handler)
```

Parameters

handler

for receiving status messages from the transport

getBluetoothSerialServerInstance (Handler handler, boolean keepSocketAlive)

Deprecated

This method has been deprecated. It will return an instance of MultiplexBluetoothTransport but **should not** be used.

```
public static synchronized com.smartdevicelink.transport.  
MultiplexBluetoothTransport getBluetoothSerialServerInstance(Handler  
handler,boolean keepSocketAlive)
```

Parameters

handler

for receiving status messages from the transport

keepSocketAlive

Flag for keeping the socket alive

getBluetoothSerialServerInstance()

Deprecated

This method has been deprecated. **It will always return null.**

```
public static synchronized com.smartdevicelink.transport.  
MultiplexBluetoothTransport getBluetoothSerialServerInstance()
```

setStateManually(int state)

```
protected synchronized void setStateManually(int state)
```

Parameters

state

getState()

Return the current connection state.

```
public synchronized int getState()
```

setKeepSocketAlive(boolean keepSocketAlive)

```
public void setKeepSocketAlive(boolean keepSocketAlive)
```

Parameters

keepSocketAlive

start()

Start the chat service. Specifically start AcceptThread to begin a session in listening (server) mode. Called by the Activity onResume()

```
public synchronized void start()
```

connect(BluetoothDevice device)

Start the ConnectThread to initiate a connection to a remote device.

```
public synchronized void connect(BluetoothDevice device)
```

Parameters

device

The BluetoothDevice to connect

connected(BluetoothSocket socket, BluetoothDevice device)

Start the ConnectedThread to begin managing a Bluetooth connection

```
public synchronized void connected(BluetoothSocket socket,  
BluetoothDevice device)
```

Parameters

socket

The BluetoothSocket on which the connection was made

device

The BluetoothDevice that has been connected

stop()

Stop all threads

```
public synchronized void stop()
```

stop(int stateToTransitionTo)

```
protected synchronized void stop(int stateToTransitionTo)
```

Parameters

stateToTransitionTo

write(byte[] out, int offset, int count)

Write to the ConnectedThread in an unsynchronized manner

```
public void write(byte[] out,int offset,int count)
```

Parameters

out

The bytes to write

offset

count

getBTSocket(BluetoothServerSocket bsSocket)

```
public android.bluetooth.BluetoothSocket getBTSocket(  
    BluetoothServerSocket bsSocket)
```

Parameters

bsSocket

getChannel(BluetoothSocket
bsSocket)

```
public int getChannel(BluetoothSocket bsSocket)
```

Parameters

bsSocket

Inherited Methods

From Class	Methods
<code>com.smartdevicelink.transport.MultiplexBaseTransport</code>	<code>setState</code> , <code>getAddress</code> , <code>getDeviceName</code> , <code>getTransportRecord</code> , <code>getState</code> , <code>isConnected</code> , <code>stop</code> , <code>stop</code> , <code>write</code>
<code>java.lang.Object</code>	<code>getClass</code> , <code>hashCode</code> , <code>equals</code> , <code>clone</code> , <code>toString</code> , <code>notify</code> , <code>notifyAll</code> , <code>wait</code> , <code>wait</code> , <code>wait</code> , <code>finalize</code>

Class MultiplexTcpTransport

Hierarchy

`java.lang.Object`

`com.smartdevicelink.transport.MultiplexBaseTransport`

Package

`com.smartdevicelink.transport`

Quick Jump

Constructors

- `MultiplexTcpTransport( int port, String ipAddress, boolean autoReconnect, Handler handler, Context context )`

Methods

- `start()`
- `stop(int state)`
- `write(byte[] out,int offset,int count)`

Fields

Inherited Fields

From Class	Fields
<code>com.smartdevicelink.transport.MultiplexBaseTransport</code>	<code>STATE_NONE, STATE_LISTEN, STATE_CONNECTING, STATE_CONNECTED, STATE_ERROR, LOG, DEVICE_NAME, DEVICE_ADDRESS, mState, handler, transportType, transportRecord, currentlyConnectedDevice, connectedDeviceName, connectedDeviceAddress</code>

Constructors

MultiplexTcpTransport(int port, String ipAddress, boolean autoReconnect, Handler handler, Context context)

```
public MultiplexTcpTransport(int port,String ipAddress,boolean autoReconnect,Handler handler,Context context)
```

Parameters

port
ipAddress
autoReconnect
handler
context

Methods

start()

```
public synchronized void start()
```

stop(int state)

```
protected void stop(int state)
```

Parameters

state

write(byte[] out, int offset, int count)


```
public void write(byte[] out,int offset,int count)
```

Parameters

out

offset

count

Inherited Methods

From Class	Methods
com.smartdevicelink.transport.Multiplex BaseTransport	setState, getAddress, getDeviceName, g etTransportRecord, getState, isConnected , stop, stop, write
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class MultiplexTransport

Hierarchy

java.lang.Object

[com.smartdevicelink.transport.SdlTransport](#)

Package

[com.smartdevicelink.transport](#)

Quick Jump

Fields

- [isDisconnecting](#)

Constructors

- [MultiplexTransport\(MultiplexTransportConfig transportConfig, ITransportListener transportListener \)](#)

Methods

- [isDisconnecting\(\)](#)
- [getConfig\(\)](#)
- [requestNewSession\(\)](#)
- [removeSession\(long sessionId\)](#)
- [getTransportType\(\)](#)
- [getBroadcastComment\(\)](#)
- [sendBytesOverTransport\(SdlPacket packet\)](#)
- [openConnection\(\)](#)

- `disconnect()`
- `handleTransportError(String message,Exception ex)`
- `isPendingConnected()`

Fields

isDisconnecting

```
protected boolean isDisconnecting
```

Constructors

```
MultiplexTransport(  
    MultiplexTransportConfig  
    transportConfig,  
    ITransportListener  
    transportListener )
```

```
public MultiplexTransport(MultiplexTransportConfig transportConfig,  
    ITransportListener transportListener)
```

Parameters

transportConfig
transportListener

Methods

isDisconnecting()

```
public boolean isDisconnecting()
```

getConfig()

Returns the config that was used to create this transport

```
public com.smartdevicelink.transport.MultiplexTransportConfig  
getConfig()
```

requestNewSession()

```
public boolean requestNewSession()
```

removeSession(long sessionId)

```
public void removeSession(long sessionId)
```

Parameters

sessionId

getTransportType()

Overridden abstract method which returns specific type of this transport.

```
public TransportType getTransportType()
```

getBroadcastComment()

```
public java.lang.String getBroadcastComment()
```

sendBytesOverTransport(SdIPacket packet)

```
protected boolean sendBytesOverTransport(SdIPacket packet)
```

Parameters

packet

openConnection()

```
public void openConnection() throws SdIException
```

Throws

- `SdlException`

disconnect()

```
public void disconnect()
```

handleTransportError(String message, Exception ex)

```
protected void handleTransportError(String message,Exception ex)
```

Parameters

message
ex

isPendingConnected()

```
public boolean isPendingConnected()
```

Inherited Methods

From Class	Methods
com.smartdevicelink.transport.SdlTransport	getIsConnected, handleReceivedPacket, sendBytesOverTransport, sendBytes, handleTransportConnected, handleTransportDisconnected, handleTransportError, openConnection, disconnect, getTransportType, getBroadcastComment
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class

MultiplexTransport.TransportBroker

Hierarchy

java.lang.Object

java.lang.Thread

Package

com.smartdevicelink.transport

Overview

This thread will handle the broker transaction with the router service.

Quick Jump

Constructors

- [TransportBrokerThread\(Context context, String appId, ComponentName service \)](#)

Methods

- [startConnection\(\)](#)
- [cancel\(\)](#)
- [onHardwareConnected\(TransportType type\)](#)

- `sendPacket(SdlPacket packet)`
- `requestNewSession()`
- `removeSession(long sessionId)`
- `run()`
- `initTransportBroker()`

Fields

Inherited Fields

From Class	Fields
<code>java.lang.Thread</code>	<code>MIN_PRIORITY</code> , <code>NORM_PRIORITY</code> , <code>MAX_PRIORITY</code>

Constructors

`TransportBrokerThread(Context
context, String appld,
ComponentName service)`

Thread will automatically start to prepare its looper.

```
public TransportBrokerThread(Context context,String appld,  
ComponentName service)
```

Parameters

context
appId
service

Methods

startConnection()

```
public void startConnection()
```

cancel()

```
public synchronized void cancel()
```

onHardwareConnected(TransportType type)

```
public void onHardwareConnected(TransportType type)
```

Parameters

type

sendPacket(SdlPacket packet)

```
public void sendPacket(SdlPacket packet)
```

Parameters

packet

requestNewSession()

```
public void requestNewSession()
```

removeSession(long sessionId)

```
public void removeSession(long sessionId)
```

Parameters

sessionId

run()

```
public void run()
```

initTransportBroker()

```
public void initTransportBroker()
```

Inherited Methods

From Class	Methods
java.lang.Thread	currentThread, yield, sleep, sleep, clone, start, run, stop, stop, interrupt, interrupted, isInterrupted, destroy, isAlive, suspend, resume, setPriority, getPriority, setName, getName, getThreadGroup, activeCount, enumerate, countStackFrames, join, join, join, dumpStack, setDaemon, isDaemon, checkAccess, toString, getContextClassLoader, setContextClassLoader, holdsLock, getStackTrace, getAllStackTraces, getId, getState, setDefaultUncaughtExceptionHandler, getDefaultUncaughtExceptionHandler, getUncaughtExceptionHandler, setUncaughtExceptionHandler
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class

MultiplexTransportConfig

Hierarchy

java.lang.Object

BaseTransportConfig

Package

com.smartdevicelink.transport

Quick Jump

Fields

- [FLAG_MULTI_SECURITY_OFF](#)
- [FLAG_MULTI_SECURITY_LOW](#)
- [FLAG_MULTI_SECURITY_MED](#)
- [FLAG_MULTI_SECURITY_HIGH](#)

Constructors

- [MultiplexTransportConfig\(Context context, String appld \)](#)
- [MultiplexTransportConfig\(Context context, String appld, int securityLevel \)](#)

Methods

- [getTransportType\(\)](#)

- getContext()
- getService()
- setService(ComponentName service)
- setSecurityLevel(int securityLevel)
- getSecurityLevel()
- setRequiresHighBandwidth(boolean requiresHighBandwidth)
- requiresHighBandwidth()
- setPrimaryTransports(java.util.List<TransportType> transports)
- getPrimaryTransports()
- setSecondaryTransports(java.util.List<TransportType> transports)
- getSecondaryTransports()
- setTransportListener(MultiplexTransportConfig.TransportListener listener)
- getTransportListener()

Fields

FLAG_MULTI_SECURITY_OFF

```
public static final int FLAG_MULTI_SECURITY_OFF
```

FLAG_MULTI_SECURITY_LOW

```
public static final int FLAG_MULTI_SECURITY_LOW
```

Constant Value

FLAG_MULTI_SECURITY_MED

```
public static final int FLAG_MULTI_SECURITY_MED
```

Constant Value

32

FLAG_MULTI_SECURITY_HIGH

```
public static final int FLAG_MULTI_SECURITY_HIGH
```

Constant Value

48

Constructors

MultiplexTransportConfig(Context context, String appId)

```
public MultiplexTransportConfig(Context context,String appId)
```

Parameters

context
appId

MultiplexTransportConfig(Context context, String appId, int securityLevel)

```
public MultiplexTransportConfig(Context context,String appId,int  
securityLevel)
```

Parameters

context
appId
securityLevel

Methods

getTransportType()

Overridden abstract method which returns specific type of this transport configuration.

```
public TransportType getTransportType()
```

getContext()

Gets the context attached to this config

```
public android.content.Context getContext()
```

getService()

Gets the ComponentName of the router service attached to this config

```
public android.content.ComponentName getService()
```

setService(ComponentName service)

Supplies the config with the router service that should be bound to

```
public void setService(ComponentName service)
```

Parameters

service

the router service that should be bound to

setSecurityLevel(int securityLevel)

Sets the security level that should be used to verify a router service that is to be bound

```
public void setSecurityLevel(int securityLevel)
```

Parameters

securityLevel

the security level that will be used to perform certain tests

getSecurityLevel()

Get the security level that will be used to verify a router service before binding

```
public int getSecurityLevel()
```

setRequiresHighBandwidth(boolean requiresHighBandwidth)

Set whether or not this app requires the use of a transport that supports high bandwidth services. Common use is when an app uses the video/audio streaming services and there is no other integration that could be useful to the user.

For example:

1. If an app intends to perform audio or video streaming and does not wish to appear on the module when that isn't possible, a value of true should be sent.
2. If the same app wishes to appear on the module even when those services aren't available a value of true should be sent. In this case, the app could display a message prompting the user to "Please connect USB or Wifi" or it could have a separate integration like giving turn by turn directions in place of streaming the full navigation map.

```
public void setRequiresHighBandwidth(boolean requiresHighBandwidth)
```

Parameters

`requiresHighBandwidth`

whether the app should be treated as requiring a high bandwidth transport.

`requiresHighBandwidth()`

Get the setting from this config to see whether the app should be treated as requiring a high bandwidth transport.

```
public boolean requiresHighBandwidth()
```

`setPrimaryTransports( java.util.List<TransportType> transports )`

This will set the order in which a primary transport is determined to be accepted or not. In the case of previous protocol versions (< 5.1)

```
public void setPrimaryTransports(java.util.List<TransportType>  
transports)
```

Parameters

transports

list of transports that can be used as primary

getPrimaryTransports()

Get the list of primary transports that are set to be accepted by this config

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

setSecondaryTransports(java.util.List<TransportType> transports)

This will set the order in which a primary transport is determined to be accepted or not. In the case of previous protocol versions (< 5.1)

```
public void setSecondaryTransports(java.util.List<TransportType>  
transports)
```


Parameters

transports

list of transports that can be used as secondary

getSecondaryTransports()

Get the list of secondary transports that are set to be accepted by this config

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

setTransportListener(MultiplexTransportConfig.TransportListe listener)

Set a listener for transport events. Useful when connected high bandwidth services like audio or video streaming

```
public void setTransportListener(MultiplexTransportConfig.  
TransportListener listener)
```

Parameters

listener

the TransportListener that will be called back when transport events happen

getTransportListener()

Getter for the supplied transport listener

```
public com.smartdevicelink.transport.MultiplexTransportConfig.  
TransportListener getTransportListener()
```

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Interface MultiplexTransportConfig.Transport

Package

com.smartdevicelink.transport

Overview

Callback to be used if the state of the transports needs to be monitored for any reason.

Quick Jump

Methods

- [onTransportEvent\(java.util.List<TransportRecord> connectedTransports,boolean audioStreamTransportAvail,boolean videoStreamTransportAvail\)](#)

Methods

onTransportEvent(java.util.List<TransportRecord> connectedTransports, boolean audioStreamTransportAvail, boolean videoStreamTransportAvail)

Gets called whenever there is a change in the available transports for use

```
public void onTransportEvent(java.util.List<TransportRecord>  
connectedTransports,boolean audioStreamTransportAvail,boolean  
videoStreamTransportAvail)
```

Parameters

connectedTransports

the currently connected transports

audioStreamTransportAvail

true if there is either an audio streaming supported transport currently connected or a transport is available to connect with. False if there is no transport connected to support audio streaming and no possibility in the foreseeable future.

videoStreamTransportAvail

true if there is either a video streaming supported transport currently connected or a transport is available to connect with. False if there is no transport connected to support video streaming and no possibility in the foreseeable future.

Class MultiplexUsbTransport

Hierarchy

java.lang.Object

[com.smartdevicelink.transport.MultiplexBaseTransport](#)

Package

com.smartdevicelink.transport

Quick Jump

Fields

- [MANUFACTURER](#)
- [MODEL](#)
- [VERSION](#)
- [URI](#)
- [SERIAL](#)
- [DESCRIPTION](#)

Methods

- `start()`
- `stop(int stateToTransitionTo)`
- `write(byte[] out,int offset,int count)`

Fields

MANUFACTURER

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

Constant Value

"manufacturer"

MODEL

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

Constant Value

"model"

VERSION

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

Constant Value

"version"

URI

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

Constant Value

"uri"

SERIAL

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

Constant Value

"serial"

DESCRIPTION

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

Constant Value

"description"

Inherited Fields

From Class	Fields
com.smartdevicelink.transport.MultiplexBaseTransport	STATE_NONE, STATE_LISTEN, STATE_CONNECTING, STATE_CONNECTED, STATE_ERROR, LOG, DEVICE_NAME, DEVICE_ADDRESS, mState, handler, transportType, transportRecord, currentlyConnectedDevice, connectedDeviceName, connectedDeviceAddress

Methods

start()

```
public synchronized void start()
```

stop(int stateToTransitionTo)

```
protected synchronized void stop(int stateToTransitionTo)
```

Parameters

stateToTransitionTo

write(byte[] out, int offset, int count)

Write to the ConnectedThread in an unsynchronized manner

```
public void write(byte[] out,int offset,int count)
```

Parameters

out

The bytes to write

offset

count

Inherited Methods

From Class	Methods
<code>com.smartdevicelink.transport.MultiplexBaseTransport</code>	<code>setState</code> , <code>getAddress</code> , <code>getDeviceName</code> , <code>getTransportRecord</code> , <code>getState</code> , <code>isConnected</code> , <code>stop</code> , <code>stop</code> , <code>write</code>
<code>java.lang.Object</code>	<code>getClass</code> , <code>hashCode</code> , <code>equals</code> , <code>clone</code> , <code>toString</code> , <code>notify</code> , <code>notifyAll</code> , <code>wait</code> , <code>wait</code> , <code>wait</code> , <code>finalize</code>

Class PermissionManager

Hierarchy

`java.lang.Object`

`BasePermissionManager`

Package

`com.smartdevicelink.managers.permission`

Overview

PermissionManager gives the developer information about what permissions are permitted in specific HMI level and helps developers setup listeners to be

called when specific permissions become allowed.

This should be used through the `com.smartdevicelink.managers.SdlManager` and not be instantiated by itself

Quick Jump

Constructors

- `PermissionManager( ISdl internalInterface )`

Constructors

PermissionManager(ISdl internalInterface)

Creates a new instance of the PermissionManager

```
public PermissionManager(ISdl internalInterface)
```

Parameters

internalInterface

Methods

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class ProxyBridge

Hierarchy

java.lang.Object

Package

com.smartdevicelink.managers

Quick Jump

Constructors

- `ProxyBridge( ProxyBridge.LifecycleListener lifecycleListener )`

Methods

- `onProxyOpened()`
- `onRegisterAppInterfaceResponse(RegisterAppInterfaceResponse response)`
- `onOnAppInterfaceUnregistered(OnAppInterfaceUnregistered notification)`
- `onUnregisterAppInterfaceResponse(UnregisterAppInterfaceResponse response)`
- `onRPCReceived(RPCMessage message)`
- `addRpcListener(FunctionID id,OnRPCListener listener)`
- `removeOnRPCListener(FunctionID id,OnRPCListener listener)`
- `onProxyClosed(String info,Exception e,SdlIDisconnectedReason reason)`
- `onServiceEnded(OnServiceEnded serviceEnded)`
- `onServiceNACKed(OnServiceNACKed serviceNACKed)`
- `onError(String info,Exception e)`
- `onOnStreamRPC(OnStreamRPC notification)`
- `onStreamRPCResponse(StreamRPCResponse response)`
- `onOnHMISStatus(OnHMISStatus notification)`
- `onGenericResponse(GenericResponse response)`
- `onOnCommand(OnCommand notification)`
- `onAddCommandResponse(AddCommandResponse response)`
- `onAddSubMenuResponse(AddSubMenuResponse response)`
- `onCreateInteractionChoiceSetResponse(CreateInteractionChoiceSetResponse response)`
- `onAlertResponse(AlertResponse response)`
- `onDeleteCommandResponse>DeleteCommandResponse response)`
- `onDeleteInteractionChoiceSetResponse>DeleteInteractionChoiceSetResponse response)`
- `onDeleteSubMenuResponse>DeleteSubMenuResponse response)`
- `onPerformInteractionResponse(PerformInteractionResponse response)`
- `onResetGlobalPropertiesResponse(ResetGlobalPropertiesResponse response)`
- `onSetGlobalPropertiesResponse(SetGlobalPropertiesResponse response)`
- `onSetMediaClockTimerResponse(SetMediaClockTimerResponse response)`

- `onShowResponse`(`ShowResponse` response)
- `onSpeakResponse`(`SpeakResponse` response)
- `onOnButtonEvent`(`OnButtonEvent` notification)
- `onOnButtonPress`(`OnButtonPress` notification)
- `onSubscribeButtonResponse`(`SubscribeButtonResponse` response)
- `onUnsubscribeButtonResponse`(`UnsubscribeButtonResponse` response)
- `onOnPermissionsChange`(`OnPermissionsChange` notification)
- `onSubscribeVehicleDataResponse`(`SubscribeVehicleDataResponse` response)
- `onUnsubscribeVehicleDataResponse`(`UnsubscribeVehicleDataResponse` response)
- `onGetVehicleDataResponse`(`GetVehicleDataResponse` response)
- `onOnVehicleData`(`OnVehicleData` notification)
- `onPerformAudioPassThruResponse`(`PerformAudioPassThruResponse` response)
- `onEndAudioPassThruResponse`(`EndAudioPassThruResponse` response)
- `onOnAudioPassThru`(`OnAudioPassThru` notification)
- `onPutFileResponse`(`PutFileResponse` response)
- `onDeleteFileResponse`(`DeleteFileResponse` response)
- `onListFilesResponse`(`ListFilesResponse` response)
- `onSetApplIconResponse`(`SetApplIconResponse` response)
- `onScrollableMessageResponse`(`ScrollableMessageResponse` response)
- `onChangeRegistrationResponse`(`ChangeRegistrationResponse` response)
- `onSetDisplayLayoutResponse`(`SetDisplayLayoutResponse` response)
- `onOnLanguageChange`(`OnLanguageChange` notification)
- `onOnHashChange`(`OnHashChange` notification)
- `onSliderResponse`(`SliderResponse` response)
- `onOnDriverDistraction`(`OnDriverDistraction` notification)
- `onOnTBTClientState`(`OnTBTClientState` notification)
- `onOnSystemRequest`(`OnSystemRequest` notification)
- `onSystemRequestResponse`(`SystemRequestResponse` response)
- `onOnKeyboardInput`(`OnKeyboardInput` notification)
- `onOnTouchEvent`(`OnTouchEvent` notification)
- `onDiagnosticMessageResponse`(`DiagnosticMessageResponse` response)
- `onReadDIDResponse`(`ReadDIDResponse` response)
- `onGetDTCsResponse`(`GetDTCsResponse` response)
- `onOnLockScreenNotification`(`OnLockScreenStatus` notification)
- `onDialNumberResponse`(`DialNumberResponse` response)
- `onSendLocationResponse`(`SendLocationResponse` response)
- `onShowConstantTbtResponse`(`ShowConstantTbtResponse` response)
- `onAlertManeuverResponse`(`AlertManeuverResponse` response)
- `onUpdateTurnListResponse`(`UpdateTurnListResponse` response)
- `onServiceDataACK`(int dataSize)
- `onGetWayPointsResponse`(`GetWayPointsResponse` response)
- `onSubscribeWayPointsResponse`(`SubscribeWayPointsResponse` response)
- `onUnsubscribeWayPointsResponse`(`UnsubscribeWayPointsResponse` response)

- `onOnWayPointChange(OnWayPointChange notification)`
- `onGetSystemCapabilityResponse(GetSystemCapabilityResponse response)`
- `onGetInteriorVehicleDataResponse (GetInteriorVehicleDataResponse response)`
- `onButtonPressResponse(ButtonPressResponse response)`
- `onSetInteriorVehicleDataResponse (SetInteriorVehicleDataResponse response)`
- `onOnInteriorVehicleData(OnInteriorVehicleData notification)`
- `onSendHapticDataResponse(SendHapticDataResponse response)`
- `onOnRCStatus(OnRCStatus notification)`
- `onSetCloudAppProperties(SetCloudAppPropertiesResponse response)`
- `onGetCloudAppProperties(GetCloudAppPropertiesResponse response)`
- `onPublishAppServiceResponse(PublishAppServiceResponse response)`
- `onGetAppServiceDataResponse(GetAppServiceDataResponse response)`
- `onGetFileResponse(GetFileResponse response)`
- `onPerformAppServiceInteractionResponse (PerformAppServiceInteractionResponse response)`
- `onOnAppServiceData(OnAppServiceData notification)`
- `onOnSystemCapabilityUpdated(OnSystemCapabilityUpdated notification)`

Constructors

ProxyBridge(
ProxyBridge.LifecycleListener
lifecycleListener)

```
public ProxyBridge(ProxyBridge.LifecycleListener lifecycleListener)
```


Parameters

lifecycleListener

Methods

onProxyOpened()

```
public void onProxyOpened()
```

onRegisterAppInterfaceResponse (RegisterAppInterfaceResponse response)

```
public void onRegisterAppInterfaceResponse(  
RegisterAppInterfaceResponse response)
```

Parameters

response

onOnAppInterfaceUnregistered(OnAppInterfaceUnregistered notification)

```
public void onOnAppInterfaceUnregistered(OnAppInterfaceUnregistered  
notification)
```

Parameters

notification

onUnregisterAppInterfaceResponse (UnregisterAppInterfaceResponse response)

```
public void onUnregisterAppInterfaceResponse(  
UnregisterAppInterfaceResponse response)
```

Parameters

response

onRPCReceived(RPCMessage message)

```
public boolean onRPCReceived(RPCMessage message)
```

Parameters

message

addRpcListener(FunctionID id,
OnRPCListener listener)

```
protected void addRpcListener(FunctionID id,OnRPCListener listener)
```

Parameters

id

listener

removeOnRPCListener(FunctionID id, OnRPCListener listener)

```
public boolean removeOnRPCListener(FunctionID id, OnRPCListener  
listener)
```

Parameters

id
listener

onProxyClosed(String info, Exception e, SdIDisconnectedReason reason)

```
public void onProxyClosed(String info,Exception e,  
SdIDisconnectedReason reason)
```

Parameters

info
e
reason

onServiceEnded(OnServiceEnded serviceEnded)

```
public void onServiceEnded(OnServiceEnded serviceEnded)
```

Parameters

serviceEnded

onServiceNACKed(OnServiceNACKed serviceNACKed)

```
public void onServiceNACKed(OnServiceNACKed serviceNACKed)
```

Parameters

serviceNACKed

onError(String info, Exception e)

```
public void onError(String info,Exception e)
```

Parameters

info
e

onOnStreamRPC(OnStreamRPC notification)

```
public void onOnStreamRPC(OnStreamRPC notification)
```

Parameters

notification

onStreamRPCResponse(StreamRPCResponse response)

```
public void onStreamRPCResponse(StreamRPCResponse response)
```

Parameters

response

onOnHMISStatus(OnHMISStatus notification)

```
public void onOnHMISStatus(OnHMISStatus notification)
```

Parameters

notification

onGenericResponse(
GenericResponse response)

```
public void onGenericResponse(GenericResponse response)
```

Parameters

response

onOnCommand(OnCommand
notification)

```
public void onOnCommand(OnCommand notification)
```

Parameters

notification

onAddCommandResponse(AddCommandResponse response)

```
public void onAddCommandResponse(AddCommandResponse response  
)
```

Parameters

response

onAddSubMenuResponse(
AddSubMenuResponse response
)

```
public void onAddSubMenuResponse(AddSubMenuResponse response)
```

Parameters

response

onCreateInteractionChoiceSetResponse(
CreateInteractionChoiceSetResponse
response)

```
public void onCreateInteractionChoiceSetResponse(  
CreateInteractionChoiceSetResponse response)
```

Parameters

response

onAlertResponse(AlertResponse
response)

```
public void onAlertResponse(AlertResponse response)
```

Parameters

response

onDeleteCommandResponse(
DeleteCommandResponse
response)

```
public void onDeleteCommandResponse(DeleteCommandResponse response)
```

Parameters

response

onDeleteInteractionChoiceSetResponse (DeleteInteractionChoiceSetResponse response)

```
public void onDeleteInteractionChoiceSetResponse(  
DeleteInteractionChoiceSetResponse response)
```

Parameters

response

onDeleteSubMenuResponse(DeleteSubMenuResponse response)

```
public void onDeleteSubMenuResponse>DeleteSubMenuResponse  
response)
```

Parameters

response

onPerformInteractionResponse(PerformInteractionResponse response)

```
public void onPerformInteractionResponse(PerformInteractionResponse  
response)
```

Parameters

response

onResetGlobalPropertiesResponse
(ResetGlobalPropertiesResponse
response)

```
public void onResetGlobalPropertiesResponse(  
ResetGlobalPropertiesResponse response)
```

Parameters

response

onSetGlobalPropertiesResponse(SetGlobalPropertiesResponse response)

```
public void onSetGlobalPropertiesResponse(  
SetGlobalPropertiesResponse response)
```

Parameters

response

onSetMediaClockTimerResponse(SetMediaClockTimerResponse response)

```
public void onSetMediaClockTimerResponse(  
SetMediaClockTimerResponse response)
```

Parameters

response

onShowResponse(
ShowResponse response)

```
public void onShowResponse>ShowResponse response)
```

Parameters

response

onSpeakResponse(
SpeakResponse response)

```
public void onSpeakResponse(SpeakResponse response)
```

Parameters

response

onOnButtonEvent(OnButtonEvent notification)

```
public void onOnButtonEvent(OnButtonEvent notification)
```

Parameters

notification

onOnButtonPress(OnButtonPress notification)

```
public void onOnButtonPress(OnButtonPress notification)
```

Parameters

notification

onSubscribeButtonResponse(SubscribeButtonResponse response)

```
public void onSubscribeButtonResponse(SubscribeButtonResponse  
response)
```

Parameters

response

onUnsubscribeButtonResponse(
UnsubscribeButtonResponse
response)

```
public void onUnsubscribeButtonResponse(UnsubscribeButtonResponse  
response)
```

Parameters

response

onOnPermissionsChange(OnPermissionsChange notification)

```
public void onOnPermissionsChange(OnPermissionsChange notification)
```

Parameters

notification

onSubscribeVehicleDataResponse (SubscribeVehicleDataResponse response)

```
public void onSubscribeVehicleDataResponse(  
SubscribeVehicleDataResponse response)
```

Parameters

response

onUnsubscribeVehicleDataResponse
(
UnsubscribeVehicleDataResponse
response)

```
public void onUnsubscribeVehicleDataResponse(  
UnsubscribeVehicleDataResponse response)
```

Parameters

response

onGetVehicleDataResponse(GetVehicleDataResponse response)

```
public void onGetVehicleDataResponse(GetVehicleDataResponse  
response)
```

Parameters

response

onOnVehicleData(OnVehicleData notification)

```
public void onOnVehicleData(OnVehicleData notification)
```


Parameters

notification

onPerformAudioPassThruResponse
(
PerformAudioPassThruResponse
response)

```
public void onPerformAudioPassThruResponse(  
    PerformAudioPassThruResponse response)
```

Parameters

response

onEndAudioPassThruResponse(EndAudioPassThruResponse response)

```
public void onEndAudioPassThruResponse(EndAudioPassThruResponse  
response)
```

Parameters

response

onOnAudioPassThru(OnAudioPassThru notification)

```
public void onOnAudioPassThru(OnAudioPassThru notification)
```

Parameters

notification

onPutFileResponse(PutFileResponse response)

```
public void onPutFileResponse(PutFileResponse response)
```

Parameters

response

onDeleteFileResponse(DeleteFileResponse response)

```
public void onDeleteFileResponse(DeleteFileResponse response)
```

Parameters

response

onListFilesResponse(ListFilesResponse response)

```
public void onListFilesResponse(ListFilesResponse response)
```

Parameters

response

onSetAppIconResponse(SetAppIconResponse response)

```
public void onSetAppIconResponse(SetAppIconResponse response)
```

Parameters

response

onScrollableMessageResponse(ScrollableMessageResponse response)

```
public void onScrollableMessageResponse(ScrollableMessageResponse  
response)
```

Parameters

response

onChangeRegistrationResponse(
ChangeRegistrationResponse
response)

```
public void onChangeRegistrationResponse(  
ChangeRegistrationResponse response)
```

Parameters

response

onSetDisplayLayoutResponse(SetDisplayLayoutResponse response)

```
public void onSetDisplayLayoutResponse(SetDisplayLayoutResponse  
response)
```

Parameters

response

onOnLanguageChange(OnLanguageChange notification)

```
public void onOnLanguageChange(OnLanguageChange notification)
```

Parameters

notification

onOnHashChange(
OnHashChange notification)

```
public void onOnHashChange(OnHashChange notification)
```

Parameters

notification

onSliderResponse(
SliderResponse response)


```
public void onSliderResponse(SliderResponse response)
```

Parameters

response

onOnDriverDistraction(OnDriverDistraction notification)

```
public void onOnDriverDistraction(OnDriverDistraction notification)
```

Parameters

notification

onOnTBTCClientState(OnTBTCClientState notification)

```
public void onOnTBTCClientState(OnTBTCClientState notification)
```

Parameters

notification

onOnSystemRequest(OnSystemRequest notification)

```
public void onOnSystemRequest(OnSystemRequest notification)
```

Parameters

notification

onSystemRequestResponse(
SystemRequestResponse
response)

```
public void onSystemRequestResponse(SystemRequestResponse  
response)
```

Parameters

response

onOnKeyboardInput(
OnKeyboardInput notification)

```
public void onOnKeyboardInput(OnKeyboardInput notification)
```

Parameters

notification

onOnTouchEvent(OnTouchEvent notification)

```
public void onOnTouchEvent(OnTouchEvent notification)
```

Parameters

notification

onDiagnosticMessageResponse(DiagnosticMessageResponse response)

```
public void onDiagnosticMessageResponse(  
DiagnosticMessageResponse response)
```

Parameters

response

onReadDIDResponse(ReadDIDResponse response)

```
public void onReadDIDResponse(ReadDIDResponse response)
```

Parameters

response

onGetDTCsResponse(
GetDTCsResponse response)

```
public void onGetDTCsResponse(GetDTCsResponse response)
```

Parameters

response

onOnLockScreenNotification(
OnLockScreenStatus notification
)

```
public void onOnLockScreenNotification(OnLockScreenStatus  
notification)
```

Parameters

notification

onDialNumberResponse(DialNumberResponse response)

```
public void onDialNumberResponse(DialNumberResponse response)
```

Parameters

response

onSendLocationResponse(SendLocationResponse response)

```
public void onSendLocationResponse(SendLocationResponse response)
```

Parameters

response

onShowConstantTbtResponse(ShowConstantTbtResponse response)

```
public void onShowConstantTbtResponse(ShowConstantTbtResponse  
response)
```


Parameters

response

onAlertManeuverResponse(
AlertManeuverResponse
response)

```
public void onAlertManeuverResponse(AlertManeuverResponse  
response)
```

Parameters

response

onUpdateTurnListResponse(UpdateTurnListResponse response)

```
public void onUpdateTurnListResponse(UpdateTurnListResponse  
response)
```

Parameters

response

onServiceDataACK(int dataSize)

```
public void onServiceDataACK(int dataSize)
```

Parameters

dataSize

```
onGetWayPointsResponse(  
GetWayPointsResponse response  
)
```

```
public void onGetWayPointsResponse(GetWayPointsResponse response)
```

Parameters

response

```
onSubscribeWayPointsResponse(  
SubscribeWayPointsResponse  
response )
```

```
public void onSubscribeWayPointsResponse(  
SubscribeWayPointsResponse response)
```

Parameters

response

onUnsubscribeWayPointsResponse
(
UnsubscribeWayPointsResponse
response)

```
public void onUnsubscribeWayPointsResponse(  
UnsubscribeWayPointsResponse response)
```

Parameters

response

onOnWayPointChange(OnWayPointChange notification)

```
public void onOnWayPointChange(OnWayPointChange notification)
```

Parameters

notification

onGetSystemCapabilityResponse (GetSystemCapabilityResponse response)

```
public void onGetSystemCapabilityResponse(  
GetSystemCapabilityResponse response)
```

Parameters

response

onGetInteriorVehicleDataResponse
(
GetInteriorVehicleDataResponse
response)

```
public void onGetInteriorVehicleDataResponse(  
GetInteriorVehicleDataResponse response)
```

Parameters

response

onButtonPressResponse(ButtonPressResponse response)

```
public void onButtonPressResponse(ButtonPressResponse response)
```

Parameters

response

onSetInteriorVehicleDataResponse (SetInteriorVehicleDataResponse response)

```
public void onSetInteriorVehicleDataResponse(  
SetInteriorVehicleDataResponse response)
```

Parameters

response

onOnInteriorVehicleData(
OnInteriorVehicleData
notification)

```
public void onOnInteriorVehicleData(OnInteriorVehicleData notification)
```

Parameters

notification

onSendHapticDataResponse(
SendHapticDataResponse
response)


```
public void onSendHapticDataResponse(SendHapticDataResponse response)
```

Parameters

response

onOnRCStatus(OnRCStatus notification)

```
public void onOnRCStatus(OnRCStatus notification)
```

Parameters

notification

onSetCloudAppProperties(SetCloudAppPropertiesResponse response)

```
public void onSetCloudAppProperties(SetCloudAppPropertiesResponse  
response)
```

Parameters

response

onGetCloudAppProperties(GetCloudAppPropertiesResponse response)

```
public void onGetCloudAppProperties(GetCloudAppPropertiesResponse  
response)
```

Parameters

response

onPublishAppServiceResponse(
PublishAppServiceResponse
response)

```
public void onPublishAppServiceResponse(PublishAppServiceResponse  
response)
```

Parameters

response

onGetAppServiceDataResponse(GetAppServiceDataResponse response)

```
public void onGetAppServiceDataResponse(  
GetAppServiceDataResponse response)
```

Parameters

response

onGetFileResponse(GetFileResponse response)

```
public void onGetFileResponse(GetFileResponse response)
```

Parameters

response

onPerformAppServiceInteractionResponse
(
PerformAppServiceInteractionResponse
response)

```
public void onPerformAppServiceInteractionResponse(  
    PerformAppServiceInteractionResponse response)
```

Parameters

response

onOnAppServiceData(OnAppServiceData notification)

```
public void onOnAppServiceData(OnAppServiceData notification)
```

Parameters

notification

onOnSystemCapabilityUpdated(OnSystemCapabilityUpdated notification)

```
public void onOnSystemCapabilityUpdated(OnSystemCapabilityUpdated  
notification)
```

Parameters

notification

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Interface ProxyBridge.LifecycleListener

Package

com.smartdevicelink.managers

Quick Jump

Methods

- `onProxyConnected()`
- `onProxyClosed(String info,Exception e,SdIDisconnectedReason reason)`
- `onServiceEnded(OnServiceEnded serviceEnded)`
- `onServiceNACKed(OnServiceNACKed serviceNACKed)`
- `onError(String info,Exception e)`

Methods

`onProxyConnected()`

```
public void onProxyConnected()
```

`onProxyClosed( String info, Exception e, SdIDisconnectedReason reason )`

```
public void onProxyClosed(String info,Exception e,  
SdIDisconnectedReason reason)
```


Parameters

info
e
reason

onServiceEnded(OnServiceEnded serviceEnded)

```
public void onServiceEnded(OnServiceEnded serviceEnded)
```

Parameters

serviceEnded

onServiceNACKed(OnServiceNACKed serviceNACKed)

```
public void onServiceNACKed(OnServiceNACKed serviceNACKed)
```

Parameters

serviceNACKed

onError(String info, Exception e)

```
public void onError(String info,Exception e)
```

Parameters

info

e

Class RPCRequestFactory

Hierarchy

java.lang.Object

Package

com.smartdevicelink.proxy

Quick Jump

Fields

- [SDL_MSG_MAJOR_VERSION](#)
- [SDL_MSG_MINOR_VERSION](#)

Constructors

- `RPCRequestFactory()`

Methods

- `buildSystemRequest(String data,Integer correlationID)`
- `buildSystemRequestLegacy(java.util.Vector<java.lang.String> data,Integer correlationID)`
- `buildAddCommand(Integer commandID,String menuText,Integer parentID,Integer position,java.util.Vector<java.lang.String> vrCommands,Image cmdIcon,Integer correlationID)`
- `buildAddCommand(Integer commandID,String menuText,Integer parentID,Integer position,java.util.Vector<java.lang.String> vrCommands,String IconValue,ImageType IconType,Integer correlationID)`
- `buildAddCommand(Integer commandID,String menuText,Integer parentID,Integer position,java.util.Vector<java.lang.String> vrCommands,Integer correlationID)`
- `buildAddCommand(Integer commandID,String menuText,java.util.Vector<java.lang.String> vrCommands,Integer correlationID)`
- `buildAddCommand(Integer commandID,java.util.Vector<java.lang.String> vrCommands,Integer correlationID)`
- `buildAddSubMenu(Integer menuID,String menuName,Integer correlationID)`
- `buildAddSubMenu(Integer menuID,String menuName,Integer position,Integer correlationID)`
- `buildAlert(String ttsText,Boolean playTone,java.util.Vector<SoftButton> softButtons,Integer correlationID)`
- `buildAlert(String alertText1,String alertText2,String alertText3,Integer duration,java.util.Vector<SoftButton> softButtons,Integer correlationID)`
- `buildAlert(String ttsText,String alertText1,String alertText2,String alertText3,Boolean playTone,Integer duration,java.util.Vector<SoftButton> softButtons,Integer correlationID)`
- `buildAlert(java.util.Vector<TTSCChunk> chunks,Boolean playTone,java.util.Vector<SoftButton> softButtons,Integer correlationID)`
- `buildAlert(java.util.Vector<TTSCChunk> ttsChunks,String alertText1,String alertText2,String alertText3,Boolean playTone,Integer duration,java.util.Vector<SoftButton> softButtons,Integer correlationID)`
- `buildAlert(String ttsText,Boolean playTone,Integer correlationID)`
- `buildAlert(String alertText1,String alertText2,Integer duration,Integer correlationID)`
- `buildAlert(String ttsText,String alertText1,String alertText2,Boolean playTone,Integer duration,Integer correlationID)`

- buildAlert(java.util.Vector<TTSCChunk> chunks,Boolean playTone,Integer correlationID)
- buildAlert(java.util.Vector<TTSCChunk> ttsChunks,String alertText1,String alertText2,Boolean playTone,Integer duration,Integer correlationID)
- buildCreateInteractionChoiceSet(java.util.Vector<Choice> choiceSet,Integer interactionChoiceSetID,Integer correlationID)
- buildDeleteCommand(Integer commandID,Integer correlationID)
- buildDeleteFile(String sdlFileName,Integer correlationID)
- buildDeleteInteractionChoiceSet(Integer interactionChoiceSetID,Integer correlationID)
- buildDeleteSubMenu(Integer menuID,Integer correlationID)
- buildListFiles(Integer correlationID)
- buildPerformInteraction(java.util.Vector<TTSCChunk> initChunks,String displayText,java.util.Vector<java.lang.Integer> interactionChoiceSetIDList,java.util.Vector<TTSCChunk> helpChunks,java.util.Vector<TTSCChunk> timeoutChunks,InteractionMode interactionMode,Integer timeout,java.util.Vector<VrHelpItem> vrHelp,Integer correlationID)
- buildPerformInteraction(String initPrompt,String displayText,java.util.Vector<java.lang.Integer> interactionChoiceSetIDList,String helpPrompt,String timeoutPrompt,InteractionMode interactionMode,Integer timeout,java.util.Vector<VrHelpItem> vrHelp,Integer correlationID)
- buildPerformInteraction(String initPrompt,String displayText,Integer interactionChoiceSetID,String helpPrompt,String timeoutPrompt,InteractionMode interactionMode,Integer timeout,java.util.Vector<VrHelpItem> vrHelp,Integer correlationID)
- buildPerformInteraction(String initPrompt,String displayText,Integer interactionChoiceSetID,java.util.Vector<VrHelpItem> vrHelp,Integer correlationID)
- buildPerformInteraction(java.util.Vector<TTSCChunk> initChunks,String displayText,java.util.Vector<java.lang.Integer> interactionChoiceSetIDList,java.util.Vector<TTSCChunk> helpChunks,java.util.Vector<TTSCChunk> timeoutChunks,InteractionMode interactionMode,Integer timeout,Integer correlationID)
- buildPerformInteraction(String initPrompt,String displayText,java.util.Vector<java.lang.Integer> interactionChoiceSetIDList,String helpPrompt,String timeoutPrompt,InteractionMode interactionMode,Integer timeout,Integer correlationID)
- buildPerformInteraction(String initPrompt,String displayText,Integer interactionChoiceSetID,String helpPrompt,String timeoutPrompt,InteractionMode interactionMode,Integer timeout,Integer correlationID)
- buildPerformInteraction(String initPrompt,String displayText,Integer interactionChoiceSetID,Integer correlationID)
- buildPerformInteraction(java.util.Vector<TTSCChunk> initChunks,String displayText,java.util.Vector<java.lang.Integer> interactionChoiceSetIDList,java.util.Vector<TTSCChunk>

- helpChunks,InteractionMode interactionMode,Integer timeout,Integer correlationID) **Deprecated**
- buildPerformInteraction(String initPrompt,String displayText,java.util.Vector<java.lang.Integer> interactionChoiceSetIDList,String helpPrompt,InteractionMode interactionMode,Integer timeout,Integer correlationID) **Deprecated**
- buildPutFile(String sdlFileName,FileType fileType,Boolean persistentFile,byte[] fileData,Integer correlationID)
- buildPutFile(String sdlFileName,Integer iOffset,Integer iLength) **Deprecated**
- buildPutFile(String sdlFileName,Long iOffset,Long iLength)
- buildPutFile(String syncFileName,Integer iOffset,Integer iLength,FileType fileType,Boolean bPersistentFile,Boolean bSystemFile) **Deprecated**
- buildPutFile(String syncFileName,Long iOffset,Long iLength,FileType fileType,Boolean bPersistentFile,Boolean bSystemFile)
- buildPutFile(String sdlFileName,Integer iOffset,Integer iLength,FileType fileType,Boolean bPersistentFile,Boolean bSystemFile,Integer iCorrelationID) **Deprecated**
- buildPutFile(String sdlFileName,Long iOffset,Long iLength,FileType fileType,Boolean bPersistentFile,Boolean bSystemFile,Boolean isPayloadProtected,Integer iCorrelationID)
- buildRegisterAppInterface(String appName,String appID)
- buildRegisterAppInterface(String appName,Boolean isMediaApp,String appID)
- buildRegisterAppInterface(SdlMsgVersion sdlMsgVersion,String appName,java.util.Vector<TTSCChunk> ttsName,String ngnMediaScreenAppName,java.util.Vector<java.lang.String> vrSynonyms,Boolean isMediaApp,Language languageDesired,Language hmiDisplayLanguageDesired,java.util.Vector<AppHMitype> appType,String appID,Integer correlationID,DeviceInfo deviceInfo)
- buildSetAppIcon(String sdlFileName,Integer correlationID)
- buildSetGlobalProperties(String helpPrompt,String timeoutPrompt,Integer correlationID)
- buildSetGlobalProperties(java.util.Vector<TTSCChunk> helpChunks,java.util.Vector<TTSCChunk> timeoutChunks,Integer correlationID)
- buildSetGlobalProperties(String helpPrompt,String timeoutPrompt,String vrHelpTitle,java.util.Vector<VrHelpItem> vrHelp,Integer correlationID)
- buildSetGlobalProperties(java.util.Vector<TTSCChunk> helpChunks,java.util.Vector<TTSCChunk> timeoutChunks,String vrHelpTitle,java.util.Vector<VrHelpItem> vrHelp,Integer correlationID)
- buildSetMediaClockTimer(Integer hours,Integer minutes,Integer seconds,UpdateMode updateMode,Integer correlationID)
- buildSetMediaClockTimer(UpdateMode updateMode,Integer correlationID) **Deprecated**
- buildShow(String mainText1,String mainText2,String mainText3,String mainText4,String statusBar,String mediaClock,String mediaTrack,Image graphic,java.util.Vector<SoftButton>

- softButtons,java.util.Vector<java.lang.String>
customPresets,TextAlignment alignment,Integer correlationID)
- buildShow(String mainText1,String mainText2,String
mainText3,String mainText4,TextAlignment alignment,Integer
correlationID)
- buildShow(String mainText1,String mainText2,String
statusBar,String mediaClock,String mediaTrack,TextAlignment
alignment,Integer correlationID)
- buildShow(String mainText1,String mainText2,TextAlignment
alignment,Integer correlationID)
- buildSpeak(String ttsText,Integer correlationID)
- buildSpeak(java.util.Vector<TTSCChunk> ttsChunks,Integer
correlationID)
- buildSubscribeButton(ButtonName buttonName,Integer
correlationID)
- buildUnregisterAppInterface(Integer correlationID)
- buildUnsubscribeButton(ButtonName buttonName,Integer
correlationID)
- BuildSubscribeVehicleData(boolean gps,boolean speed,boolean
rpm,boolean fuelLevel,boolean fuelLevel_State,boolean
instantFuelConsumption,boolean externalTemperature,boolean
prndl,boolean tirePressure,boolean odometer,boolean
beltStatus,boolean bodyInformation,boolean deviceStatus,boolean
driverBraking,Integer correlationID)
- BuildUnsubscribeVehicleData(boolean gps,boolean speed,boolean
rpm,boolean fuelLevel,boolean fuelLevel_State,boolean
instantFuelConsumption,boolean externalTemperature,boolean
prndl,boolean tirePressure,boolean odometer,boolean
beltStatus,boolean bodyInformation,boolean deviceStatus,boolean
driverBraking,Integer correlationID)
- BuildGetVehicleData(boolean gps,boolean speed,boolean
rpm,boolean fuelLevel,boolean fuelLevel_State,boolean
instantFuelConsumption,boolean externalTemperature,boolean
vin,boolean prndl,boolean tirePressure,boolean odometer,boolean
beltStatus,boolean bodyInformation,boolean deviceStatus,boolean
driverBraking,Integer correlationID)
- BuildScrollableMessage(String scrollableMessageBody,Integer
timeout,java.util.Vector<SoftButton> softButtons,Integer
correlationID)
- BuildSlider(Integer numTicks,Integer position,String
sliderHeader,java.util.Vector<java.lang.String>
sliderFooter,Integer timeout,Integer correlationID)
- BuildChangeRegistration(Language language,Language
hmiDisplayLanguage,Integer correlationID)
- BuildSetDisplayLayout(String displayLayout,Integer correlationID)
- BuildPerformAudioPassThru(String ttsText,String
audioPassThruDisplayText1,String
audioPassThruDisplayText2,SamplingRate samplingRate,Integer
maxDuration,BitsPerSample bitsPerSample,AudioType
audioType,Boolean muteAudio,Integer correlationID)
- BuildPerformAudioPassThru(java.util.Vector<TTSCChunk>
initialPrompt,String audioPassThruDisplayText1,String
audioPassThruDisplayText2,SamplingRate samplingRate,Integer
maxDuration,BitsPerSample bitsPerSample,AudioType
audioType,Boolean muteAudio,Integer correlationID)

- `BuildEndAudioPassThru(Integer correlationID)`
- `BuildDeviceInfo(String carrierName)`

Fields

SDL_MSG_MAJOR_VERSION

```
public static final int SDL_MSG_MAJOR_VERSION
```

Constant Value

1

SDL_MSG_MINOR_VERSION

```
public static final int SDL_MSG_MINOR_VERSION
```

Constructors

RPCRequestFactory()

```
public RPCRequestFactory()
```

Methods

buildSystemRequest(String data, Integer correlationID)

```
public static SystemRequest buildSystemRequest(String data,Integer correlationID)
```

Parameters

data
correlationID

buildSystemRequestLegacy(java.util.Vector<java.lang.String> data, Integer correlationID)

```
public static SystemRequest buildSystemRequestLegacy(java.util.Vector  
<java.lang.String> data,Integer correlationID)
```

Parameters

data
correlationID

buildAddCommand(Integer
commandID, String menuText,
Integer parentID, Integer
position,
java.util.Vector<java.lang.String>
vrCommands, Image cmdIcon,
Integer correlationID)

```
public static AddCommand buildAddCommand(Integer commandID,  
String menuText,Integer parentID,Integer position,java.util.Vector<java.  
lang.String> vrCommands,Image cmdIcon,Integer correlationID)
```

Parameters

commandID
menuText
parentID

position
vrCommands
cmdIcon
correlationID

buildAddCommand(Integer
commandID, String menuText,
Integer parentID, Integer
position,
java.util.Vector<java.lang.String>
vrCommands, String IconValue,
ImageType IconType, Integer
correlationID)

```
public static AddCommand buildAddCommand(Integer commandID,  
String menuText,Integer parentID,Integer position,java.util.Vector<java.  
lang.String> vrCommands,String IconValue,ImageType IconType,Integer  
correlationID)
```

Parameters

commandID
menuText
parentID
position
vrCommands
IconValue
IconType
correlationID

buildAddCommand(Integer
commandID, String menuText,
Integer parentID, Integer
position,
java.util.Vector<java.lang.String>
vrCommands, Integer
correlationID)

```
public static AddCommand buildAddCommand(Integer commandID,  
String menuText,Integer parentID,Integer position,java.util.Vector<java.  
lang.String> vrCommands,Integer correlationID)
```

Parameters

commandID
menuText
parentID

position

vrCommands

correlationID

```
buildAddCommand( Integer  
commandID, String menuText,  
java.util.Vector<java.lang.String>  
vrCommands, Integer  
correlationID )
```

```
public static AddCommand buildAddCommand(Integer commandID,  
String menuText,java.util.Vector<java.lang.String> vrCommands,  
Integer correlationID)
```

Parameters

commandID

menuText
vrCommands
correlationID

buildAddCommand(Integer
commandID,
java.util.Vector<java.lang.String>
vrCommands, Integer
correlationID)

```
public static AddCommand buildAddCommand(Integer commandID,java  
.util.Vector<java.lang.String> vrCommands,Integer correlationID)
```

Parameters

commandID

vrCommands
correlationID

buildAddSubMenu(Integer
menuID, String menuName,
Integer correlationID)

```
public static AddSubMenu buildAddSubMenu(Integer menuID,String  
menuName,Integer correlationID)
```

Parameters

menuID
menuName
correlationID

buildAddSubMenu(Integer menuID, String menuName, Integer position, Integer correlationID)

```
public static AddSubMenu buildAddSubMenu(Integer menuID,String menuName,Integer position,Integer correlationID)
```

Parameters

menuID
menuName
position
correlationID

buildAlert(String ttsText,
Boolean playTone,
java.util.Vector<SoftButton>
softButtons, Integer correlationID
)

```
public static Alert buildAlert(String ttsText, Boolean playTone, java.util.  
Vector<SoftButton> softButtons, Integer correlationID)
```

Parameters

ttsText
playTone
softButtons
correlationID

buildAlert(String alertText1,
String alertText2, String
alertText3, Integer duration,
java.util.Vector<SoftButton>
softButtons, Integer correlationID
)

```
public static Alert buildAlert(String alertText1,String alertText2,String  
alertText3,Integer duration,java.util.Vector<SoftButton> softButtons,  
Integer correlationID)
```

Parameters

alertText1
alertText2
alertText3
duration

softButtons
correlationID

```
buildAlert( String ttsText, String  
alertText1, String alertText2,  
String alertText3, Boolean  
playTone, Integer duration,  
java.util.Vector<SoftButton>  
softButtons, Integer correlationID  
)
```

```
public static Alert buildAlert(String ttsText,String alertText1,String  
alertText2,String alertText3,Boolean playTone,Integer duration,java.util.  
Vector<SoftButton> softButtons,Integer correlationID)
```

Parameters

ttsText

alertText1
alertText2
alertText3
playTone
duration
softButtons
correlationID

buildAlert(
java.util.Vector<TTSCChunk>
chunks, Boolean playTone,
java.util.Vector<SoftButton>
softButtons, Integer correlationID
)

```
public static Alert buildAlert(java.util.Vector<TTSCChunk> chunks,  
Boolean playTone,java.util.Vector<SoftButton> softButtons,Integer  
correlationID)
```

Parameters

chunks
playTone
softButtons
correlationID

```
buildAlert(  
    java.util.Vector<TTSCChunk>  
    ttsChunks, String alertText1,  
    String alertText2, String  
    alertText3, Boolean playTone,  
    Integer duration,  
    java.util.Vector<SoftButton>  
    softButtons, Integer correlationID  
)
```

```
public static Alert buildAlert(java.util.Vector<TTSCChunk> ttsChunks,  
    String alertText1,String alertText2,String alertText3,Boolean playTone,  
    Integer duration,java.util.Vector<SoftButton> softButtons,Integer  
    correlationID)
```

Parameters

ttsChunks
alertText1

alertText2
alertText3
playTone
duration
softButtons
correlationID

buildAlert(String ttsText,
Boolean playTone, Integer
correlationID)

```
public static Alert buildAlert(String ttsText, Boolean playTone, Integer  
correlationID)
```

Parameters

ttsText

playTone

correlationID

**buildAlert(String alertText1,
String alertText2, Integer
duration, Integer correlationID)**

```
public static Alert buildAlert(String alertText1,String alertText2,Integer  
duration,Integer correlationID)
```

Parameters

alertText1

alertText2
duration
correlationID

buildAlert(String ttsText, String alertText1, String alertText2, Boolean playTone, Integer duration, Integer correlationID)

```
public static Alert buildAlert(String ttsText,String alertText1,String alertText2,Boolean playTone,Integer duration,Integer correlationID)
```

Parameters

ttsText

alertText1
alertText2
playTone
duration
correlationID

buildAlert(
java.util.Vector<TTSCChunk>
chunks, Boolean playTone,
Integer correlationID)

```
public static Alert buildAlert(java.util.Vector<TTSCChunk> chunks,  
Boolean playTone,Integer correlationID)
```

Parameters

chunks
playTone
correlationID

buildAlert(
java.util.Vector<TTSCChunk>
ttsChunks, String alertText1,
String alertText2, Boolean
playTone, Integer duration,
Integer correlationID)

```
public static Alert buildAlert(java.util.Vector<TTSCChunk> ttsChunks,  
String alertText1,String alertText2,Boolean playTone,Integer duration,  
Integer correlationID)
```

Parameters

ttsChunks
alertText1
alertText2
playTone
duration
correlationID

buildCreateInteractionChoiceSet(
java.util.Vector<Choice>
choiceSet, Integer
interactionChoiceSetID, Integer
correlationID)

```
public static CreateInteractionChoiceSet  
buildCreateInteractionChoiceSet(java.util.Vector<Choice> choiceSet,  
Integer interactionChoiceSetID,Integer correlationID)
```

Parameters

choiceSet
interactionChoiceSetID
correlationID

buildDeleteCommand(Integer commandID, Integer correlationID)

```
public static DeleteCommand buildDeleteCommand(Integer  
commandID,Integer correlationID)
```

Parameters

commandID
correlationID

buildDeleteFile(String sdlFileName, Integer correlationID)

```
public static DeleteFile buildDeleteFile(String sdlFileName,Integer  
correlationID)
```


Parameters

sdIFileName

correlationID

**buildDeleteInteractionChoiceSet(
Integer interactionChoiceSetID,
Integer correlationID)**

```
public static DeleteInteractionChoiceSet  
buildDeleteInteractionChoiceSet(Integer interactionChoiceSetID,Integer  
correlationID)
```

Parameters

interactionChoiceSetID

correlationID

buildDeleteSubMenu(Integer menuID, Integer correlationID)

```
public static DeleteSubMenu buildDeleteSubMenu(Integer menuID,  
Integer correlationID)
```

Parameters

menuID

correlationID

buildListFiles(Integer correlationID)

```
public static ListFiles buildListFiles(Integer correlationID)
```

Parameters

correlationID

```
buildPerformInteraction(  
    java.util.Vector<TTSCChunk>  
    initChunks, String displayText,  
    java.util.Vector<java.lang.Integer>  
    interactionChoiceSetIDList,  
    java.util.Vector<TTSCChunk>  
    helpChunks,  
    java.util.Vector<TTSCChunk>  
    timeoutChunks, InteractionMode  
    interactionMode, Integer  
    timeout,  
    java.util.Vector<VrHelpItem>  
    vrHelp, Integer correlationID )
```

```
public static PerformInteraction buildPerformInteraction(java.util.Vector<TTSCChunk> initChunks,String displayText,java.util.Vector<java.lang.Integer> interactionChoiceSetIDList,java.util.Vector<TTSCChunk> helpChunks,java.util.Vector<TTSCChunk> timeoutChunks,InteractionMode interactionMode,Integer timeout,java.util.Vector<VrHelpItem> vrHelp,Integer correlationID)
```

Parameters

initChunks
displayText
interactionChoiceSetIDList
helpChunks
timeoutChunks
interactionMode

timeout
vrHelp
correlationID

```
buildPerformInteraction( String  
initPrompt, String displayText,  
java.util.Vector<java.lang.Integer>  
interactionChoiceSetIDList,  
String helpPrompt, String  
timeoutPrompt, InteractionMode  
interactionMode, Integer  
timeout,  
java.util.Vector<VrHelpItem>  
vrHelp, Integer correlationID )
```

```
public static PerformInteraction buildPerformInteraction(String  
initPrompt,String displayText,java.util.Vector<java.lang.Integer>  
interactionChoiceSetIDList,String helpPrompt,String timeoutPrompt,  
InteractionMode interactionMode,Integer timeout,java.util.Vector<  
VrHelpItem> vrHelp,Integer correlationID)
```

Parameters

initPrompt

displayText

interactionChoiceSetIDList

helpPrompt

timeoutPrompt

interactionMode

timeout

vrHelp

correlationID

```
buildPerformInteraction( String  
initPrompt, String displayText,  
Integer interactionChoiceSetID,  
String helpPrompt, String  
timeoutPrompt, InteractionMode  
interactionMode, Integer  
timeout,  
java.util.Vector<VrHelpItem>  
vrHelp, Integer correlationID )
```

```
public static PerformInteraction buildPerformInteraction(String  
initPrompt,String displayText,Integer interactionChoiceSetID,String  
helpPrompt,String timeoutPrompt,InteractionMode interactionMode,  
Integer timeout,java.util.Vector<VrHelpItem> vrHelp,Integer  
correlationID)
```

Parameters

initPrompt

displayText
interactionChoiceSetID
helpPrompt
timeoutPrompt
interactionMode
timeout
vrHelp
correlationID

buildPerformInteraction(String
initPrompt, String displayText,
Integer interactionChoiceSetID,
java.util.Vector<VrHelpItem>
vrHelp, Integer correlationID)

```
public static PerformInteraction buildPerformInteraction(String  
initPrompt,String displayText,Integer interactionChoiceSetID,java.util.  
Vector<VrHelpItem> vrHelp,Integer correlationID)
```

Parameters

initPrompt
displayText
interactionChoiceSetID
vrHelp

correlationID

```
buildPerformInteraction(  
    java.util.Vector<TTSCChunk>  
    initChunks, String displayText,  
    java.util.Vector<java.lang.Integer>  
    interactionChoiceSetIDList,  
    java.util.Vector<TTSCChunk>  
    helpChunks,  
    java.util.Vector<TTSCChunk>  
    timeoutChunks, InteractionMode  
    interactionMode, Integer  
    timeout, Integer correlationID )
```

```
public static PerformInteraction buildPerformInteraction(java.util.Vector  
<TTSCChunk> initChunks,String displayText,java.util.Vector<java.lang.  
Integer> interactionChoiceSetIDList,java.util.Vector<TTSCChunk>  
helpChunks,java.util.Vector<TTSCChunk> timeoutChunks,  
InteractionMode interactionMode,Integer timeout,Integer correlationID)
```

Parameters

initChunks

displayText

interactionChoiceSetIDList

helpChunks

timeoutChunks

interactionMode

timeout

correlationID

```
buildPerformInteraction( String  
initPrompt, String displayText,  
java.util.Vector<java.lang.Integer>  
interactionChoiceSetIDList,  
String helpPrompt, String  
timeoutPrompt, InteractionMode  
interactionMode, Integer  
timeout, Integer correlationID )
```

```
public static PerformInteraction buildPerformInteraction(String  
initPrompt,String displayText,java.util.Vector<java.lang.Integer>  
interactionChoiceSetIDList,String helpPrompt,String timeoutPrompt,  
InteractionMode interactionMode,Integer timeout,Integer correlationID)
```

Parameters

initPrompt
displayText
interactionChoiceSetIDList

helpPrompt
timeoutPrompt
interactionMode
timeout
correlationID

```
buildPerformInteraction( String  
initPrompt, String displayText,  
Integer interactionChoiceSetID,  
String helpPrompt, String  
timeoutPrompt, InteractionMode  
interactionMode, Integer  
timeout, Integer correlationID )
```

```
public static PerformInteraction buildPerformInteraction(String  
initPrompt,String displayText,Integer interactionChoiceSetID,String  
helpPrompt,String timeoutPrompt,InteractionMode interactionMode,  
Integer timeout,Integer correlationID)
```

Parameters

initPrompt
displayText
interactionChoiceSetID
helpPrompt
timeoutPrompt
interactionMode
timeout

correlationID

buildPerformInteraction(String
initPrompt, String displayText,
Integer interactionChoiceSetID,
Integer correlationID)

```
public static PerformInteraction buildPerformInteraction(String  
initPrompt,String displayText,Integer interactionChoiceSetID,Integer  
correlationID)
```

Parameters

initPrompt
displayText
interactionChoiceSetID

correlationID

```
buildPerformInteraction(  
    java.util.Vector<TTSCChunk>  
    initChunks, String displayText,  
    java.util.Vector<java.lang.Integer>  
    interactionChoiceSetIDList,  
    java.util.Vector<TTSCChunk>  
    helpChunks, InteractionMode  
    interactionMode, Integer  
    timeout, Integer correlationID )
```

Deprecated

```
public static PerformInteraction buildPerformInteraction(java.util.Vector  
<TTSCChunk> initChunks,String displayText,java.util.Vector<java.lang.  
Integer> interactionChoiceSetIDList,java.util.Vector<TTSCChunk>  
helpChunks,InteractionMode interactionMode,Integer timeout,Integer  
correlationID)
```


Parameters

initChunks
displayText
interactionChoiceSetIDList
helpChunks
interactionMode
timeout
correlationID

buildPerformInteraction(String
initPrompt, String displayText,
java.util.Vector<java.lang.Integer>
interactionChoiceSetIDList,
String helpPrompt,
InteractionMode
interactionMode, Integer
timeout, Integer correlationID)

Deprecated

```
public static PerformInteraction buildPerformInteraction(String  
initPrompt,String displayText,java.util.Vector<java.lang.Integer>  
interactionChoiceSetIDList,String helpPrompt,InteractionMode  
interactionMode,Integer timeout,Integer correlationID)
```

Parameters

initPrompt
displayText

interactionChoiceSetIDList
helpPrompt
interactionMode
timeout
correlationID

buildPutFile(String sdlFileName,
FileType fileType, Boolean
persistentFile, byte[] fileData,
Integer correlationID)

```
public static PutFile buildPutFile(String sdlFileName,FileType fileType,  
Boolean persistentFile,byte[] fileData,Integer correlationID)
```

Parameters

sdlFileName
fileType
persistentFile
fileData
correlationID

buildPutFile(String sdlFileName,
Integer iOffset, Integer iLength)

Deprecated

```
public static PutFile buildPutFile(String sdlFileName,Integer iOffset,  
Integer iLength)
```

Parameters

sdIFileName
iOffset
iLength

**buildPutFile(String sdIFileName,
Long iOffset, Long iLength)**

```
public static PutFile buildPutFile(String sdIFileName,Long iOffset,Long iLength)
```

Parameters

sdIFileName

iOffset
iLength

buildPutFile(String
syncFileName, Integer iOffset,
Integer iLength, FileType fileType,
Boolean bPersistentFile, Boolean
bSystemFile)

Deprecated

```
public static PutFile buildPutFile(String syncFileName,Integer iOffset,  
Integer iLength,FileType fileType,Boolean bPersistentFile,Boolean  
bSystemFile)
```

Parameters

syncFileName

iOffset
iLength
fileType
bPersistentFile
bSystemFile

**buildPutFile(String
syncFileName, Long iOffset, Long
iLength, FileType fileType,
Boolean bPersistentFile, Boolean
bSystemFile)**

```
public static PutFile buildPutFile(String syncFileName, Long iOffset, Long  
iLength, FileType fileType, Boolean bPersistentFile, Boolean bSystemFile)
```

Parameters

syncFileName
iOffset
iLength
fileType
bPersistentFile
bSystemFile

buildPutFile(String sdlFileName,
Integer iOffset, Integer iLength,
FileType fileType, Boolean
bPersistentFile, Boolean
bSystemFile, Integer
iCorrelationID)

Deprecated

```
public static PutFile buildPutFile(String sdlFileName,Integer iOffset,  
Integer iLength,FileType fileType,Boolean bPersistentFile,Boolean  
bSystemFile,Integer iCorrelationID)
```

Parameters

sdlFileName
iOffset
iLength

fileType
bPersistentFile
bSystemFile
iCorrelationID

buildPutFile(String sdlFileName,
Long iOffset, Long iLength,
FileType fileType, Boolean
bPersistentFile, Boolean
bSystemFile, Boolean
isPayloadProtected, Integer
iCorrelationID)

```
public static PutFile buildPutFile(String sdlFileName,Long iOffset,Long  
iLength,FileType fileType,Boolean bPersistentFile,Boolean bSystemFile,  
Boolean isPayloadProtected,Integer iCorrelationID)
```

Parameters

sdIFileName
iOffset
iLength
fileType
bPersistentFile
bSystemFile
isPayloadProtected
iCorrelationID

buildRegisterAppInterface(String appName, String appID)

```
public static RegisterAppInterface buildRegisterAppInterface(String  
appName,String appID)
```

Parameters

appName
appID

buildRegisterAppInterface(String appName, Boolean isMediaApp, String appID)

```
public static RegisterAppInterface buildRegisterAppInterface(String  
appName,Boolean isMediaApp,String appID)
```

Parameters

appName
isMediaApp
appID

```
buildRegisterAppInterface(  
SdlMsgVersion sdlMsgVersion,  
String appName,  
java.util.Vector<TTSChunk>  
ttsName, String  
ngnMediaScreenAppName,  
java.util.Vector<java.lang.String>  
vrSynonyms, Boolean  
isMediaApp, Language  
languageDesired, Language  
hmiDisplayLanguageDesired,  
java.util.Vector<AppHMType>  
appType, String appId, Integer  
correlationID, DeviceInfo  
deviceInfo )
```

```
public static RegisterAppInterface buildRegisterAppInterface(  
SdlMsgVersion sdlMsgVersion,String appName,java.util.Vector<  
TTSChunk> ttsName,String ngnMediaScreenAppName,java.util.Vector<  
java.lang.String> vrSynonyms,Boolean isMediaApp,Language  
languageDesired,Language hmiDisplayLanguageDesired,java.util.Vector  
<AppHMType> appType,String appId,Integer correlationID,DeviceInfo  
deviceInfo)
```

Parameters

sdIMsgVersion
appName
ttsName
ngnMediaScreenAppName
vrSynonyms
isMediaApp
languageDesired
hmiDisplayLanguageDesired

appType
applID
correlationID
deviceInfo

buildSetApplcon(String
sdlFileName, Integer
correlationID)

```
public static SetApplcon buildSetApplcon(String sdlFileName,Integer  
correlationID)
```

Parameters

sdlFileName

correlationID

buildSetGlobalProperties(String helpPrompt, String timeoutPrompt, Integer correlationID)

```
public static SetGlobalProperties buildSetGlobalProperties(String  
helpPrompt,String timeoutPrompt,Integer correlationID)
```

Parameters

helpPrompt

timeoutPrompt

correlationID

buildSetGlobalProperties(
java.util.Vector<TTSCChunk>
helpChunks,
java.util.Vector<TTSCChunk>
timeoutChunks, Integer
correlationID)

```
public static SetGlobalProperties buildSetGlobalProperties(java.util.  
Vector<TTSCChunk> helpChunks,java.util.Vector<TTSCChunk>  
timeoutChunks,Integer correlationID)
```

Parameters

helpChunks
timeoutChunks
correlationID

buildSetGlobalProperties(String
helpPrompt, String
timeoutPrompt, String
vrHelpTitle,
java.util.Vector<VrHelpItem>
vrHelp, Integer correlationID)

```
public static SetGlobalProperties buildSetGlobalProperties(String  
helpPrompt,String timeoutPrompt,String vrHelpTitle,java.util.Vector<  
VrHelpItem> vrHelp,Integer correlationID)
```

Parameters

helpPrompt
timeoutPrompt
vrHelpTitle
vrHelp

correlationID

```
buildSetGlobalProperties(  
java.util.Vector<TTSCChunk>  
helpChunks,  
java.util.Vector<TTSCChunk>  
timeoutChunks, String  
vrHelpTitle,  
java.util.Vector<VrHelpItem>  
vrHelp, Integer correlationID )
```

```
public static SetGlobalProperties buildSetGlobalProperties(java.util.  
Vector<TTSCChunk> helpChunks,java.util.Vector<TTSCChunk>  
timeoutChunks,String vrHelpTitle,java.util.Vector<VrHelpItem> vrHelp,  
Integer correlationID)
```

Parameters

helpChunks

timeoutChunks
vrHelpTitle
vrHelp
correlationID

buildSetMediaClockTimer(
Integer hours, Integer minutes,
Integer seconds, UpdateMode
updateMode, Integer
correlationID)

```
public static SetMediaClockTimer buildSetMediaClockTimer(Integer  
hours,Integer minutes,Integer seconds,UpdateMode updateMode,  
Integer correlationID)
```

Parameters

hours
minutes
seconds
updateMode
correlationID

buildSetMediaClockTimer(
UpdateMode updateMode,
Integer correlationID)

Deprecated

```
public static SetMediaClockTimer buildSetMediaClockTimer(  
UpdateMode updateMode,Integer correlationID)
```

Parameters

updateMode
correlationID

buildShow(String mainText1,
String mainText2, String
mainText3, String mainText4,
String statusBar, String
mediaClock, String mediaTrack,
Image graphic,
java.util.Vector<SoftButton>
softButtons,
java.util.Vector<java.lang.String>
customPresets, TextAlignment
alignment, Integer correlationID)

```
public static Show buildShow(String mainText1,String mainText2,String  
mainText3,String mainText4,String statusBar,String mediaClock,String  
mediaTrack,Image graphic,java.util.Vector<SoftButton> softButtons,  
java.util.Vector<java.lang.String> customPresets,TextAlignment  
alignment,Integer correlationID)
```

Parameters

mainText1

mainText2

mainText3

mainText4

statusBar

mediaClock

mediaTrack

graphic

softButtons

customPresets

alignment

correlationID

buildShow(String mainText1,
String mainText2, String
mainText3, String mainText4,
TextAlignment alignment, Integer
correlationID)

```
public static Show buildShow(String mainText1,String mainText2,String  
mainText3,String mainText4,TextAlignment alignment,Integer  
correlationID)
```

Parameters

mainText1
mainText2
mainText3

mainText4
alignment
correlationID

buildShow(String mainText1,
String mainText2, String
statusBar, String mediaClock,
String mediaTrack,
TextAlignment alignment, Integer
correlationID)

```
public static Show buildShow(String mainText1,String mainText2,String  
statusBar,String mediaClock,String mediaTrack,TextAlignment  
alignment,Integer correlationID)
```

Parameters

mainText1

mainText2

statusBar

mediaClock

mediaTrack

alignment

correlationID

buildShow(String mainText1, String mainText2, TextAlignment alignment, Integer correlationID)

```
public static Show buildShow(String mainText1,String mainText2,  
TextAlignment alignment,Integer correlationID)
```

Parameters

mainText1
mainText2
alignment
correlationID

buildSpeak(String ttsText, Integer correlationID)

```
public static Speak buildSpeak(String ttsText,Integer correlationID)
```

Parameters

ttsText
correlationID

buildSpeak(java.util.Vector<TTSCChunk> ttsChunks, Integer correlationID)

```
public static Speak buildSpeak(java.util.Vector<TTSCChunk> ttsChunks,  
Integer correlationID)
```

Parameters

ttsChunks
correlationID

buildSubscribeButton(
ButtonName buttonName,
Integer correlationID)

```
public static SubscribeButton buildSubscribeButton(ButtonName  
buttonName,Integer correlationID)
```

Parameters

buttonName
correlationID

buildUnregisterAppInterface(Integer correlationID)

```
public static UnregisterAppInterface buildUnregisterAppInterface(  
Integer correlationID)
```

Parameters

correlationID

buildUnsubscribeButton(ButtonName buttonName, Integer correlationID)

```
public static UnsubscribeButton buildUnsubscribeButton(ButtonName  
buttonName,Integer correlationID)
```


Parameters

buttonName
correlationID

```
BuildSubscribeVehicleData(  
  boolean gps, boolean speed,  
  boolean rpm, boolean fuelLevel,  
  boolean fuelLevel_State, boolean  
  instantFuelConsumption, boolean  
  externalTemperature, boolean  
  prndl, boolean tirePressure,  
  boolean odometer, boolean  
  beltStatus, boolean  
  bodyInformation, boolean  
  deviceStatus, boolean  
  driverBraking, Integer  
  correlationID )
```

```
public static SubscribeVehicleData BuildSubscribeVehicleData(boolean
gps,boolean speed,boolean rpm,boolean fuelLevel,boolean
fuelLevel_State,boolean instantFuelConsumption,boolean
externalTemperature,boolean prndl,boolean tirePressure,boolean
odometer,boolean beltStatus,boolean bodyInformation,boolean
deviceStatus,boolean driverBraking,Integer correlationID)
```

Parameters

gps
speed
rpm
fuelLevel
fuelLevel_State
instantFuelConsumption

externalTemperature

prndl

tirePressure

odometer

beltStatus

bodyInformation

deviceStatus

driverBraking

correlationID

BuildUnsubscribeVehicleData(
boolean gps, boolean speed,
boolean rpm, boolean fuelLevel,
boolean fuelLevel_State, boolean
instantFuelConsumption, boolean
externalTemperature, boolean
prndl, boolean tirePressure,
boolean odometer, boolean
beltStatus, boolean
bodyInformation, boolean
deviceStatus, boolean
driverBraking, Integer
correlationID)

```
public static UnsubscribeVehicleData BuildUnsubscribeVehicleData(  
boolean gps,boolean speed,boolean rpm,boolean fuelLevel,boolean  
fuelLevel_State,boolean instantFuelConsumption,boolean  
externalTemperature,boolean prndl,boolean tirePressure,boolean  
odometer,boolean beltStatus,boolean bodyInformation,boolean  
deviceStatus,boolean driverBraking,Integer correlationID)
```

Parameters

gps
speed
rpm
fuelLevel
fuelLevel_State
instantFuelConsumption
externalTemperature
prndl

tirePressure
odometer
beltStatus
bodyInformation
deviceStatus
driverBraking
correlationID

BuildGetVehicleData(boolean
gps, boolean speed, boolean
rpm, boolean fuelLevel, boolean
fuelLevel_State, boolean
instantFuelConsumption, boolean
externalTemperature, boolean
vin, boolean prndl, boolean
tirePressure, boolean odometer,
boolean beltStatus, boolean
bodyInformation, boolean
deviceStatus, boolean
driverBraking, Integer
correlationID)

```
public static GetVehicleData BuildGetVehicleData(boolean gps,boolean  
speed,boolean rpm,boolean fuelLevel,boolean fuelLevel_State,boolean  
instantFuelConsumption,boolean externalTemperature,boolean vin,  
boolean prndl,boolean tirePressure,boolean odometer,boolean  
beltStatus,boolean bodyInformation,boolean deviceStatus,boolean  
driverBraking,Integer correlationID)
```

Parameters

gps
speed
rpm
fuelLevel
fuelLevel_State
instantFuelConsumption
externalTemperature
vin

prndl

tirePressure

odometer

beltStatus

bodyInformation

deviceStatus

driverBraking

correlationID

BuildScrollableMessage(String
scrollableMessageBody, Integer
timeout,
java.util.Vector<SoftButton>
softButtons, Integer correlationID
)

```
public static ScrollableMessage BuildScrollableMessage(String  
scrollableMessageBody,Integer timeout,java.util.Vector<SoftButton>  
softButtons,Integer correlationID)
```

Parameters

scrollableMessageBody
timeout
softButtons
correlationID

```
BuildSlider( Integer numTicks,  
Integer position, String  
sliderHeader,  
java.util.Vector<java.lang.String>  
sliderFooter, Integer timeout,  
Integer correlationID )
```

```
public static Slider BuildSlider(Integer numTicks,Integer position,String  
sliderHeader,java.util.Vector<java.lang.String> sliderFooter,Integer  
timeout,Integer correlationID)
```

Parameters

numTicks
position
sliderHeader

sliderFooter
timeout
correlationID

BuildChangeRegistration(
Language language, Language
hmiDisplayLanguage, Integer
correlationID)

```
public static ChangeRegistration BuildChangeRegistration(Language  
language, Language hmiDisplayLanguage, Integer correlationID)
```

Parameters

language
hmiDisplayLanguage

correlationID

BuildSetDisplayLayout(String displayLayout, Integer correlationID)

```
public static SetDisplayLayout BuildSetDisplayLayout(String  
displayLayout,Integer correlationID)
```

Parameters

displayLayout
correlationID

```
BuildPerformAudioPassThru(  
String ttsText, String  
audioPassThruDisplayText1,  
String  
audioPassThruDisplayText2,  
SamplingRate samplingRate,  
Integer maxDuration,  
BitsPerSample bitsPerSample,  
AudioType audioType, Boolean  
muteAudio, Integer correlationID  
)
```

```
public static PerformAudioPassThru BuildPerformAudioPassThru(String  
ttsText,String audioPassThruDisplayText1,String  
audioPassThruDisplayText2,SamplingRate samplingRate,Integer  
maxDuration,BitsPerSample bitsPerSample,AudioType audioType,  
Boolean muteAudio,Integer correlationID)
```

Parameters

ttsText

audioPassThruDisplayText1

audioPassThruDisplayText2

samplingRate

maxDuration

bitsPerSample

audioType

muteAudio

correlationID

```
BuildPerformAudioPassThru(  
java.util.Vector<TTSCChunk>  
initialPrompt, String  
audioPassThruDisplayText1,  
String  
audioPassThruDisplayText2,  
SamplingRate samplingRate,  
Integer maxDuration,  
BitsPerSample bitsPerSample,  
AudioType audioType, Boolean  
muteAudio, Integer correlationID  
)
```

```
public static PerformAudioPassThru BuildPerformAudioPassThru(java.util.  
.Vector<TTSCChunk> initialPrompt,String audioPassThruDisplayText1,  
String audioPassThruDisplayText2,SamplingRate samplingRate,Integer  
maxDuration,BitsPerSample bitsPerSample,AudioType audioType,  
Boolean muteAudio,Integer correlationID)
```

Parameters

initialPrompt

audioPassThruDisplayText1
audioPassThruDisplayText2
samplingRate
maxDuration
bitsPerSample
audioType
muteAudio
correlationID

BuildEndAudioPassThru(Integer
correlationID)

```
public static EndAudioPassThru BuildEndAudioPassThru(Integer correlationID)
```

Parameters

correlationID

BuildDeviceInfo(String carrierName)

```
public static DeviceInfo BuildDeviceInfo(String carrierName)
```

Parameters

carrierName

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class RPCStreamController

Hierarchy

java.lang.Object

Package

com.smartdevicelink.proxy

Quick Jump

Constructors

- [RPCStreamController\(StreamRPCPacketizer rpcPacketizer, Integer iCorrelationID \)](#)

Methods

- `getCorrelationID()`
- `pause()`
- `resume()`
- `stop()`

Constructors

`RPCStreamController(
StreamRPCPacketizer
rpcPacketizer, Integer
iCorrelationID)`

```
public RPCStreamController(StreamRPCPacketizer rpcPacketizer,Integer  
iCorrelationID)
```

Parameters

`rpcPacketizer`

`iCorrelationID`

Methods

getCorrelationID()

```
public java.lang.Integer getCorrelationID()
```

pause()

```
public void pause()
```

resume()

```
public void resume()
```

stop()

```
public void stop()
```

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class RouterServiceValidator

Hierarchy

java.lang.Object

Package

com.smartdevicelink.transport

Overview

This class will tell us if the currently running router service is valid or not. To use this class simply create a new instance of RouterServiceValidator with a

supplied context. After that, you have the option to set if you want to test in a production setting. If not, it will default to a debug setting. Once you are ready to check if the router service is trusted you simply call `routerServiceValidator.validate()`;

This validator should be passed into the multiplexing transport construction as well.

Quick Jump

Fields

- `ROUTER_SERVICE_PACKAGE`
- `FLAG_DEBUG_NONE`
- `FLAG_DEBUG_PACKAGE_CHECK`
- `FLAG_DEBUG_VERSION_CHECK`
- `FLAG_DEBUG_INSTALLED_FROM_CHECK`
- `FLAG_DEBUG_USE_TIMESTAMP_CHECK`
- `FLAG_DEBUG_PERFORM_ALL_CHECKS`

Constructors

- `RouterServiceValidator( Context context )`
- `RouterServiceValidator( Context context, ComponentName service )`
- `RouterServiceValidator( MultiplexTransportConfig config )`

Methods

- `validate()`
- `getService()`
- `setFlags(int flags)`
- `setSecurityLevel(int securityLevel)`
- `getRefreshRate()`
- `componentNameForServiceRunning(PackageManager pm)`
- `wasInstalledByAppStore(String packageName)`
- `inDebugMode()`
- `verifyVersion(int version,JSONArray versions)`
- `createTrustedListRequest(Context context,boolean forceRefresh)`

- `createTrustedListRequest(Context context,boolean forceRefresh,RouterServiceValidator.TrustedListCallback listCallback)`
- `createTrustedListRequest(Context context,boolean forceRefresh,HttpRequestTask.HttpRequestTaskCallback cb)`
- **Deprecated**
- `createTrustedListRequest(Context context,boolean forceRefresh,HttpRequestTask.HttpRequestTaskCallback cb,RouterServiceValidator.TrustedListCallback listCallback)`
- `isServiceRunning(Context context,ComponentName service)`
- `stringToJson(String json)`
- `invalidateList(Context context)`
- `setTrustedList(Context context,String jsonString)`
- `getTrustedList(Context context)`
- `getTrustedAppListTimeStamp(Context context)`
- `setLastRequest(Context context,String request)`
- `getLastRequest(Context context)`
- `cacheSecurityLevel(Context context,int securityLevel)`
- `getSecurityLevel(Context context)`

Fields

ROUTER_SERVICE_PACKAGE

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

Constant Value

"com.sdl.router"

FLAG_DEBUG_NONE

```
public static final int FLAG_DEBUG_NONE
```

FLAG_DEBUG_PACKAGE_CHECK

```
public static final int FLAG_DEBUG_PACKAGE_CHECK
```

Constant Value

1

FLAG_DEBUG_VERSION_CHECK

```
public static final int FLAG_DEBUG_VERSION_CHECK
```

Constant Value

3

FLAG_DEBUG_INSTALLED_FROM_CHECK

```
public static final int FLAG_DEBUG_INSTALLED_FROM_CHECK
```

Constant Value

4

FLAG_DEBUG_USE_TIMESTAMP_CHECK

```
public static final int FLAG_DEBUG_USE_TIMESTAMP_CHECK
```

Constant Value

5

FLAG_DEBUG_PERFORM_ALL_CHECKS

```
public static final int FLAG_DEBUG_PERFORM_ALL_CHECKS
```

Constant Value

255

Constructors

RouterServiceValidator(Context context)

```
public RouterServiceValidator(Context context)
```

Parameters

context

RouterServiceValidator(Context context, ComponentName service)

```
public RouterServiceValidator(Context context,ComponentName service )
```

Parameters

context
service

RouterServiceValidator(MultiplexTransportConfig config)

```
public RouterServiceValidator(MultiplexTransportConfig config)
```

Parameters

```
config
```

Methods

validate()

Main function to call to ensure we are connecting to a validated router service

```
public boolean validate()
```

getService()

```
public android.content.ComponentName getService()
```

setFlags(int flags)

Use this method if you would like to test your app in a production setting rather than defaulting to a debug mode where you connect to whatever router service is running.

These flags are only used in debugging mode. During production they will be ignored.

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

Parameters

flags

setSecurityLevel(int securityLevel)

```
public void setSecurityLevel(int securityLevel)
```

Parameters

```
securityLevel
```

getRefreshRate()

```
protected static long getRefreshRate()
```

componentNameForServiceRunning (PackageManager pm)

This method will find which router service is running. Use that info to find out more about that app and service. It will store the found service for later use and return the package name if found.

```
public android.content.ComponentName  
componentNameForServiceRunning(PackageManager pm)
```

Parameters

pm

wasInstalledByAppStore(String packageName)

Check to see if the app was installed from a trusted app store.

```
public boolean wasInstalledByAppStore(String packageName)
```

Parameters

packageName

the package name of the app to be tested

inDebugMode()

This method will check to see if this app is a debug build. If it is, we will attempt to connect to any router service. If false, it will only connect to approved apps with router services.

```
public boolean inDebugMode()
```

verifyVersion(int version, JSONArray versions)

```
protected boolean verifyVersion(int version,JSONArray versions)
```

Parameters

version
versions

createTrustedListRequest(Context context, boolean forceRefresh)

Performs a look up against installed SDL apps that support the router service. When it receives a list back from the server it will store it for later use.

```
public static boolean createTrustedListRequest(Context context,boolean  
forceRefresh)
```

Parameters

context
forceRefresh

createTrustedListRequest(
Context context, boolean
forceRefresh,
RouterServiceValidator.TrustedListCallback
listCallback)

```
public static boolean createTrustedListRequest(Context context, boolean  
forceRefresh, RouterServiceValidator.TrustedListCallback listCallback)
```

Parameters

context
forceRefresh
listCallback

```
createTrustedListRequest(  
Context context, boolean  
forceRefresh,  
HttpRequestTask.HttpRequestTaskCallback  
cb )
```

Deprecated

```
protected static boolean createTrustedListRequest(Context context,  
boolean forceRefresh,HttpRequestTask.HttpRequestTaskCallback cb)
```

Parameters

context
forceRefresh
cb

```
createTrustedListRequest(  
Context context, boolean  
forceRefresh,  
HttpRequestTask.HttpRequestTaskCallback  
cb,  
RouterServiceValidator.TrustedListCallback  
listCallback )
```

```
protected static boolean createTrustedListRequest(Context context,  
boolean forceRefresh,HttpRequestTask.HttpRequestTaskCallback cb,  
RouterServiceValidator.TrustedListCallback listCallback)
```

Parameters

context
forceRefresh
cb

listCallback

isServiceRunning(Context context, ComponentName service)

This method will determine if our supplied component name is really running.

```
protected boolean isServiceRunning(Context context,ComponentName service)
```

Parameters

context

service

stringToJson(String json)

Parses a string into a JSON array

```
protected org.json.JSONObject stringToJson(String json)
```

Parameters

json

invalidateList(Context context)

```
public static boolean invalidateList(Context context)
```

Parameters

context

setTrustedList(Context context, String jsonString)

Saves the list of available applications into user's shared prefs.

```
protected static boolean setTrustedList(Context context,String  
jsonString)
```

Parameters

context

The application's environment

jsonString

The JSON string to save.

getTrustedList(Context context)

Retrieves the list of available applications from user's shared prefs.


```
protected static java.lang.String getTrustedList(Context context)
```

Parameters

context

The application's environment.

getTrustedAppListTimeStamp(Context context)

Retrieves the time stamp from the user's shared prefs.

```
protected static java.lang.Long getTrustedAppListTimeStamp(Context  
context)
```

Parameters

context

The application's environment.

setLastRequest(Context context, String request)

```
protected static boolean setLastRequest(Context context,String request
)
```

Parameters

context
request

getLastRequest(Context context)

Gets the last request JSON object we sent to the RSVP server. It basically contains a list of sdl enabled apps

```
protected static java.lang.String getLastRequest(Context context)
```

Parameters

context

cacheSecurityLevel(Context context, int securityLevel)

```
protected static boolean cacheSecurityLevel(Context context,int securityLevel)
```

Parameters

context

securityLevel

getSecurityLevel(Context context)

```
protected static int getSecurityLevel(Context context)
```

Parameters

context

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class

RouterServiceValidator.SdlApp

Hierarchy

java.lang.Object

Package

com.smartdevicelink.transport

Overview

Class that holds all the info we want to send/receive from the validation server

Quick Jump

Methods

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class

RouterServiceValidator.TrustedAppS

Hierarchy

java.lang.Object

java.lang.Enum

Package

com.smartdevicelink.transport

Quick Jump

Methods

- `values()`
- `valueOf(String name)`
- `isTrustedStore(String packageString)`

Methods

values()

```
public static com.smartdevicelink.transport.RouterServiceValidator.  
TrustedAppStore values()
```

valueOf(String name)

```
public static com.smartdevicelink.transport.RouterServiceValidator.  
TrustedAppStore valueOf(String name)
```

Parameters

name

isTrustedStore(String packageString)

Test if the supplied store package is one of the trusted app stores

```
public static boolean isTrustedStore(String packageString)
```

Parameters

packageString

Inherited Methods

From Class	Methods
java.lang.Enum	name, ordinal, toString, equals, hashCode, clone, compareTo, getDeclaringClass, valueOf, finalize
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Interface RouterServiceValidator.TrustedListC

Package

com.smartdevicelink.transport

Overview

This interface is used as a callback to know when we have either obtained a list or at least returned from our attempt.

Quick Jump

Methods

- `onListObtained(boolean successful)`

Methods

`onListObtained(boolean
successful)`

```
public void onListObtained(boolean successful)
```

Parameters

`successful`

Class SDLLockScreenActivity

Hierarchy

java.lang.Object

android.content.Context

android.content.ContextWrapper

android.view.ContextThemeWrapper

android.app.Activity

Package

com.smartdevicelink.managers.lockscreen

Quick Jump

Fields

- LOCKSCREEN_COLOR_EXTRA
- LOCKSCREEN_ICON_EXTRA
- LOCKSCREEN_DEVICE_LOGO_EXTRA
- LOCKSCREEN_DEVICE_LOGO_BITMAP
- LOCKSCREEN_CUSTOM_VIEW_EXTRA
- LOCKSCREEN_DEVICE_LOGO_DOWNLOADED
- CLOSE_LOCK_SCREEN_ACTION

Constructors

- [SDLLockScreenActivity\(\)](#)

Methods

- `onCreate(Bundle savedInstanceState)`
- `onDestroy()`
- `onBackPressed()`
- `onNewIntent(Intent intent)`
- `initializeActivity(Intent intent)`

Fields

LOCKSCREEN_COLOR_EXTRA

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

Constant Value

```
"LOCKSCREEN_COLOR_EXTRA"
```

LOCKSCREEN_ICON_EXTRA

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

Constant Value

"LOCKSCREEN_ICON_EXTRA"

LOCKSCREEN_DEVICE_LOGO_EXTRA

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

Constant Value

"LOCKSCREEN_DEVICE_LOGO_EXTRA"

LOCKSCREEN_DEVICE_LOGO_BITMAP

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

Constant Value

"LOCKSCREEN_DEVICE_LOGO_BITMAP"

LOCKSCREEN_CUSTOM_VIEW_EXTRA

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

Constant Value

"LOCKSCREEN_CUSTOM_VIEW_EXTRA"

LOCKSCREEN_DEVICE_LOGO_DOWNLOADED

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

Constant Value

"LOCKSCREEN_DEVICE_LOGO_DOWNLOADED"

CLOSE_LOCK_SCREEN_ACTION

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

Constant Value

"CLOSE_LOCK_SCREEN"

Inherited Fields

From Class	Fields
android.app.Activity	DEFAULT_KEYS_DIALER, DEFAULT_KEYS_DISABLE, DEFAULT_KEYS_SEARCH_GLOBAL, DEFAULT_KEYS_SEARCH_LOCAL, DEFAULT_KEYS_SHORTCUT, FOCUSED_STATE_SET, RESULT_CANCELED, RESULT_FIRST_USER, RESULT_OK

android.content.Context

ACCESSIBILITY_SERVICE,
ACCOUNT_SERVICE, ACTIVITY_SERVICE,
ALARM_SERVICE, APPWIDGET_SERVICE,
APP_OPS_SERVICE, AUDIO_SERVICE,
BATTERY_SERVICE, BIND_ABOVE_CLIENT,
BIND_ADJUST_WITH_ACTIVITY,
BIND_ALLOW_OOM_MANAGEMENT,
BIND_AUTO_CREATE,
BIND_DEBUG_UNBIND, BIND_IMPORTANT,
BIND_NOT_FOREGROUND,
BIND_WAIVE_PRIORITY,
BLUETOOTH_SERVICE, CAMERA_SERVICE,
CAPTIONING_SERVICE,
CARRIER_CONFIG_SERVICE,
CLIPBOARD_SERVICE,
CONNECTIVITY_SERVICE,
CONSUMER_IR_SERVICE,
CONTEXT_IGNORE_SECURITY,
CONTEXT_INCLUDE_CODE,
CONTEXT_RESTRICTED,
DEVICE_POLICY_SERVICE,
DISPLAY_SERVICE, DOWNLOAD_SERVICE,
DROPBOX_SERVICE,
FINGERPRINT_SERVICE,
INPUT_METHOD_SERVICE,
INPUT_SERVICE,
JOB_SCHEDULER_SERVICE,
KEYGUARD_SERVICE,
LAUNCHER_APPS_SERVICE,
LAYOUT_INFLATER_SERVICE,
LOCATION_SERVICE,
MEDIA_PROJECTION_SERVICE,
MEDIA_ROUTER_SERVICE,
MEDIA_SESSION_SERVICE, MIDI_SERVICE,
MODE_APPEND,
MODE_ENABLE_WRITE_AHEAD_LOGGING,
MODE_MULTI_PROCESS, MODE_PRIVATE,
MODE_WORLD_READABLE,
MODE_WORLD_WRITEABLE,
NETWORK_STATS_SERVICE,
NFC_SERVICE, NOTIFICATION_SERVICE,
NSD_SERVICE, POWER_SERVICE,
PRINT_SERVICE, RESTRICTIONS_SERVICE,
SEARCH_SERVICE, SENSOR_SERVICE,
STORAGE_SERVICE, TELECOM_SERVICE,
TELEPHONY_SERVICE,
TELEPHONY_SUBSCRIPTION_SERVICE

Constructors

SDLLockScreenActivity()

```
public SDLLockScreenActivity()
```

Methods

onCreate(Bundle savedInstanceState)

```
protected void onCreate(Bundle savedInstanceState)
```

Parameters

savedInstanceState

onDestroy()

```
protected void onDestroy()
```

onBackPressed()

```
public void onBackPressed()
```

onNewIntent(Intent intent)

```
protected void onNewIntent(Intent intent)
```

Parameters

intent

initializeActivity(Intent intent)

```
public void initializeActivity(Intent intent)
```

Parameters

intent

Inherited Methods

From Class

Methods

getIntent, setIntent, getApplication,
isChild, getParent, getWindowManager,
getWindow, getLoaderManager,
getCurrentFocus, onCreate, onCreate,
onRestoreInstanceState,
onRestoreInstanceState, onPostCreate,
onPostCreate, onStart, onRestart,
onStateNotSaved, onResume,
onPostResume, isVoiceInteraction,
isVoiceInteractionRoot,
getVoiceInteractor, onNewIntent,
onSaveInstanceState,
onSaveInstanceState, onPause,
onUserLeaveHint, onCreateThumbnail,
onCreateDescription,
onProvideAssistData,
onProvideAssistContent, showAssist,
onStop, onDestroy, reportFullyDrawn,
onConfigurationChanged,
getChangingConfigurations,
getLastNonConfigurationInstance,
onRetainNonConfigurationInstance,
onLowMemory, onTrimMemory,
getFragmentManager,
onAttachFragment, managedQuery,
startManagingCursor,
stopManagingCursor, findViewById,
getActionBar, setActionBar,
setContentView, setContentView,
setContentView, addContentView,
getContentTransitionManager,
setContentTransitionManager,
getContentScene,
setFinishOnTouchOutside,
setDefaultKeyMode, onKeyDown,
onKeyLongPress, onKeyUp,
onKeyMultiple, onBackPressed,
onKeyShortcut, onTouchEvent,
onTrackballEvent, onGenericMotionEvent,
onUserInteraction,
onWindowAttributesChanged,
onContentChanged,
onWindowFocusChanged,
onAttachedToWindow,
onDetachedFromWindow,
hasWindowFocus, dispatchKeyEvent,
dispatchKeyShortcutEvent

android.view.ContextThemeWrapper

attachBaseContext,
applyOverrideConfiguration,
getResources, setTheme, getTheme,
getSystemService,
onApplyThemeResource

android.content.ContextWrapper

attachBaseContext, getBaseContext,
getAssets, getResources,
getPackageManager,
getContentResolver, getMainLooper,
getApplicationContext, setTheme,
getTheme, getClassLoader,
getPackageName, getApplicationInfo,
getPackageResourcePath,
getPackageCodePath,
getSharedPreferences, openFileInput,
openFileOutput, deleteFile,
getFileStreamPath, fileList, getFilesDir,
getNoBackupFilesDir, getExternalFilesDir,
getExternalFilesDirs, getObbDir,
getObbDirs, getCacheDir,
getCodeCacheDir, getExternalCacheDir,
getExternalCacheDirs,
getExternalMediaDirs, getDir,
openOrCreateDatabase,
openOrCreateDatabase, deleteDatabase,
getDatabasePath, databaseList,
getWallpaper, peekWallpaper,
getWallpaperDesiredMinimumWidth,
getWallpaperDesiredMinimumHeight,
setWallpaper, setWallpaper,
clearWallpaper, startActivity,
startActivity, startActivities,
startActivities, startIntentSender,
startIntentSender, sendBroadcast,
sendBroadcast, sendOrderedBroadcast,
sendOrderedBroadcast,
sendBroadcastAsUser,
sendBroadcastAsUser,
sendOrderedBroadcastAsUser,
sendStickyBroadcast,
sendStickyOrderedBroadcast,
removeStickyBroadcast,
sendStickyBroadcastAsUser,
sendStickyOrderedBroadcastAsUser,
removeStickyBroadcastAsUser,
registerReceiver, registerReceiver,
unregisterReceiver, startService,
stopService, bindService, unbindService,
startInstrumentation, getSystemService,
getSystemServiceName,
checkPermission,
checkCallingPermission

android.content.Context

getAssets, getResources,
getPackageManager,
getContentResolver, getMainLooper,
getApplicationContext,
registerComponentCallbacks,
unregisterComponentCallbacks, getText,
getString, getString, getColor,
getDrawable, getColorStateList,
setTheme, getTheme,
obtainStyledAttributes,
obtainStyledAttributes,
obtainStyledAttributes,
obtainStyledAttributes, getClassLoader,
getPackageName, getApplicationInfo,
getPackageResourcePath,
getPackageCodePath,
getSharedPreferences, openFileInput,
openFileOutput, deleteFile,
getFileStreamPath, getFilesDir,
getNoBackupFilesDir, getExternalFilesDir,
getExternalFilesDirs, getObbDir,
getObbDirs, getCacheDir,
getCodeCacheDir, getExternalCacheDir,
getExternalCacheDirs,
getExternalMediaDirs, fileList, getDir,
openOrCreateDatabase,
openOrCreateDatabase, deleteDatabase,
getDatabasePath, databaseList,
getWallpaper, peekWallpaper,
getWallpaperDesiredMinimumWidth,
getWallpaperDesiredMinimumHeight,
setWallpaper, setWallpaper,
clearWallpaper, startActivity,
startActivity, startActivities,
startActivities, startIntentSender,
startIntentSender, sendBroadcast,
sendBroadcast, sendOrderedBroadcast,
sendOrderedBroadcast,
sendBroadcastAsUser,
sendBroadcastAsUser,
sendOrderedBroadcastAsUser,
sendStickyBroadcast,
sendStickyOrderedBroadcast,
removeStickyBroadcast,
sendStickyBroadcastAsUser,
sendStickyOrderedBroadcastAsUser,
removeStickyBroadcastAsUser

java.lang.Object

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

Class SampleBuffer

Hierarchy

java.lang.Object

Package

com.smartdevicelink.managers.audio

Overview

Wraps a buffer of raw audio samples depending on the sample type (8 bit, 16 bit) Unifies samples into double.

Quick Jump

Methods

- [wrap\(ByteBuffer buffer,int sampleType,long presentationTimeUs\)](#)
- [wrap\(ByteBuffer buffer,int sampleType,int channelCount,long presentationTimeUs\)](#)
- [allocate\(int capacity,int sampleType,ByteOrder byteOrder,long presentationTimeUs\)](#)
- [allocate\(int capacity,int sampleType,int channelCount,ByteOrder byteOrder,long presentationTimeUs\)](#)

- `capacity()`
- `limit()`
- `limit(int newLimit)`
- `position()`
- `position(int newPosition)`
- `get()`
- `get(int index)`
- `put(double sample)`
- `put(int index,double sample)`
- `getByteBuffer()`
- `getBytes()`
- `getPresentationTimeUs()`

Methods

`wrap( ByteBuffer buffer, int sampleType, long presentationTimeUs )`

Wraps a raw (mono) byte buffer to a new sample buffer.

```
public static com.smartdevicelink.managers.audio.SampleBuffer wrap(  
    ByteBuffer buffer,int sampleType,long presentationTimeUs)
```

Parameters

`buffer`

The raw buffer to be wrapped.

sampleType

The sample type of the samples in the raw buffer.

presentationTimeUs

The presentation time of the buffer.

wrap(ByteBuffer buffer, int sampleType, int channelCount, long presentationTimeUs)

Wraps a raw byte buffer to a new sample buffer.

```
public static com.smartdevicelink.managers.audio.SampleBuffer wrap(  
ByteBuffer buffer,int sampleType,int channelCount,long  
presentationTimeUs)
```

Parameters

buffer

The raw buffer to be wrapped.

sampleType

The sample type of the samples in the raw buffer.

channelCount

The number of channels (1 = mono, 2 = stereo).

presentationTimeUs

The presentation time of the buffer.

allocate(int capacity, int sampleType, ByteOrder byteOrder, long presentationTimeUs)

Allocates a new sample buffer.

```
public static com.smartdevicelink.managers.audio.SampleBuffer  
allocate(int capacity,int sampleType,ByteOrder byteOrder,long  
presentationTimeUs)
```

Parameters

capacity

The specified sample capacity of the sample buffer.

sampleType

The sample type of the samples the buffer should store.

byteOrder

The byte order for the samples (little or big endian).

presentationTimeUs

The presentation time for the buffer.

allocate(int capacity, int sampleType, int channelCount, ByteOrder byteOrder, long presentationTimeUs)

Allocates a new sample buffer.

```
public static com.smartdevicelink.managers.audio.SampleBuffer  
allocate(int capacity,int sampleType,int channelCount,ByteOrder  
byteOrder,long presentationTimeUs)
```

Parameters

capacity

The specified sample capacity of the sample buffer.

sampleType

The sample type of the samples the buffer should store.

channelCount

The number of channels (1 = mono, 2 = stereo).

byteOrder

The byte order for the samples (little or big endian).

presentationTimeUs

The presentation time for the buffer.

capacity()

Returns the capacity of the buffer per channel.

```
public int capacity()
```

limit()

Returns the number of samples in the buffer per channel.

```
public int limit()
```

limit(int newLimit)

Sets the number of samples in the buffer to the new limit.

```
public void limit(int newLimit)
```

Parameters

newLimit

The new limit of the sample buffer.

position()

Returns the current position in the buffer per channel.

```
public int position()
```

position(int newPosition)

Sets the position of the sample buffer to the new index.

```
public void position(int newPosition)
```

Parameters

newPosition

The new position of the sample buffer.

get()

Returns the sample of the current position and then increments the position.
The sample returned is a mixed sample getting all samples from each channel.

```
public double get()
```

get(int index)

Returns the sample from the given index in the buffer. If the buffer's channel count is > 1 the sample returned is a mixed sample getting all samples from each channel.

```
public double get(int index)
```

Parameters

index

The index of the sample requested.

put(double sample)

Puts a sample to the current position and increments the position.

```
public void put(double sample)
```

Parameters

sample

The sample to put into the buffer.

put(int index, double sample)

Puts a sample to the given index in the buffer. If the buffer's channel count is > 1 the sample will be stored in each channel at the given index.

```
public void put(int index,double sample)
```

Parameters

index

The index to put the sample.

sample

The sample to store in the buffer.

getByteBuffer()

Returns the raw byte buffer managed by this sample buffer.

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

getBytes()

Returns a copy of the bytes from position 0 to the current limit.

```
public byte getBytes()
```


getPresentationTimeUs()

The presentation time of this sample buffer.

```
public long getPresentationTimeUs()
```

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class ScreenManager

Hierarchy

java.lang.Object

BaseScreenManager

Package

com.smartdevicelink.managers.screen

Overview

ScreenManager

Note: This class must be accessed through the SdlManager. Do not instantiate it by itself.

Quick Jump

Constructors

- `ScreenManager( ISdl internalInterface, FileManager fileManager )`

Constructors

`ScreenManager(ISdl
internalInterface, FileManager
fileManager)`

```
public ScreenManager(ISdl internalInterface,FileManager fileManager)
```

Parameters

internalInterface
fileManager

Methods

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class SdlAppInfo

Hierarchy

java.lang.Object

Package

com.smartdevicelink.util

Overview

Created by Joey Grover on 2/2/18.

Quick Jump

Constructors

- [SdlAppInfo\(ResolveInfo resolveInfo, PackageInfo packageInfo \)](#)

Methods

- [getRouterServiceVersion\(\)](#)
- [isCustomRouterService\(\)](#)
- [getRouterServiceComponentName\(\)](#)
- [toString\(\)](#)

Constructors

SdlAppInfo(ResolveInfo resolveInfo, PackageInfo packageInfo)

```
public SdlAppInfo(ResolveInfo resolveInfo,PackageInfo packageInfo)
```

Parameters

resolveInfo
packageInfo

Methods

getRouterServiceVersion()

```
public int getRouterServiceVersion()
```

isCustomRouterService()

```
public boolean isCustomRouterService()
```

getRouterServiceComponentName()

```
public android.content.ComponentName  
getRouterServiceComponentName()
```

toString()

```
public java.lang.String toString()
```

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class

SdlAppInfo.BestRouterComparator

Hierarchy

java.lang.Object

Package

com.smartdevicelink.util

All Implemented Interfaces

java.util.Comparator

Overview

This comparator will sort a list to find the best router service to start out of the known SDL enabled apps

Quick Jump

Constructors

- [BestRouterComparator\(\)](#)

Methods

- `compare(SdlAppInfo one,SdlAppInfo two)`

Constructors

BestRouterComparator()

```
public BestRouterComparator()
```

Methods

`compare(SdlAppInfo one,
SdlAppInfo two)`

```
public int compare(SdlAppInfo one,SdlAppInfo two)
```


Parameters

one

two

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class SdlArtwork

Hierarchy

java.lang.Object

[com.smartdevicelink.managers.file.filetypes.SdlFile](#)

Package

Overview

A class that extends `SdlFile`, representing artwork (JPEG, PNG, or BMP) to be uploaded to core

Quick Jump

Constructors

- `SdlArtwork()`
- `SdlArtwork( String fileName, FileType fileType, int id, boolean persistentFile )`
- `SdlArtwork( String fileName, FileType fileType, Uri uri, boolean persistentFile )`
- `SdlArtwork( String fileName, FileType fileType, byte[] data, boolean persistentFile )`
- `SdlArtwork( StaticIconName staticIconName )`

Methods

- `setTemplatelImage(boolean isTemplate)`
- `isTemplatelImage()`
- `setType(FileType fileType)`
- `getImageRPC()`

Constructors

SdlArtwork()

Creates a new instance of SdlArtwork

```
public SdlArtwork()
```

SdlArtwork(String fileName, FileType fileType, int id, boolean persistentFile)

Creates a new instance of SdlArtwork

```
public SdlArtwork(String fileName,FileType fileType,int id,boolean persistentFile)
```

Parameters

fileName

a String value representing the name that will be used to store the file in the head unit

fileType

a FileType enum value representing the type of the file

id

an int value representing the android resource id of the file

persistentFile

a boolean value that indicates if the file is meant to persist between sessions / ignition cycles

SdlArtwork(String fileName, FileType fileType, Uri uri, boolean persistentFile)

Creates a new instance of SdlArtwork

```
public SdlArtwork(String fileName,FileType fileType,Uri uri,boolean persistentFile)
```

Parameters

fileName

	a String value representing the name that will be used to store the file in the head unit
fileType	a FileType enum value representing the type of the file
uri	a URI value representing a file's location. Currently, it only supports local files
persistentFile	a boolean value that indicates if the file is meant to persist between sessions / ignition cycles

SdlArtwork(String fileName, FileType fileType, byte[] data, boolean persistentFile)

Creates a new instance of SdlArtwork

```
public SdlArtwork(String fileName,FileType fileType,byte[] data,boolean persistentFile)
```

Parameters

fileName

a String value representing the name that will be used to store the file in the head unit

fileType

a FileType enum value representing the type of the file

data

a byte array representing the data of the file

persistentFile

a boolean value that indicates if the file is meant to persist between sessions / ignition cycles

SdlArtwork(StaticIconName staticIconName)

Creates a new instance of SdlArtwork

```
public SdlArtwork(StaticIconName staticIconName)
```

Parameters

staticIconName

a StaticIconName enum value representing the name of a static file that comes pre-shipped with the head unit

Methods

setTemplateImage(boolean isTemplate)

Sets whether this SdlArtwork is a template image whose coloring should be decided by the HMI

```
public void setTemplateImage(boolean isTemplate)
```

Parameters

isTemplate

boolean that tells whether this SdlArtwork is a template image

isTemplateImage()

Gets whether this SdlArtwork is a template image whose coloring should be decided by the HMI

```
public boolean isTemplateImage()
```

setType(FileType fileType)

```
public void setType(FileType fileType)
```

Parameters

fileType

getImageRPC()

Gets the Image RPC representing this artwork. Generally for use internally, you should instead pass an artwork to a Screen Manager method

```
public Image getImageRPC()
```

Inherited Methods

From Class	Methods
com.smartdevicelink.managers.file.filetypes.SdlFile	setName, getName, setResourceId, getResourceId, setUri, getUri, setFileData, getFileData, setType, getType, setPersistent, isPersistent, setStaticIcon, isStaticIcon
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class SdlBroadcastReceiver

Hierarchy

java.lang.Object

android.content.BroadcastReceiver

Package

com.smartdevicelink.transport

Quick Jump

Fields

- [SDL_ROUTER_SERVICE_CLASS_NAME](#)
- [LOCAL_ROUTER_SERVICE_EXTRA](#)
- [LOCAL_ROUTER_SERVICE_DID_START_OWN](#)
- [TRANSPORT_GLOBAL_PREFS](#)
- [IS_TRANSPORT_CONNECTED](#)
- [runningBluetoothServicePackage](#)

Constructors

- [SdlBroadcastReceiver\(\)](#)

Methods

- [getRouterServiceVersion\(\)](#)
- [onReceive\(Context context, Intent intent\)](#)
- [pingRouterService\(Context context, String packageName, String className\)](#)
- [queryForConnectedService\(Context context\)](#)
- [requestTransportStatus\(Context context, SdlRouterStatusProvider.ConnectedStatusCallback callback\)](#) **Deprecated**
- [consumeQueuedRouterService\(\)](#)
- [defineLocalSdlRouterClass\(\)](#)

- `onSdlEnabled(Context context, Intent intent)`

Fields

SDL_ROUTER_SERVICE_CLASS_NAME

```
protected static final java.lang.String  
SDL_ROUTER_SERVICE_CLASS_NAME
```

Constant Value

"sdlrouterservice"

LOCAL_ROUTER_SERVICE_EXTRA

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

Constant Value

"router_service"

LOCAL_ROUTER_SERVICE_DID_START_OWN

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

Constant Value

"did_start"

TRANSPORT_GLOBAL_PREFS

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

Constant Value

"SdlTransportPrefs"

IS_TRANSPORT_CONNECTED

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

Constant Value

"isTransportConnected"

runningBluetoothServicePackage

```
public static java.util.Vector runningBluetoothServicePackage
```

Constructors

SdlBroadcastReceiver()

```
public SdlBroadcastReceiver()
```

Methods

getRouterServiceVersion()

```
public int getRouterServiceVersion()
```

onReceive(Context context, Intent intent)

```
public void onReceive(Context context,Intent intent)
```

Parameters

context
intent

pingRouterService(Context context, String packageName, String className)

Attempts to ping a running router service

```
protected static void pingRouterService(Context context,String  
packageName,String className)
```

Parameters

context

A context to access Android system services through.

packageName

Package name for service to ping

className

Class name for service to ping

queryForConnectedService(Context context)

This call will reach out to all SDL related router services to check if they're connected. If a the router service is connected, it will react by pinging all clients. This receiver will then receive that ping and if the router service is trusted, the onSdlEnabled method will be called.

```
public static void queryForConnectedService(Context context)
```

Parameters

context

requestTransportStatus(Context context, SdlRouterServiceProvider.ConnectedSta callback)

Deprecated

If a Router Service is running, this method determines if that service is connected to a device over some form of transport.

```
public static void requestTransportStatus(Context context,
SdlRouterServiceProvider.ConnectedStatusCallback callback)
```

Parameters

context

A context to access Android system services through. If null is passed, this will always return false

callback

Use this callback to find out if the router service is connected or not.

consumeQueuedRouterService()

```
public static android.content.ComponentName  
consumeQueuedRouterService()
```

defineLocalSdlRouterClass()

We need to define this for local copy of the Sdl Router Service class. It will be the main point of connection for Sdl enabled apps

```
public abstract java.lang.Class defineLocalSdlRouterClass()
```

onSdlEnabled(Context context, Intent intent)

The developer will need to define exactly what should happen when Sdl is enabled. This method will only get called when a Sdl session is initiated.

The most useful code here would be to start the activity or service that handles most of the Livio Connect code.

```
public abstract void onSdlEnabled(Context context,Intent intent)
```

Parameters

context

this is the context that was passed to this receiver when it was called.

intent

this is the intent that alerted this broadcast. Make sure to pass all extra it came with to your service/activity {@inheritDoc}

Inherited Methods

From Class	Methods
android.content.BroadcastReceiver	onReceive, goAsync, peekService, setResultCode, getResultCode, setResultData, getResultData, setResultExtras, getResultExtras, setResult, getAbortBroadcast, abortBroadcast, clearAbortBroadcast, isOrderedBroadcast, isInitialStickyBroadcast, setOrderedHint, setDebugUnregister, getDebugUnregister
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class SdlConnection

Hierarchy

java.lang.Object

Package

com.smartdevicelink.SdlConnection

Quick Jump

Fields

- [cachedMultiConfig](#)

Constructors

- [SdlConnection\(BaseTransportConfig transportConfig \)](#)
- [SdlConnection\(BaseTransportConfig transportConfig, RouterServiceValidator rsvp \)](#)

Methods

- [getWiProProtocol\(\)](#)
- [startTransport\(\)](#)
- [getIsConnected\(\)](#)
- [getBroadcastComment\(\)](#)
- [sendMessage\(ProtocolMessage msg\)](#)
- [onTransportPacketReceived\(SdlPacket packet\)](#)
- [onTransportConnected\(\)](#)
- [onTransportDisconnected\(String info\)](#)

- onTransportError(String info,Exception e)
- onProtocolMessageBytesToSend(SdIPacket packet)
- onProtocolMessageReceived(ProtocolMessage msg)
- onProtocolSessionStarted(SessionType sessionType,byte sessionID,byte version,String correlationID,int hashID,boolean isEncrypted)
- onProtocolSessionNACKed(SessionType sessionType,byte sessionID,byte version,String correlationID,java.util.List<java.lang.String> rejectedParams)
- onProtocolSessionEnded(SessionType sessionType,byte sessionID,String correlationID)
- onProtocolError(String info,Exception e)
- getCurrentTransportType()
- startService(SessionType sessionType,byte sessionID,boolean isEncrypted)
- endService(SessionType sessionType,byte sessionID)
- sendHeartbeat(SdlSession mySession)
- unregisterSession(SdlSession registerListener)
- findSessionById(byte id)
- onAuthTokenReceived(String authToken,byte sessionID)
- getRegisterCount()
- onProtocolHeartbeat(SessionType sessionType,byte sessionID)
- onProtocolHeartbeatACK(SessionType sessionType,byte sessionID)
- onResetOutgoingHeartbeat(SessionType sessionType,byte sessionID)
- onResetIncomingHeartbeat(SessionType sessionType,byte sessionID)
- forceHardwareConnectEvent(TransportType type)
- enableLegacyMode(boolean enable,TransportType type)
- isLegacyModeEnabled()
- onProtocolSessionEndedNACKed(SessionType sessionType,byte sessionID,String correlationID)
- onProtocolServiceDataACK(SessionType serviceType,int dataSize,byte sessionID)

Fields

cachedMultiConfig

```
protected static com.smartdevicelink.transport.  
MultiplexTransportConfig cachedMultiConfig
```

Constructors

SdlConnection(BaseTransportConfig transportConfig)

Constructor.

```
public SdlConnection(BaseTransportConfig transportConfig)
```

Parameters

transportConfig

Transport configuration for this connection.

SdlConnection(BaseTransportConfig transportConfig, RouterServiceValidator rsvp)

```
protected SdlConnection(BaseTransportConfig transportConfig,  
RouterServiceValidator rsvp)
```

Parameters

transportConfig
rsvp

Methods

getWiProProtocol()

```
public com.smartdevicelink.protocol.AbstractProtocol getWiProProtocol  
()
```

startTransport()

```
public void startTransport() throws SdlException
```

Throws

- SdlException

getIsConnected()

```
public java.lang.Boolean getIsConnected()
```

getBroadcastComment()

```
public java.lang.String getBroadcastComment()
```

sendMessage(ProtocolMessage msg)


```
public void sendMessage(ProtocolMessage msg)
```

Parameters

msg

onTransportPacketReceived(SdlPacket packet)

```
public void onTransportPacketReceived(SdlPacket packet)
```

Parameters

packet

onTransportConnected()

```
public void onTransportConnected()
```

onTransportDisconnected(String info)

```
public void onTransportDisconnected(String info)
```

Parameters

info

onTransportError(String info, Exception e)

```
public void onTransportError(String info,Exception e)
```

Parameters

info
e

onProtocolMessageBytesToSend(SdIPacket packet)

```
public void onProtocolMessageBytesToSend(SdIPacket packet)
```

Parameters

packet

onProtocolMessageReceived(ProtocolMessage msg)

```
public void onProtocolMessageReceived(ProtocolMessage msg)
```

Parameters

msg

onProtocolSessionStarted(SessionType sessionType, byte sessionID, byte version, String correlationID, int hashID, boolean isEncrypted)

```
public void onProtocolSessionStarted(SessionType sessionType,byte  
sessionID,byte version,String correlationID,int hashID,boolean  
isEncrypted)
```

Parameters

sessionType
sessionID
version
correlationID
hashID
isEncrypted

onProtocolSessionNACKed(
SessionType sessionType, byte
sessionID, byte version, String
correlationID,
java.util.List<java.lang.String>
rejectedParams)

```
public void onProtocolSessionNACKed(SessionType sessionType,byte  
sessionID,byte version,String correlationID,java.util.List<java.lang.  
String> rejectedParams)
```

Parameters

sessionType
sessionID
version
correlationID

rejectedParams

onProtocolSessionEnded(SessionType sessionType, byte sessionID, String correlationID)

```
public void onProtocolSessionEnded(SessionType sessionType,byte  
sessionID,String correlationID)
```

Parameters

sessionType
sessionID
correlationID

onProtocolError(String info, Exception e)

```
public void onProtocolError(String info,Exception e)
```

Parameters

info
e

getCurrentTransportType()

Gets type of transport currently used by this connection.

```
public TransportType getCurrentTransportType()
```


startService(SessionType
sessionType, byte sessionID,
boolean isEncrypted)

```
public void startService(SessionType sessionType,byte sessionID,  
boolean isEncrypted)
```

Parameters

sessionType
sessionID
isEncrypted

endService(SessionType
sessionType, byte sessionID)

```
public void endService(SessionType sessionType,byte sessionID)
```

Parameters

sessionType
sessionID

sendHeartbeat(SdlSession mySession)

```
public void sendHeartbeat(SdlSession mySession)
```

Parameters

mySession

unregisterSession(SdlSession registerListener)

```
public void unregisterSession(SdlSession registerListener)
```

Parameters

registerListener

findSessionById(byte id)

```
public com.smartdevicelink.SdlConnection.SdlSession findSessionById(  
byte id)
```

Parameters

id

onAuthTokenReceived(String
authToken, byte sessionId)

```
public void onAuthTokenReceived(String authToken,byte sessionId)
```

Parameters

authToken

sessionId

getRegisterCount()

```
public int getRegisterCount()
```

onProtocolHeartbeat(SessionType sessionType, byte sessionID)

```
public void onProtocolHeartbeat(SessionType sessionType,byte  
sessionID)
```

Parameters

sessionType
sessionID

onProtocolHeartbeatACK(SessionType sessionType, byte sessionID)

```
public void onProtocolHeartbeatACK(SessionType sessionType,byte  
sessionID)
```

Parameters

sessionType
sessionID

onResetOutgoingHeartbeat(SessionType sessionType, byte sessionID)

```
public void onResetOutgoingHeartbeat(SessionType sessionType,byte  
sessionID)
```

Parameters

sessionType
sessionID

onResetIncomingHeartbeat(
SessionType sessionType, byte
sessionID)

```
public void onResetIncomingHeartbeat(SessionType sessionType,byte  
sessionID)
```

Parameters

sessionType
sessionID

forceHardwareConnectEvent(TransportType type)

```
public void forceHardwareConnectEvent(TransportType type)
```

Parameters

type

enableLegacyMode(boolean enable, TransportType type)

```
public static void enableLegacyMode(boolean enable,TransportType  
type)
```


Parameters

enable
type

isLegacyModeEnabled()

```
public static boolean isLegacyModeEnabled()
```

onProtocolSessionEndedNACKed(SessionType sessionType, byte sessionID, String correlationID)

```
public void onProtocolSessionEndedNACKed(SessionType sessionType,  
byte sessionID,String correlationID)
```

Parameters

sessionType
sessionID
correlationID

onProtocolServiceDataACK(
SessionType serviceType, int
dataSize, byte sessionID)

```
public void onProtocolServiceDataACK(SessionType serviceType,int  
dataSize,byte sessionID)
```

Parameters

serviceType

dataSize
sessionID

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class SdlEncoder

Hierarchy

java.lang.Object

Package
com.smartdevicelink.encoder

Quick Jump

Constructors

- `SdlEncoder()`

Methods

- `setFrameRate(int iVal)`
- `setFrameInterval(int iVal)`
- `setFrameWidth(int iVal)`
- `setFrameHeight(int iVal)`
- `setBitrate(int iVal)`
- `setOutputStream(PipedOutputStream mStream)`
- `setOutputListener(IVideoStreamListener listener)`
- `prepareEncoder()`
- `startEncoder()`
- `releaseEncoder()`
- `drainEncoder(boolean endOfStream)`

Constructors

SdlEncoder()

```
public SdlEncoder()
```

Methods

setFrameRate(int iVal)

```
public void setFrameRate(int iVal)
```

Parameters

iVal

setFrameInterval(int iVal)

```
public void setFrameInterval(int iVal)
```

Parameters

iVal

setWidth(int iVal)

```
public void setWidth(int iVal)
```

Parameters

iVal

setFrameHeight(int iVal)

```
public void setFrameHeight(int iVal)
```

Parameters

iVal

setBitrate(int iVal)

```
public void setBitrate(int iVal)
```

Parameters

iVal

setOutputStream(PipedOutputStream mStream)

```
public void setOutputStream(PipedOutputStream mStream)
```

Parameters

mStream

setOutputListener(IVideoStreamListener listener)

```
public void setOutputListener(IVideoStreamListener listener)
```

Parameters

listener

prepareEncoder()

```
public android.view.Surface prepareEncoder()
```

startEncoder()


```
public void startEncoder()
```

releaseEncoder()

Releases encoder resources.

```
public void releaseEncoder()
```

drainEncoder(boolean endOfStream)

Extracts all pending data from the encoder

If endOfStream is not set, this returns when there is no more data to drain. If it is set, we send EOS to the encoder, and then iterate until we see EOS on the output. Calling this with endOfStream set should be done once, right before stopping the muxer.

```
public void drainEncoder(boolean endOfStream)
```

Parameters

endOfStream

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class SdlFile

Hierarchy

java.lang.Object

Package

com.smartdevicelink.managers.file.filetypes

All Known Subclasses

[com.smartdevicelink.managers.file.filetypes.SdlArtwork](#)

Overview

A class representing data to be uploaded to core

Quick Jump

Constructors

- `SdlFile()`
- `SdlFile( String fileName, FileType fileType, int id, boolean persistentFile )`
- `SdlFile( String fileName, FileType fileType, Uri uri, boolean persistentFile )`
- `SdlFile( String fileName, FileType fileType, byte[] data, boolean persistentFile )`
- `SdlFile( StaticIconName staticIconName )`

Methods

- `setName(String fileName)`
- `getName()`
- `setResourceId(int id)`
- `getResourceId()`
- `setUri(Uri uri)`
- `getUri()`
- `setFileData(byte[] data)`
- `getFileData()`
- `setType(FileType fileType)`
- `getType()`
- `setPersistent(boolean persistentFile)`
- `isPersistent()`
- `setStaticIcon(boolean staticIcon)`
- `isStaticIcon()`

Constructors

SdlFile()

Creates a new instance of SdlFile

```
public SdlFile()
```

SdlFile(String fileName, FileType fileType, int id, boolean persistentFile)

Creates a new instance of SdlFile

```
public SdlFile(String fileName,FileType fileType,int id,boolean  
persistentFile)
```

Parameters

fileName

a String value representing the name that will be used to store the file in the head unit

fileType

a FileType enum value representing the type of the file

id

an int value representing the android resource id of the file

persistentFile

a boolean value that indicates if the file is meant to persist between sessions / ignition cycles

SdlFile(String fileName, FileType fileType, Uri uri, boolean persistentFile)

Creates a new instance of SdlFile

```
public SdlFile(String fileName,FileType fileType,Uri uri,boolean persistentFile)
```

Parameters

fileName

a String value representing the name that will be used to store the file in the head unit

fileType

a FileType enum value representing the type of the file

uri

a URI value representing a file's location. Currently, it only supports local files

persistentFile

a boolean value that indicates if the file is meant to persist between sessions / ignition cycles

SdlFile(String fileName, FileType fileType, byte[] data, boolean persistentFile)

Creates a new instance of SdlFile

```
public SdlFile(String fileName,FileType fileType,byte[] data,boolean persistentFile)
```

Parameters

fileName

a String value representing the name that will be used to store the file in the head unit

fileType

a FileType enum value representing the type of the file

data

a byte array representing the data of the file

persistentFile

a boolean value that indicates if the file is meant to persist between sessions / ignition cycles

SdlFile(StaticIconName staticIconName)

Creates a new instance of SdlFile

```
public SdlFile(StaticIconName staticIconName)
```

Parameters

staticIconName

a StaticIconName enum value representing the name of a static file that comes pre-shipped with the head unit

Methods

setName(String fileName)

Sets the name of the file

```
public void setName(String fileName)
```

Parameters

fileName

a String value representing the name that will be used to store the file in the head unit

getName()

Gets the name of the file

```
public java.lang.String getName()
```

setResourceId(int id)

Sets the resource ID of the file

```
public void setResourceId(int id)
```

Parameters

id

an int value representing the android resource id of the file

getResourceId()

Gets the resource id of the file

```
public int getResourceId()
```

setUri(Uri uri)

Sets the uri of the file

```
public void setUri(Uri uri)
```

Parameters

uri

a URI value representing a file's location. Currently, it only supports local files

getUri()

Gets the uri of the file

```
public android.net.Uri getUri()
```

setFileData(byte[] data)

Sets the byte array that represents the content of the file

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

Parameters

data

a byte array representing the data of the file

getFileData()

Gets the byte array that represents the content of the file

```
public byte getFileData()
```

setType(FileType fileType)

Sets the type of the file

```
public void setType(FileType fileType)
```

Parameters

fileType

a FileType enum value representing the type of the file

getType()

Gets the type of the file

```
public FileType getType()
```

setPersistent(boolean persistentFile)

Sets whether the file should persist between sessions / ignition cycles

```
public void setPersistent(boolean persistentFile)
```

Parameters

persistentFile

a boolean value that indicates if the file is meant to persist between sessions / ignition cycles

isPersistent()

Gets whether the file should persist between sessions / ignition cycles

```
public boolean isPersistent()
```

setStaticIcon(boolean staticIcon)

Sets the the name of the static file. Static files comes pre-shipped with the head unit

```
public void setStaticIcon(boolean staticIcon)
```

Parameters

staticIcon

a StaticIconName enum value representing the name of a static file that comes pre-shipped with the head unit

isStaticIcon()

Gets the the name of the static file. Static files comes pre-shipped with the head unit

```
public boolean isStaticIcon()
```

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class SdlManager

Hierarchy

java.lang.Object

BaseSdlManager

Package

com.smartdevicelink.managers

Overview

SDLManager

This is the main point of contact between an application and SDL

It is broken down to these areas:

1. SDLManagerBuilder
2. ISdl Interface along with its overridden methods - This can be passed into attached managers
3. Sending Requests
4. Helper methods

Quick Jump

Constructors

- [SdlManager\(\)](#)

Methods

- [initialize\(\)](#)
- [dispose\(\)](#)
- [getPermissionManager\(\)](#)
- [getFileManager\(\)](#)
- [getVideoStreamManager\(\)](#)
- [getAudioStreamManager\(\)](#)
- [getScreenManager\(\)](#)
- [getLockScreenManager\(\)](#)
- [getSystemCapabilityManager\(\)](#)
- [getRegisterAppInterfaceResponse\(\)](#)
- [getCurrentHMIStatus\(\)](#)
- [getAuthToken\(\)](#)
- [getLockScreenConfig\(\)](#)
- [sendRPC\(RPCMessage message\)](#)
- [sendSequentialRPCs\(java.util.List<? extends RPCMessage> rpcs, OnMultipleRequestListener listener\)](#)
- [sendRPCs\(java.util.List<? extends RPCMessage> rpcs, OnMultipleRequestListener listener\)](#)

- `addOnRPCNotificationListener(FunctionID notificationId, OnRPCNotificationListener listener)`
- `removeOnRPCNotificationListener(FunctionID notificationId, OnRPCNotificationListener listener)`
- `addOnRPCRequestListener(FunctionID requestId, OnRPCRequestListener listener)`
- `removeOnRPCRequestListener(FunctionID requestId, OnRPCRequestListener listener)`
- `start()`
- `setProxy(com.smartdevicelink.proxy.SdlProxyBase proxy)`

Constructors

SdlManager()

```
public SdlManager()
```

Methods

initialize()

```
protected void initialize()
```

dispose()

```
public void dispose()
```

getPermissionManager()

Gets the PermissionManager.

Note: PermissionManager should be used only after SdlManager.start()
() CompletionListener callback is completed successfully.

```
public com.smartdevicelink.managers.permission.PermissionManager  
getPermissionManager()
```

getFileManager()

Gets the FileManager.

Note: FileManager should be used only after SdlManager.start()
CompletionListener callback is completed successfully.

```
public com.smartdevicelink.managers.file.FileManager getFileManager()
```

getVideoStreamManager()

Gets the VideoStreamManager.

The VideoStreamManager returned will only be not null if the registered app type is either NAVIGATION or PROJECTION. Once the VideoStreamManager is retrieved, its start() method will need to be called before use.

Note: VideoStreamManager should be used only after SdlManager.start() CompletionListener callback is completed successfully.

```
public com.smartdevicelink.managers.video.VideoStreamManager  
getVideoStreamManager()
```

getAudioStreamManager()

Gets the AudioStreamManager.

The AudioStreamManager returned will only be not null if the registered app type is either NAVIGATION or PROJECTION. Once the AudioStreamManager is retrieved, its start() method will need to be called before use.

Note: AudioStreamManager should be used only after SdlManager.start() CompletionListener callback is completed successfully.

```
public com.smartdevicelink.managers.audio.AudioStreamManager  
getAudioStreamManager()
```

getScreenManager()

Gets the ScreenManager.

Note: ScreenManager should be used only after SdlManager.start() CompletionListener callback is completed successfully.

```
public com.smartdevicelink.managers.screen.ScreenManager  
getScreenManager()
```

getLockScreenManager()

Gets the LockScreenManager.

Note: LockScreenManager should be used only after SdlManager.start() CompletionListener callback is completed successfully.

```
public com.smartdevicelink.managers.lockscreen.LockScreenManager  
getLockScreenManager()
```

getSystemCapabilityManager()

Gets the SystemCapabilityManager.

Note: SystemCapabilityManager should be used only after SdlManager.start() CompletionListener callback is completed successfully.

```
public SystemCapabilityManager getSystemCapabilityManager()
```

getRegisterAppInterfaceResponse()

Method to retrieve the RegisterAppInterface Response message that was sent back from the module. It contains various attributes about the connected module and can be used to adapt to different module types and their supported features.

```
public RegisterAppInterfaceResponse getRegisterAppInterfaceResponse()
```

getCurrentHMIStatus()

Get the current OnHMIStatus

```
public OnHMIStatus getCurrentHMIStatus()
```

getAuthToken()

Retrieves the auth token, if any, that was attached to the StartServiceACK for the RPC service from the module. For example, this should be used to login to a user account.

```
public java.lang.String getAuthToken()
```

getLockScreenConfig()

```
protected com.smartdevicelink.managers.lockscreen.LockScreenConfig  
getLockScreenConfig()
```

sendRPC(RPCMessage message)

Send RPC Message

```
public void sendRPC(RPCMessage message)
```

Parameters

message

RPCMessage

sendSequentialRPCs(java.util.List<? extends RPCMessage> rpcs, OnMultipleRequestListener listener)

Takes a list of RPCMessages and sends it to SDL in a synchronous fashion. Responses are captured through callback on OnMultipleRequestListener. For sending requests asynchronously, use sendRequests

NOTE: This will override any listeners on individual RPCs

ADDITIONAL NOTE: This only takes the type of RPCRequest for now, notifications and responses will be thrown out

```
public void sendSequentialRPCs(java.util.List<? extends RPCMessage>  
rpcs, OnMultipleRequestListener listener)
```

Parameters

rpcs

is the list of RPCMessages being sent

listener

listener for updates and completions

sendRPCs(java.util.List<? extends RPCMessage> rpcs, OnMultipleRequestListener listener)

Takes a list of RPCMessages and sends it to SDL. Responses are captured through callback on OnMultipleRequestListener. For sending requests synchronously, use sendSequentialRPCs

NOTE: This will override any listeners on individual RPCs

ADDITIONAL NOTE: This only takes the type of RPCRequest for now, notifications and responses will be thrown out

```
public void sendRPCs(java.util.List<? extends RPCMessage> rpcs,  
OnMultipleRequestListener listener)
```


Parameters

`rpcs`

is the list of RPCMessages being sent

`listener`

listener for updates and completions

`addOnRPCNotificationListener( FunctionID notificationId, OnRPCNotificationListener listener )`

Add an OnRPCNotificationListener

```
public void addOnRPCNotificationListener(FunctionID notificationId,  
OnRPCNotificationListener listener)
```

Parameters

notificationId

listener

listener that will be called when a notification is received

removeOnRPCNotificationListener (FunctionID notificationId, OnRPCNotificationListener listener)

Remove an OnRPCNotificationListener

```
public void removeOnRPCNotificationListener(FunctionID notificationId,  
OnRPCNotificationListener listener)
```

Parameters

notificationId

listener
listener that was previously added

addOnRPCRequestListener(FunctionID requestId, OnRPCRequestListener listener)

Add an OnRPCRequestListener

```
public void addOnRPCRequestListener(FunctionID requestId,  
OnRPCRequestListener listener)
```

Parameters

requestId
listener
listener that will be called when a request is received

removeOnRPCRequestListener(FunctionID requestId, OnRPCRequestListener listener)

Remove an OnRPCRequestListener

```
public void removeOnRPCRequestListener(FunctionID requestId,  
OnRPCRequestListener listener)
```

Parameters

requestId
listener listener that was previously added

start()

Starts up a SdlManager, and calls provided callback called once all BaseSubManagers are done setting up

```
public void start()
```

```
setProxy(  
com.smartdevicelink.proxy.SdlProxyBase  
proxy )
```

```
protected void setProxy(com.smartdevicelink.proxy.SdlProxyBase proxy  
)
```

Parameters

proxy

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class SdlManager.Builder

Hierarchy

java.lang.Object

Package

com.smartdevicelink.managers

Quick Jump

Constructors

- [Builder\(Context context, String appld, String appName, SdlManagerListener listener \)](#)

Methods

- [setAppId\(String appld\)](#)
- [setAppName\(String appName\)](#)
- [setShortAppName\(String shortAppName\)](#)
- [setMinimumProtocolVersion\(Version minimumProtocolVersion\)](#)
- [setMinimumRPCVersion\(Version minimumRPCVersion\)](#)
- [setLanguage\(Language hmiLanguage\)](#)
- [setDayColorScheme\(TemplateColorScheme dayColorScheme\)](#)
- [setNightColorScheme\(TemplateColorScheme nightColorScheme\)](#)
- [setLockScreenConfig\(LockScreenConfig lockScreenConfig\)](#)
- [setAppIcon\(SdlArtwork sdlArtwork\)](#)
- [setAppTypes\(java.util.Vector<AppHMIType> hmiTypes\)](#)
- [setVrSynonyms\(java.util.Vector<java.lang.String> vrSynonyms\)](#)
- [setTtsName\(java.util.Vector<TTSCheck> ttsChunks\)](#)
- [setTransportType\(BaseTransportConfig transport\)](#)
- [setContext\(Context context\)](#)

- `setSdlSecurity(java.util.List<java.lang.Class<? extends com.smartdevicelink.security.SdlSecurityBase>> secList)`
- `setManagerListener(SdlManagerListener listener)`
- `setRPCNotificationListeners(java.util.Map<FunctionID, OnRPCNotificationListener> listeners)`
- `build()`

Constructors

Builder(Context context, String applId, String appName, SdlManagerListener listener)

Builder for the SdlManager. Parameters in the constructor are required.

```
public Builder(Context context,String applId,String appName,  
SdlManagerListener listener)
```

Parameters

context

the current context

applId

the app's ID

appName

the app's name

listener

a SdlManagerListener object

Methods

setAppId(String appId)

Sets the App ID

```
public com.smartdevicelink.managers.SdlManager.Builder setAppId(  
String appId)
```

Parameters

appId

setAppName(String appName)

Sets the Application Name

```
public com.smartdevicelink.managers.SdlManager.Builder setAppName  
(String appName)
```

Parameters

appName

setShortAppName(String shortAppName)

Sets the Short Application Name

```
public com.smartdevicelink.managers.SdlManager.Builder  
setShortAppName(String shortAppName)
```

Parameters

shortAppName

setMinimumProtocolVersion(Version minimumProtocolVersion)

Sets the minimum protocol version that will be permitted to connect. If the protocol version of the head unit connected is below this version, the app will disconnect with an EndService protocol message and will not register.

```
public com.smartdevicelink.managers.SdlManager.Builder  
setMinimumProtocolVersion(Version minimumProtocolVersion)
```

Parameters

minimumProtocolVersion

the minimum Protocol spec version that should be accepted

setMinimumRPCVersion(Version minimumRPCVersion)

The minimum RPC version that will be permitted to connect. If the RPC version of the head unit connected is below this version, an UnregisterAppInterface will be sent.

```
public com.smartdevicelink.managers.SdlManager.Builder  
setMinimumRPCVersion(Version minimumRPCVersion)
```

Parameters

minimumRPCVersion

the minimum RPC spec version that should be accepted

setLanguage(Language hmiLanguage)

Sets the Language of the App

```
public com.smartdevicelink.managers.SdlManager.Builder setLanguage  
(Language hmiLanguage)
```

Parameters

hmiLanguage

the desired language to be used on the display/HMI of the connected module

setDayColorScheme(TemplateColorScheme dayColorScheme)

Sets the TemplateColorScheme for daytime

```
public com.smartdevicelink.managers.SdlManager.Builder  
setDayColorScheme(TemplateColorScheme dayColorScheme)
```

Parameters

dayColorScheme

color scheme that will be used (if supported) when the display is in a "Day Mode" or similar. Should comprise of colors that contrast well during the day under sunlight.

setNightColorScheme(TemplateColorScheme nightColorScheme)

Sets the TemplateColorScheme for nighttime

```
public com.smartdevicelink.managers.SdlManager.Builder  
setNightColorScheme(TemplateColorScheme nightColorScheme)
```

Parameters

nightColorScheme

color scheme that will be used (if supported) when the display is in a "Night Mode" or similar. Should comprise of colors that contrast well during the night and are not brighter than average.

setLockScreenConfig(LockScreenConfig lockScreenConfig)

Sets the LockScreenConfig for the session.

Note: If not set, the default configuration will be used.

```
public com.smartdevicelink.managers.SdlManager.Builder  
setLockScreenConfig(LockScreenConfig lockScreenConfig)
```

Parameters

lockScreenConfig
- configuration options

setIcon(SdlArtwork sdlArtwork)

Sets the icon for the app on head unit / In-Vehicle-Infotainment system

```
public com.smartdevicelink.managers.SdlManager.Builder setAppIcon(  
SdlArtwork sdlArtwork)
```

Parameters

sdlArtwork

the icon that will be used to represent this application on the connected module

setAppTypes(java.util.Vector<AppHMType> hmiTypes)

Sets the vector of AppHMType

Note: This should be an ordered list from most -> least relevant

```
public com.smartdevicelink.managers.SdlManager.Builder setAppTypes(  
java.util.Vector<AppHMType> hmiTypes)
```

Parameters

hmiTypes

HMI types that represent this application. For example, if the app is a music player, the MEDIA HMType should be included.

setVrSynonyms(java.util.Vector<java.lang.String> vrSynonyms)

Sets the voice recognition synonyms that can be used to identify this application.

```
public com.smartdevicelink.managers.SdlManager.Builder  
setVrSynonyms(java.util.Vector<java.lang.String> vrSynonyms)
```

Parameters

vrSynonyms

a vector of Strings that can be associated with this app. For example the app's name should be included as well as any phonetic spellings of the app name that might help the on-board VR system associated a users spoken word with the supplied synonyms.

setTtsName(java.util.Vector<TTSCChunk> ttsChunks)

Sets the Text-To-Speech Name of the application. These TTSChecks might be used by the module as an audio representation of the app's name.

```
public com.smartdevicelink.managers.SdlManager.Builder setTtsName(  
java.util.Vector<TTSCheck> ttsChecks)
```

Parameters

ttsChecks

the TTS checks that can represent this app's name

setTransportType(BaseTransportConfig transport)

This Object type may change with the transport refactor Sets the BaseTransportConfig

```
public com.smartdevicelink.managers.SdlManager.Builder  
setTransportType(BaseTransportConfig transport)
```

Parameters

transport

the type of transport that should be used for this SdlManager instance.

setContext(Context context)

Sets the Context

```
public com.smartdevicelink.managers.SdlManager.Builder setContext(  
Context context)
```

Parameters

context

```
setSdlSecurity(  
java.util.List<java.lang.Class<?  
extends  
com.smartdevicelink.security.SdlSecurityBase  
secList )
```

Sets the Security library

```
public com.smartdevicelink.managers.SdlManager.Builder  
setSdlSecurity(java.util.List<java.lang.Class<? extends com.  
smartdevicelink.security.SdlSecurityBase>> secList)
```

Parameters

secList

The list of security class(es)

```
setManagerListener(  
SdlManagerListener listener )
```

Set the SdlManager Listener

```
public com.smartdevicelink.managers.SdlManager.Builder  
setManagerListener(SdlManagerListener listener)
```

Parameters

listener

the listener

setRPCNotificationListeners(java.util.Map<FunctionID, OnRPCNotificationListener> listeners)

Set RPCNotification listeners. SdlManager will preload these listeners before any RPCs are sent/received.

```
public com.smartdevicelink.managers.SdlManager.Builder  
setRPCNotificationListeners(java.util.Map<FunctionID,  
OnRPCNotificationListener> listeners)
```

Parameters

listeners

a map of listeners that will be called when a notification is received. Key represents the FunctionID of the notification and value represents the listener

build()

```
public com.smartdevicelink.managers.SdlManager build()
```

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Interface SdlManagerListener

Package

com.smartdevicelink.managers

Quick Jump

Methods

- [onStart\(\)](#)
- [onDestroy\(\)](#)
- [onError\(String info,Exception e\)](#)

Methods

onStart()

Called when a manager is ready for use

```
public void onStart()
```

onDestroy()

Called when the manager is destroyed

```
public void onDestroy()
```

onError(String info, Exception e)

Called when the manager encounters an error

```
public void onError(String info,Exception e)
```

Parameters

info

info regarding the error

e

the exception

Class SdlProtocol

Hierarchy

java.lang.Object

SdlProtocolBase

Package

com.smartdevicelink.protocol

Quick Jump

Constructors

- [SdlProtocol\(ISdlProtocol iSdlProtocol, MultiplexTransportConfig config \)](#)

Constructors

SdlProtocol(ISdlProtocol
iSdlProtocol,
MultiplexTransportConfig config)


```
public SdlProtocol(ISdlProtocol iSdlProtocol, MultiplexTransportConfig config)
```

Parameters

iSdlProtocol
config

Methods

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class SdlProxyALM

Hierarchy

java.lang.Object

com.smartdevicelink.proxy.SdlProxyBase

Package

com.smartdevicelink.proxy

Quick Jump

Constructors

- SdlProxyALM(IProxyListenerALM listener, String appName, Boolean isMediaApp, Language languageDesired, Language hmiDisplayLanguageDesired, String appID)
- SdlProxyALM(Context context, IProxyListenerALM listener, String appName, Boolean isMediaApp, Language languageDesired, Language hmiDisplayLanguageDesired, String appID)
- SdlProxyALM(IProxyListenerALM listener, String appName, String ngnMediaScreenAppName, java.util.Vector<java.lang.String> vrSynonyms, Boolean isMediaApp, SdlMsgVersion sdlMsgVersion, Language languageDesired, Language hmiDisplayLanguageDesired, String appID, String autoActivateID)
- SdlProxyALM(Context context, IProxyListenerALM listener, String appName, String ngnMediaScreenAppName, java.util.Vector<java.lang.String> vrSynonyms, Boolean isMediaApp, SdlMsgVersion sdlMsgVersion, Language languageDesired, Language hmiDisplayLanguageDesired, String appID, String autoActivateID)
- SdlProxyALM(IProxyListenerALM listener, SdlProxyConfigurationResources sdlProxyConfigurationResources, String appName, String ngnMediaScreenAppName, java.util.Vector<java.lang.String> vrSynonyms, Boolean isMediaApp, SdlMsgVersion sdlMsgVersion, Language

- languageDesired, Language hmiDisplayLanguageDesired, String appId, String autoActivateID)
- SdlProxyALM(Context context, IProxyListenerALM listener, SdlProxyConfigurationResources sdlProxyConfigurationResources, String appName, String ngnMediaScreenAppName, java.util.Vector<java.lang.String> vrSynonyms, Boolean isMediaApp, SdlMsgVersion sdlMsgVersion, Language languageDesired, Language hmiDisplayLanguageDesired, String appId, String autoActivateID)
 - SdlProxyALM(IProxyListenerALM listener, String appName, String ngnMediaScreenAppName, java.util.Vector<java.lang.String> vrSynonyms, Boolean isMediaApp, SdlMsgVersion sdlMsgVersion, Language languageDesired, Language hmiDisplayLanguageDesired, String appId, String autoActivateID, boolean callbackToUiThread)
 - SdlProxyALM(Context context, IProxyListenerALM listener, String appName, String ngnMediaScreenAppName, java.util.Vector<java.lang.String> vrSynonyms, Boolean isMediaApp, SdlMsgVersion sdlMsgVersion, Language languageDesired, Language hmiDisplayLanguageDesired, String appId, String autoActivateID, boolean callbackToUiThread)
 - SdlProxyALM(IProxyListenerALM listener, SdlProxyConfigurationResources sdlProxyConfigurationResources, String appName, String ngnMediaScreenAppName, java.util.Vector<java.lang.String> vrSynonyms, Boolean isMediaApp, SdlMsgVersion sdlMsgVersion, Language languageDesired, Language hmiDisplayLanguageDesired, String appId, String autoActivateID, boolean callbackToUiThread)
 - SdlProxyALM(Context context, IProxyListenerALM listener, SdlProxyConfigurationResources sdlProxyConfigurationResources, String appName, String ngnMediaScreenAppName, java.util.Vector<java.lang.String> vrSynonyms, Boolean isMediaApp, SdlMsgVersion sdlMsgVersion, Language languageDesired, Language hmiDisplayLanguageDesired, String appId, String autoActivateID, boolean callbackToUiThread)
 - SdlProxyALM(IProxyListenerALM listener, SdlProxyConfigurationResources sdlProxyConfigurationResources, String appName, String ngnMediaScreenAppName, java.util.Vector<java.lang.String> vrSynonyms, Boolean isMediaApp, SdlMsgVersion sdlMsgVersion, Language languageDesired, Language hmiDisplayLanguageDesired, String appId, String autoActivateID, boolean callbackToUiThread, boolean preRegister)
 - SdlProxyALM(Context context, IProxyListenerALM listener, SdlProxyConfigurationResources sdlProxyConfigurationResources, String appName, String ngnMediaScreenAppName, java.util.Vector<java.lang.String> vrSynonyms, Boolean isMediaApp, SdlMsgVersion sdlMsgVersion, Language languageDesired, Language hmiDisplayLanguageDesired, String appId, String autoActivateID, boolean callbackToUiThread, boolean preRegister)
 - SdlProxyALM(IProxyListenerALM listener, String appName, Boolean isMediaApp, Language languageDesired, Language hmiDisplayLanguageDesired, String appId, BaseTransportConfig transportConfig)

- SdlProxyALM(IProxyListenerALM listener, String appName, String ngnMediaScreenAppName, java.util.Vector<java.lang.String> vrSynonyms, Boolean isMediaApp, SdlMsgVersion sdlMsgVersion, Language languageDesired, Language hmiDisplayLanguageDesired, String applID, String autoActivateID, TransportType transportType, BaseTransportConfig transportConfig)
- SdlProxyALM(IProxyListenerALM listener, SdlProxyConfigurationResources sdlProxyConfigurationResources, String appName, String ngnMediaScreenAppName, java.util.Vector<java.lang.String> vrSynonyms, Boolean isMediaApp, SdlMsgVersion sdlMsgVersion, Language languageDesired, Language hmiDisplayLanguageDesired, String applID, String autoActivateID, BaseTransportConfig transportConfig)
- SdlProxyALM(IProxyListenerALM listener, String appName, String ngnMediaScreenAppName, java.util.Vector<java.lang.String> vrSynonyms, Boolean isMediaApp, SdlMsgVersion sdlMsgVersion, Language languageDesired, Language hmiDisplayLanguageDesired, String applID, String autoActivateID, boolean callbackToUiThread, BaseTransportConfig transportConfig)
- SdlProxyALM(IProxyListenerALM listener, SdlProxyConfigurationResources sdlProxyConfigurationResources, String appName, String ngnMediaScreenAppName, java.util.Vector<java.lang.String> vrSynonyms, Boolean isMediaApp, SdlMsgVersion sdlMsgVersion, Language languageDesired, Language hmiDisplayLanguageDesired, String applID, String autoActivateID, boolean callbackToUiThread, BaseTransportConfig transportConfig)
- SdlProxyALM(IProxyListenerALM listener, SdlProxyConfigurationResources sdlProxyConfigurationResources, String appName, String ngnMediaScreenAppName, java.util.Vector<java.lang.String> vrSynonyms, Boolean isMediaApp, SdlMsgVersion sdlMsgVersion, Language languageDesired, Language hmiDisplayLanguageDesired, String applID, String autoActivateID, boolean callbackToUiThread, boolean preRegister, BaseTransportConfig transportConfig)
- SdlProxyALM(IProxyListenerALM listener, String appName, Boolean isMediaApp, Language languageDesired, Language hmiDisplayLanguageDesired, String applID, boolean callbackToUiThread, boolean preRegister)
- SdlProxyALM(Context context, IProxyListenerALM listener, String appName, Boolean isMediaApp, Language languageDesired, Language hmiDisplayLanguageDesired, String applID, boolean callbackToUiThread, boolean preRegister)
- SdlProxyALM(IProxyListenerALM listener, String appName, Boolean isMediaApp, String applID)
- SdlProxyALM(Context context, IProxyListenerALM listener, String appName, Boolean isMediaApp, String applID)
- SdlProxyALM(IProxyListenerALM listener, SdlProxyConfigurationResources sdlProxyConfigurationResources, String appName, Boolean isMediaApp, String applID)

- SdlProxyALM(Context context, IProxyListenerALM listener, SdlProxyConfigurationResources sdlProxyConfigurationResources, String appName, Boolean isMediaApp, String appId)
- SdlProxyALM(IProxyListenerALM listener, String appName, Boolean isMediaApp, String appId, BaseTransportConfig transportConfig)
- SdlProxyALM(IProxyListenerALM listener, String appName, Boolean isMediaApp, String appId, TemplateColorScheme dayColorScheme, TemplateColorScheme nightColorScheme, BaseTransportConfig transportConfig)
- SdlProxyALM(IProxyListenerALM listener, String appName, Boolean isMediaApp, String appId, boolean callbackToUiThread, boolean preRegister)
- SdlProxyALM(Context context, IProxyListenerALM listener, String appName, Boolean isMediaApp, String appId, boolean callbackToUiThread, boolean preRegister)
- SdlProxyALM(Service appService, IProxyListenerALM listener, SdlProxyConfigurationResources sdlProxyConfigurationResources, String appName, java.util.Vector<TTSCChunk> ttsName, String ngnMediaScreenAppName, java.util.Vector<java.lang.String> vrSynonyms, Boolean isMediaApp, SdlMsgVersion sdlMsgVersion, Language languageDesired, Language hmiDisplayLanguageDesired, String appId, String autoActivateID, boolean callbackToUiThread, boolean preRegister)
- SdlProxyALM(Service appService, IProxyListenerALM listener, SdlProxyConfigurationResources sdlProxyConfigurationResources, String appName, java.util.Vector<TTSCChunk> ttsName, String ngnMediaScreenAppName, java.util.Vector<java.lang.String> vrSynonyms, Boolean isMediaApp, SdlMsgVersion sdlMsgVersion, Language languageDesired, Language hmiDisplayLanguageDesired, String appId, String autoActivateID, boolean callbackToUiThread, boolean preRegister, BaseTransportConfig transportConfig)
- SdlProxyALM(IProxyListenerALM listener, SdlProxyConfigurationResources sdlProxyConfigurationResources, String appName, java.util.Vector<TTSCChunk> ttsName, String ngnMediaScreenAppName, java.util.Vector<java.lang.String> vrSynonyms, Boolean isMediaApp, SdlMsgVersion sdlMsgVersion, Language languageDesired, Language hmiDisplayLanguageDesired, String appId, String autoActivateID, boolean callbackToUiThread, boolean preRegister)
- SdlProxyALM(Context context, IProxyListenerALM listener, SdlProxyConfigurationResources sdlProxyConfigurationResources, String appName, java.util.Vector<TTSCChunk> ttsName, String ngnMediaScreenAppName, java.util.Vector<java.lang.String> vrSynonyms, Boolean isMediaApp, SdlMsgVersion sdlMsgVersion, Language languageDesired, Language hmiDisplayLanguageDesired, String appId, String autoActivateID, boolean callbackToUiThread, boolean preRegister)
- SdlProxyALM(IProxyListenerALM listener, SdlProxyConfigurationResources sdlProxyConfigurationResources, String appName, java.util.Vector<TTSCChunk> ttsName, String ngnMediaScreenAppName, java.util.Vector<java.lang.String> vrSynonyms, Boolean isMediaApp, SdlMsgVersion sdlMsgVersion, Language languageDesired, Language

- hmiDisplayLanguageDesired, String appId, String autoActivateID, boolean callbackToUIThread, boolean preRegister, BaseTransportConfig transportConfig)
- SdlProxyALM(IProxyListenerALM listener, SdlProxyConfigurationResources sdlProxyConfigurationResources, String appName, java.util.Vector<TTSCChunk> ttsName, String ngnMediaScreenAppName, java.util.Vector<java.lang.String> vrSynonyms, Boolean isMediaApp, SdlMsgVersion sdlMsgVersion, Language languageDesired, Language hmiDisplayLanguageDesired, java.util.Vector<AppHMIMType> appType, String appId, String autoActivateID, boolean callbackToUIThread, boolean preRegister)
 - SdlProxyALM(Context context, IProxyListenerALM listener, SdlProxyConfigurationResources sdlProxyConfigurationResources, String appName, java.util.Vector<TTSCChunk> ttsName, String ngnMediaScreenAppName, java.util.Vector<java.lang.String> vrSynonyms, Boolean isMediaApp, SdlMsgVersion sdlMsgVersion, Language languageDesired, Language hmiDisplayLanguageDesired, java.util.Vector<AppHMIMType> appType, String appId, String autoActivateID, boolean callbackToUIThread, boolean preRegister)
 - SdlProxyALM(IProxyListenerALM listener, SdlProxyConfigurationResources sdlProxyConfigurationResources, String appName, java.util.Vector<TTSCChunk> ttsName, String ngnMediaScreenAppName, java.util.Vector<java.lang.String> vrSynonyms, Boolean isMediaApp, SdlMsgVersion sdlMsgVersion, Language languageDesired, Language hmiDisplayLanguageDesired, java.util.Vector<AppHMIMType> appType, String appId, String autoActivateID, boolean callbackToUIThread, boolean preRegister, BaseTransportConfig transportConfig)
 - SdlProxyALM(Service appService, IProxyListenerALM listener, SdlProxyConfigurationResources sdlProxyConfigurationResources, String appName, java.util.Vector<TTSCChunk> ttsName, String ngnMediaScreenAppName, java.util.Vector<java.lang.String> vrSynonyms, Boolean isMediaApp, SdlMsgVersion sdlMsgVersion, Language languageDesired, Language hmiDisplayLanguageDesired, java.util.Vector<AppHMIMType> appType, String appId, String autoActivateID, boolean callbackToUIThread, boolean preRegister, BaseTransportConfig transportConfig)
 - SdlProxyALM(IProxyListenerALM listener, SdlProxyConfigurationResources sdlProxyConfigurationResources, String appName, java.util.Vector<TTSCChunk> ttsName, String ngnMediaScreenAppName, java.util.Vector<java.lang.String> vrSynonyms, Boolean isMediaApp, SdlMsgVersion sdlMsgVersion, Language languageDesired, Language hmiDisplayLanguageDesired, java.util.Vector<AppHMIMType> appType, String appId, String autoActivateID, boolean callbackToUIThread, boolean preRegister, String sHashID, BaseTransportConfig transportConfig)
 - SdlProxyALM(Service appService, IProxyListenerALM listener, SdlProxyConfigurationResources sdlProxyConfigurationResources, String appName, java.util.Vector<TTSCChunk> ttsName, String ngnMediaScreenAppName, java.util.Vector<java.lang.String>

- vrSynonyms, Boolean isMediaApp, SdlMsgVersion sdlMsgVersion, Language languageDesired, Language hmiDisplayLanguageDesired, java.util.Vector<AppHMIType> appType, String applID, String autoActivateID, boolean callbackToUiThread, boolean preRegister, String sHashID, BaseTransportConfig transportConfig)
- SdlProxyALM(Service appService, IProxyListenerALM listener, SdlProxyConfigurationResources sdlProxyConfigurationResources, String appName, java.util.Vector<TTSCChunk> ttsName, String ngnMediaScreenAppName, java.util.Vector<java.lang.String> vrSynonyms, Boolean isMediaApp, SdlMsgVersion sdlMsgVersion, Language languageDesired, Language hmiDisplayLanguageDesired, java.util.Vector<AppHMIType> appType, String applID, String autoActivateID, TemplateColorScheme dayColorScheme, TemplateColorScheme nightColorScheme, boolean callbackToUiThread, boolean preRegister, String sHashID, BaseTransportConfig transportConfig)

Methods

- resetProxy()
- getButtonCapabilities() **Deprecated**
- getSoftButtonCapabilities() **Deprecated**
- getPresetBankCapabilities() **Deprecated**
- getProxyVersionInfo()
- getDisplayCapabilities() **Deprecated**
- getHmiZoneCapabilities() **Deprecated**
- getSpeechCapabilities() **Deprecated**
- getPrerecordedSpeech()
- getSdlLanguage()
- getHmiDisplayLanguage()
- getSdlMsgVersion()
- getVrCapabilities() **Deprecated**
- getVehicleType()
- getAudioPassThruCapabilities() **Deprecated**
- getSupportedDiagModes()
- getHmiCapabilities() **Deprecated**
- getSystemSoftwareVersion()

- `isAppResumeSuccess()`

Fields

Inherited Fields

From Class	Fields
<code>com.smartdevicelink.proxy.SdlProxyBase</code>	<code>TAG</code> , <code>MAX_SUPPORTED_RPC_VERSION</code> , <code>_appService</code> , <code>_advancedLifecycleManagementEnabled</code> , <code>_applInterfaceRegisterd</code> , <code>_preRegisterd</code> , <code>_haveReceivedFirstFocusLevel</code> , <code>_haveReceivedFirstFocusLevelFull</code> , <code>_proxyDisposed</code> , <code>_sdlConnectionState</code> , <code>_sdlInterfaceAvailability</code> , <code>_hmiLevel</code> , <code>lastHmiStatus</code> , <code>_audioStreamingState</code> , <code>_sdlMsgVersion</code> , <code>_autoActivateIdReturned</code> , <code>_sdlLanguage</code> , <code>_hmiDisplayLanguage</code> , <code>_prerecordedSpeech</code> , <code>_vehicleType</code> , <code>_systemSoftwareVersion</code> , <code>_diagModes</code> , <code>firstTimeFull</code> , <code>_proxyVersionInfo</code> , <code>_bResumeSuccess</code> , <code>_secList</code> , <code>_systemCapabilityManager</code> , <code>_iconResumed</code> , <code>raiResponse</code> , <code>protocolVersion</code> , <code>rpcSpecVersion</code> , <code>rpcResponseListeners</code> , <code>rpcNotificationListeners</code> , <code>rpcRequestListeners</code> , <code>rpcListeners</code> , <code>manager</code> , <code>authToken</code>

Constructors

SdlProxyALM(IProxyListenerALM listener, String appName, Boolean isMediaApp, Language languageDesired, Language hmiDisplayLanguageDesired, String appId)

```
public SdlProxyALM(IProxyListenerALM listener,String appName,Boolean isMediaApp,Language languageDesired,Language hmiDisplayLanguageDesired,String appId)
```

Parameters

listener

- Reference to the object in the App listening to callbacks from SDL.

appName

- Name of the application displayed on SDL.

isMediaApp

- Indicates if the app is a media application.

languageDesired
hmiDisplayLanguageDesired
appID

SdlProxyALM(Context context,
IProxyListenerALM listener,
String appName, Boolean
isMediaApp, Language
languageDesired, Language
hmiDisplayLanguageDesired,
String appID)

Constructor for the SdlProxy object, the proxy for communicating between the App and SDL Takes advantage of the advanced lifecycle management.

```
public SdlProxyALM(Context context,IProxyListenerALM listener,String  
appName,Boolean isMediaApp,Language languageDesired,Language  
hmiDisplayLanguageDesired,String appID)
```

Parameters

context

- Used to create a multiplexing transport config

listener

- Reference to the object in the App listening to callbacks from SDL.

appName

- Name of the application displayed on SDL.

isMediaApp

- Indicates if the app is a media application.

languageDesired

hmiDisplayLanguageDesired

appID

SdlProxyALM(IProxyListenerALM
listener, String appName, String
ngnMediaScreenAppName,
java.util.Vector<java.lang.String>
vrSynonyms, Boolean
isMediaApp, SdlMsgVersion
sdlMsgVersion, Language
languageDesired, Language
hmiDisplayLanguageDesired,
String appId, String
autoActivateID)

```
public SdlProxyALM(IProxyListenerALM listener,String appName,String  
ngnMediaScreenAppName,java.util.Vector<java.lang.String>  
vrSynonyms,Boolean isMediaApp,SdlMsgVersion sdlMsgVersion,  
Language languageDesired,Language hmiDisplayLanguageDesired,  
String appId,String autoActivateID)
```

Parameters

listener

- Reference to the object in the App listening to callbacks from SDL.

appName

- Name of the application displayed on SDL.

ngnMediaScreenAppName

- Name of the application displayed on SDL for Navigation equipped vehicles. Limited to five characters.

vrSynonyms

- A vector of strings, all of which can be used as voice commands to

isMediaApp

- Indicates if the app is a media application.

sdIMsgVersion

- Indicates the version of SDL SmartDeviceLink Messages desired. Must be less than or equal to the version of SDL SmartDeviceLink running on the vehicle.

languageDesired

- Indicates the language desired for the SDL interface.

hmiDisplayLanguageDesired

appID

autoActivateID

- ID used to re-register previously registered application.

```
SdlProxyALM( Context context,  
IProxyListenerALM listener,  
String appName, String  
ngnMediaScreenAppName,  
java.util.Vector<java.lang.String>  
vrSynonyms, Boolean  
isMediaApp, SdlMsgVersion  
sdlMsgVersion, Language  
languageDesired, Language  
hmiDisplayLanguageDesired,  
String appId, String  
autoActivateID )
```

Constructor for the SdlProxy object, the proxy for communicating between the App and SDL. Takes advantage of the advanced lifecycle management.

```
public SdlProxyALM(Context context,IProxyListenerALM listener,String  
appName,String ngnMediaScreenAppName,java.util.Vector<java.lang.  
String> vrSynonyms,Boolean isMediaApp,SdlMsgVersion sdlMsgVersion,  
Language languageDesired,Language hmiDisplayLanguageDesired,  
String appId,String autoActivateID)
```

Parameters

context

- Used to create a multiplexing transport config

listener

- Reference to the object in the App listening to callbacks from SDL.

appName

- Name of the application displayed on SDL.

ngnMediaScreenAppName

- Name of the application displayed on SDL for Navigation equipped vehicles. Limited to five characters.

vrSynonyms

- A vector of strings, all of which can be used as voice commands to

isMediaApp

- Indicates if the app is a media application.

sdIMsgVersion

- Indicates the version of SDL SmartDeviceLink Messages desired. Must be less than or equal to the version of SDL SmartDeviceLink running on the vehicle.

languageDesired

- Indicates the language desired for the SDL interface.

hmiDisplayLanguageDesired

appID

autoActivateID

- ID used to re-register previously registered application.

SdlProxyALM(IProxyListenerALM
listener,
SdlProxyConfigurationResources
sdlProxyConfigurationResources,
String appName, String
ngnMediaScreenAppName,
java.util.Vector<java.lang.String>
vrSynonyms, Boolean
isMediaApp, SdlMsgVersion
sdlMsgVersion, Language
languageDesired, Language
hmiDisplayLanguageDesired,
String appId, String
autoActivateID)

```
public SdlProxyALM(IProxyListenerALM listener,  
SdlProxyConfigurationResources sdlProxyConfigurationResources,String  
appName,String ngnMediaScreenAppName,java.util.Vector<java.lang.  
String> vrSynonyms,Boolean isMediaApp,SdlMsgVersion sdlMsgVersion,  
Language languageDesired,Language hmiDisplayLanguageDesired,  
String appId,String autoActivateID)
```

Parameters

listener

- Reference to the object in the App listening to callbacks from SDL.

sdIProxyConfigurationResources

appName

- Name of the application displayed on SDL.

ngnMediaScreenAppName

- Name of the application displayed on SDL for Navigation equipped vehicles. Limited to five characters.

vrSynonyms

- A vector of strings, all of which can be used as voice commands to

isMediaApp

- Indicates if the app is a media application.

sdIMsgVersion

- Indicates the version of SDL SmartDeviceLink Messages desired. Must be less than or equal to the version of SDL SmartDeviceLink running on the vehicle.

languageDesired

- Indicates the language desired for the SDL interface.

hmiDisplayLanguageDesired

appID

autoActivateID

- ID used to re-register previously registered application.

SdlProxyALM(Context context,
IProxyListenerALM listener,
SdlProxyConfigurationResources
sdlProxyConfigurationResources,
String appName, String
ngnMediaScreenAppName,
java.util.Vector<java.lang.String>
vrSynonyms, Boolean
isMediaApp, SdlMsgVersion
sdlMsgVersion, Language
languageDesired, Language
hmiDisplayLanguageDesired,
String appId, String
autoActivateID)

Constructor for the SdlProxy object, the proxy for communicating between the App and SDL. Takes advantage of the advanced lifecycle management.

```
public SdlProxyALM(Context context,IProxyListenerALM listener,  
SdlProxyConfigurationResources sdlProxyConfigurationResources,String  
appName,String ngnMediaScreenAppName,java.util.Vector<java.lang.  
String> vrSynonyms,Boolean isMediaApp,SdlMsgVersion sdlMsgVersion,  
Language languageDesired,Language hmiDisplayLanguageDesired,  
String appId,String autoActivateID)
```

Parameters

context

- Used to create a multiplexing transport config

listener

- Reference to the object in the App listening to callbacks from SDL.

sdIProxyConfigurationResources

appName

- Name of the application displayed on SDL.

ngnMediaScreenAppName

- Name of the application displayed on SDL for Navigation equipped vehicles. Limited to five characters.

vrSynonyms

- A vector of strings, all of which can be used as voice commands to

isMediaApp

- Indicates if the app is a media application.

sdIMsgVersion

- Indicates the version of SDL SmartDeviceLink Messages desired. Must be less than or equal to the version of SDL SmartDeviceLink running on the vehicle.

languageDesired

- Indicates the language desired for the SDL interface.

hmiDisplayLanguageDesired

appID

autoActivateID

- ID used to re-register previously registered application.

SdlProxyALM(IProxyListenerALM
listener, String appName, String
ngnMediaScreenAppName,
java.util.Vector<java.lang.String>
vrSynonyms, Boolean
isMediaApp, SdlMsgVersion
sdlMsgVersion, Language
languageDesired, Language
hmiDisplayLanguageDesired,
String appId, String
autoActivateID, boolean
callbackToUIThread)

```
public SdlProxyALM(IProxyListenerALM listener,String appName,String  
ngnMediaScreenAppName,java.util.Vector<java.lang.String>  
vrSynonyms,Boolean isMediaApp,SdlMsgVersion sdlMsgVersion,  
Language languageDesired,Language hmiDisplayLanguageDesired,  
String appId,String autoActivateID,boolean callbackToUIThread)
```

Parameters

listener

- Reference to the object in the App listening to callbacks from SDL.

appName

- Name of the application displayed on SDL.

ngnMediaScreenAppName

- Name of the application displayed on SDL for Navigation equipped vehicles. Limited to five characters.

vrSynonyms

- A vector of strings, all of which can be used as voice commands to

isMediaApp

- Indicates if the app is a media application.

sdIMsgVersion

- Indicates the version of SDL SmartDeviceLink Messages desired. Must be less than or equal to the version of SDL SmartDeviceLink running on the vehicle.

languageDesired

- Indicates the language desired for the SDL interface.

hmiDisplayLanguageDesired

appID

autoActivateID

- ID used to re-register previously registered application.

callbackToUiThread

- If true, all callbacks will occur on the UI thread.

SdlProxyALM(Context context,
IProxyListenerALM listener,
String appName, String
ngnMediaScreenAppName,
java.util.Vector<java.lang.String>
vrSynonyms, Boolean
isMediaApp, SdlMsgVersion
sdlMsgVersion, Language
languageDesired, Language
hmiDisplayLanguageDesired,
String appId, String
autoActivateID, boolean
callbackToUiThread)

Constructor for the SdlProxy object, the proxy for communicating between the App and SDL. Takes advantage of the advanced lifecycle management.

```
public SdlProxyALM(Context context,IProxyListenerALM listener,String  
appName,String ngnMediaScreenAppName,java.util.Vector<java.lang.  
String> vrSynonyms,Boolean isMediaApp,SdlMsgVersion sdlMsgVersion,  
Language languageDesired,Language hmiDisplayLanguageDesired,  
String applD,String autoActivateID,boolean callbackToUiThread)
```

Parameters

context

- Used to create a multiplexing transport config

listener

- Reference to the object in the App listening to callbacks from SDL.

appName

- Name of the application displayed on SDL.

ngnMediaScreenAppName

- Name of the application displayed on SDL for Navigation equipped vehicles. Limited to five characters.

vrSynonyms

- A vector of strings, all of which can be used as voice commands to

isMediaApp

- Indicates if the app is a media application.

sdIMsgVersion

- Indicates the version of SDL SmartDeviceLink Messages desired. Must be less than or equal to the version of SDL SmartDeviceLink running on the vehicle.

languageDesired

- Indicates the language desired for the SDL interface.

hmiDisplayLanguageDesired

appID

autoActivateID

- ID used to re-register previously registered application.

callbackToUIThread

- If true, all callbacks will occur on the UI thread.

SdlProxyALM(IProxyListenerALM
listener,
SdlProxyConfigurationResources
sdlProxyConfigurationResources,
String appName, String
ngnMediaScreenAppName,
java.util.Vector<java.lang.String>
vrSynonyms, Boolean
isMediaApp, SdlMsgVersion
sdlMsgVersion, Language
languageDesired, Language
hmiDisplayLanguageDesired,
String appID, String
autoActivateID, boolean
callbackToUIThread)

```
public SdlProxyALM(IProxyListenerALM listener,  
SdlProxyConfigurationResources sdlProxyConfigurationResources,String  
appName,String ngnMediaScreenAppName,java.util.Vector<java.lang.  
String> vrSynonyms,Boolean isMediaApp,SdlMsgVersion sdlMsgVersion,  
Language languageDesired,Language hmiDisplayLanguageDesired,  
String appID,String autoActivateID,boolean callbackToUIThread)
```

Parameters

listener

- Reference to the object in the App listening to callbacks from SDL.

sdIProxyConfigurationResources

appName

- Name of the application displayed on SDL.

ngnMediaScreenAppName

- Name of the application displayed on SDL for Navigation equipped vehicles. Limited to five characters.

vrSynonyms

- A vector of strings, all of which can be used as voice commands to

isMediaApp

- Indicates if the app is a media application.

sdIMsgVersion

- Indicates the version of SDL SmartDeviceLink Messages desired. Must be less than or equal to the version of SDL SmartDeviceLink running on the vehicle.

languageDesired

- Indicates the language desired for the SDL interface.

hmiDisplayLanguageDesired

appID

autoActivateID

- ID used to re-register previously registered application.

callbackToUIThread

- If true, all callbacks will occur on the UI thread.

SdlProxyALM(Context context,
IProxyListenerALM listener,
SdlProxyConfigurationResources
sdlProxyConfigurationResources,
String appName, String
ngnMediaScreenAppName,
java.util.Vector<java.lang.String>
vrSynonyms, Boolean
isMediaApp, SdlMsgVersion
sdlMsgVersion, Language
languageDesired, Language
hmiDisplayLanguageDesired,
String appId, String
autoActivateID, boolean
callbackToUIThread)

Constructor for the SdlProxy object, the proxy for communicating between the App and SDL. Takes advantage of the advanced lifecycle management.

```
public SdlProxyALM(Context context,IProxyListenerALM listener,  
SdlProxyConfigurationResources sdlProxyConfigurationResources,String  
appName,String ngnMediaScreenAppName,java.util.Vector<java.lang.  
String> vrSynonyms,Boolean isMediaApp,SdlMsgVersion sdlMsgVersion,  
Language languageDesired,Language hmiDisplayLanguageDesired,  
String appId,String autoActivateID,boolean callbackToUIThread)
```

Parameters

context

- Used to create a multiplexing transport config

listener

- Reference to the object in the App listening to callbacks from SDL.

sdIProxyConfigurationResources

appName

- Name of the application displayed on SDL.

ngnMediaScreenAppName

- Name of the application displayed on SDL for Navigation equipped vehicles. Limited to five characters.

vrSynonyms

- A vector of strings, all of which can be used as voice commands to

isMediaApp

- Indicates if the app is a media application.

sdMsgVersion

- Indicates the version of SDL SmartDeviceLink Messages desired. Must be less than or equal to the version of SDL SmartDeviceLink running on the vehicle.

languageDesired

- Indicates the language desired for the SDL interface.

hmiDisplayLanguageDesired

appID

autoActivateID

- ID used to re-register previously registered application.

callbackToUIThread

- If true, all callbacks will occur on the UI thread.

SdlProxyALM(IProxyListenerALM
listener,
SdlProxyConfigurationResources
sdlProxyConfigurationResources,
String appName, String
ngnMediaScreenAppName,
java.util.Vector<java.lang.String>
vrSynonyms, Boolean
isMediaApp, SdlMsgVersion
sdlMsgVersion, Language
languageDesired, Language
hmiDisplayLanguageDesired,
String appID, String
autoActivateID, boolean
callbackToUIThread, boolean
preRegister)

```
public SdlProxyALM(IProxyListenerALM listener,  
SdlProxyConfigurationResources sdlProxyConfigurationResources,String  
appName,String ngnMediaScreenAppName,java.util.Vector<java.lang.  
String> vrSynonyms,Boolean isMediaApp,SdlMsgVersion sdlMsgVersion,  
Language languageDesired,Language hmiDisplayLanguageDesired,  
String appID,String autoActivateID,boolean callbackToUIThread,boolean  
preRegister)
```

Parameters

listener

sdlProxyConfigurationResources

appName

ngnMediaScreenAppName

vrSynonyms

isMediaApp

sdlMsgVersion

languageDesired

hmiDisplayLanguageDesired

applID

autoActivateID

callbackToUIThread

preRegister

SdlProxyALM(Context context,
IProxyListenerALM listener,
SdlProxyConfigurationResources
sdlProxyConfigurationResources,
String appName, String
ngnMediaScreenAppName,
java.util.Vector<java.lang.String>
vrSynonyms, Boolean
isMediaApp, SdlMsgVersion
sdlMsgVersion, Language
languageDesired, Language
hmiDisplayLanguageDesired,
String appId, String
autoActivateID, boolean
callbackToUIThread, boolean
preRegister)

```
public SdlProxyALM(Context context,IProxyListenerALM listener,  
SdlProxyConfigurationResources sdlProxyConfigurationResources,String  
appName,String ngnMediaScreenAppName,java.util.Vector<java.lang.  
String> vrSynonyms,Boolean isMediaApp,SdlMsgVersion sdlMsgVersion,  
Language languageDesired,Language hmiDisplayLanguageDesired,  
String appId,String autoActivateID,boolean callbackToUIThread,boolean  
preRegister)
```

Parameters

context
listener
sdIProxyConfigurationResources
appName
ngnMediaScreenAppName
vrSynonyms
isMediaApp
sdIMsgVersion

languageDesired

hmiDisplayLanguageDesired

appId

autoActivateID

callbackToUIThread

preRegister

SdlProxyALM(IProxyListenerALM listener, String appName, Boolean isMediaApp, Language languageDesired, Language hmiDisplayLanguageDesired, String appId, BaseTransportConfig transportConfig)

Constructor for the SdlProxy object, the proxy for communicating between the App and SDL via specified transport. Takes advantage of the advanced lifecycle management.

```
public SdlProxyALM(IProxyListenerALM listener,String appName,Boolean isMediaApp,Language languageDesired,Language hmiDisplayLanguageDesired,String appId,BaseTransportConfig transportConfig)
```

Parameters

listener

Reference to the object in the App listening to callbacks from SDL.

appName

Name of the application displayed on SDL.

isMediaApp

Indicates if the app is a media application.

languageDesired

hmiDisplayLanguageDesired

appID

transportConfig

Initial configuration for transport.

```
SdlProxyALM( IProxyListenerALM
listener, String appName, String
ngnMediaScreenAppName,
java.util.Vector<java.lang.String>
vrSynonyms, Boolean
isMediaApp, SdlMsgVersion
sdlMsgVersion, Language
languageDesired, Language
hmiDisplayLanguageDesired,
String appId, String
autoActivateID, TransportType
transportType,
BaseTransportConfig
transportConfig )
```

Constructor for the SdlProxy object, the proxy for communicating between the App and SDL via specified transport. Takes advantage of the advanced lifecycle management.

```
public SdlProxyALM(IProxyListenerALM listener,String appName,String
ngnMediaScreenAppName,java.util.Vector<java.lang.String>
vrSynonyms,Boolean isMediaApp,SdlMsgVersion sdlMsgVersion,
Language languageDesired,Language hmiDisplayLanguageDesired,
String appId,String autoActivateID,TransportType transportType,
BaseTransportConfig transportConfig)
```

Parameters

listener

Reference to the object in the App listening to callbacks from SDL.

appName

Name of the application displayed on SDL.

ngnMediaScreenAppName

Name of the application displayed on SDL for Navigation equipped vehicles. Limited to five characters.

vrSynonyms

A vector of strings, all of which can be used as voice commands to

isMediaApp

Indicates if the app is a media application.

sdIMsgVersion

Indicates the version of SDL SmartDeviceLink Messages desired. Must be less than or equal to the version of SDL SmartDeviceLink running on the vehicle.

languageDesired

Indicates the language desired for the SDL interface.

hmiDisplayLanguageDesired

applID

autoActivateID

ID used to re-register previously registered application.

transportType

transportConfig

Initial configuration for transport.

```
SdlProxyALM( IProxyListenerALM
listener,
SdlProxyConfigurationResources
sdlProxyConfigurationResources,
String appName, String
ngnMediaScreenAppName,
java.util.Vector<java.lang.String>
vrSynonyms, Boolean
isMediaApp, SdlMsgVersion
sdlMsgVersion, Language
languageDesired, Language
hmiDisplayLanguageDesired,
String appId, String
autoActivateId,
BaseTransportConfig
transportConfig )
```

Constructor for the SdlProxy object, the proxy for communicating between the App and SDL via specified transport. Takes advantage of the advanced lifecycle management.

```
public SdlProxyALM(IProxyListenerALM listener,  
SdlProxyConfigurationResources sdlProxyConfigurationResources,String  
appName,String ngnMediaScreenAppName,java.util.Vector<java.lang.  
String> vrSynonyms,Boolean isMediaApp,SdlMsgVersion sdlMsgVersion,  
Language languageDesired,Language hmiDisplayLanguageDesired,  
String applD,String autoActivateID,BaseTransportConfig transportConfig  
)
```

Parameters

listener

Reference to the object in the App listening to callbacks from SDL.

sdlProxyConfigurationResources

appName

Name of the application displayed on SDL.

ngnMediaScreenAppName

Name of the application displayed on SDL for Navigation equipped vehicles. Limited to five characters.

vrSynonyms

A vector of strings, all of which can be used as voice commands to

isMediaApp

Indicates if the app is a media application.

sdIMsgVersion

Indicates the version of SDL SmartDeviceLink Messages desired. Must be less than or equal to the version of SDL SmartDeviceLink running on the vehicle.

languageDesired

Indicates the language desired for the SDL interface.

hmiDisplayLanguageDesired

appID

autoActivateID

ID used to re-register previously registered application.

transportConfig

Initial configuration for transport.

```
SdlProxyALM( IProxyListenerALM
listener, String appName, String
ngnMediaScreenAppName,
java.util.Vector<java.lang.String>
vrSynonyms, Boolean
isMediaApp, SdlMsgVersion
sdlMsgVersion, Language
languageDesired, Language
hmiDisplayLanguageDesired,
String appId, String
autoActivateID, boolean
callbackToUIThread,
BaseTransportConfig
transportConfig )
```

Constructor for the SdlProxy object, the proxy for communicating between the App and SDL via specified transport. Takes advantage of the advanced lifecycle management.

```
public SdlProxyALM(IProxyListenerALM listener,String appName,String
ngnMediaScreenAppName,java.util.Vector<java.lang.String>
vrSynonyms,Boolean isMediaApp,SdlMsgVersion sdlMsgVersion,
Language languageDesired,Language hmiDisplayLanguageDesired,
String appId,String autoActivateID,boolean callbackToUIThread,
BaseTransportConfig transportConfig)
```


Parameters

listener

Reference to the object in the App listening to callbacks from SDL.

appName

Name of the application displayed on SDL.

ngnMediaScreenAppName

Name of the application displayed on SDL for Navigation equipped vehicles. Limited to five characters.

vrSynonyms

A vector of strings, all of which can be used as voice commands to

isMediaApp

Indicates if the app is a media application.

sdIMsgVersion

Indicates the version of SDL SmartDeviceLink Messages desired. Must be less than or equal to the version of SDL SmartDeviceLink running on the vehicle.

languageDesired

Indicates the language desired for the SDL interface.

hmiDisplayLanguageDesired

applID

autoActivateID

ID used to re-register previously registered application.

callbackToUIThread

If true, all callbacks will occur on the UI thread.

transportConfig

Initial configuration for transport.

```
SdlProxyALM( IProxyListenerALM
listener,
SdlProxyConfigurationResources
sdlProxyConfigurationResources,
String appName, String
ngnMediaScreenAppName,
java.util.Vector<java.lang.String>
vrSynonyms, Boolean
isMediaApp, SdlMsgVersion
sdlMsgVersion, Language
languageDesired, Language
hmiDisplayLanguageDesired,
String appId, String
autoActivateID, boolean
callbackToUIThread,
BaseTransportConfig
transportConfig )
```

Constructor for the SdlProxy object, the proxy for communicating between the App and SDL via specified transport. Takes advantage of the advanced lifecycle management.

```
public SdlProxyALM(IProxyListenerALM listener,  
SdlProxyConfigurationResources sdlProxyConfigurationResources,String  
appName,String ngnMediaScreenAppName,java.util.Vector<java.lang.  
String> vrSynonyms,Boolean isMediaApp,SdlMsgVersion sdlMsgVersion,  
Language languageDesired,Language hmiDisplayLanguageDesired,  
String appId,String autoActivateId,boolean callbackToUiThread,  
BaseTransportConfig transportConfig)
```

Parameters

listener

Reference to the object in the App listening to callbacks from SDL.

sdlProxyConfigurationResources

appName

Name of the application displayed on SDL.

ngnMediaScreenAppName

Name of the application displayed on SDL for Navigation equipped vehicles. Limited to five characters.

vrSynonyms

A vector of strings, all of which can be used as voice commands too

isMediaApp

Indicates if the app is a media application.

sdIMsgVersion

Indicates the version of SDL SmartDeviceLink Messages desired. Must be less than or equal to the version of SDL SmartDeviceLink running on the vehicle.

languageDesired

Indicates the language desired for the SDL interface.

hmiDisplayLanguageDesired

appID

autoActivateID

ID used to re-register previously registered application.

callbackToUIThread

If true, all callbacks will occur on the UI thread.

transportConfig

Initial configuration for transport.

```
SdlProxyALM( IProxyListenerALM
listener,
SdlProxyConfigurationResources
sdlProxyConfigurationResources,
String appName, String
ngnMediaScreenAppName,
java.util.Vector<java.lang.String>
vrSynonyms, Boolean
isMediaApp, SdlMsgVersion
sdlMsgVersion, Language
languageDesired, Language
hmiDisplayLanguageDesired,
String appId, String
autoActivateID, boolean
callbackToUIThread, boolean
preRegister, BaseTransportConfig
transportConfig )
```

Constructor for the SdlProxy object, the proxy for communicating between the App and SDL via specified transport. Takes advantage of the advanced lifecycle management.

```
public SdlProxyALM(IProxyListenerALM listener,  
SdlProxyConfigurationResources sdlProxyConfigurationResources,String  
appName,String ngnMediaScreenAppName,java.util.Vector<java.lang.  
String> vrSynonyms,Boolean isMediaApp,SdlMsgVersion sdlMsgVersion,  
Language languageDesired,Language hmiDisplayLanguageDesired,  
String appId,String autoActivateId,boolean callbackToUiThread,boolean  
preRegister,BaseTransportConfig transportConfig)
```

Parameters

listener

Reference to the object in the App listening to callbacks from SDL.

sdlProxyConfigurationResources

Proxy configuration resources.

appName

Name of the application displayed on SDL.

ngnMediaScreenAppName

Name of the application displayed on SDL for Navigation equipped vehicles. Limited to five characters.

vrSynonyms

A vector of strings, all of which can be used as voice commands too

isMediaApp

Indicates if the app is a media application.

sdIMsgVersion

Indicates the version of SDL SmartDeviceLink Messages desired. Must be less than or equal to the version of SDL SmartDeviceLink running on the vehicle.

languageDesired

Indicates the language desired for the SDL interface.

hmiDisplayLanguageDesired

Desired language in HMI.

applID

Identifier of the client application.

autoActivateID

ID used to re-register previously registered application.

callbackToUIThread

If true, all callbacks will occur on the UI thread.

preRegister

Flag that indicates that client should be pre-registered or not

transportConfig

Initial configuration for transport.

**SdlProxyALM(IProxyListenerALM
listener, String appName,
Boolean isMediaApp, Language
languageDesired, Language
hmiDisplayLanguageDesired,
String appId, boolean
callbackToUiThread, boolean
preRegister)**

```
public SdlProxyALM(IProxyListenerALM listener,String appName,Boolean  
isMediaApp,Language languageDesired,Language  
hmiDisplayLanguageDesired,String appId,boolean callbackToUiThread,  
boolean preRegister)
```

Parameters

listener

Reference to the object in the App listening to callbacks from SDL.

appName

Name of the application displayed on SDL.

isMediaApp

Indicates if the app is a media application.

languageDesired

Indicates the language desired for the SDL interface.

hmiDisplayLanguageDesired

Desired language in HMI.

appID

Identifier of the client application.

callbackToUIThread

If true, all callbacks will occur on the UI thread.

preRegister

Flag that indicates that client should be pre-registered or not

SdlProxyALM(Context context,
IProxyListenerALM listener,
String appName, Boolean
isMediaApp, Language
languageDesired, Language
hmiDisplayLanguageDesired,
String appId, boolean
callbackToUiThread, boolean
preRegister)

Constructor for the SdlProxy object, the proxy for communicating between the App and SDL via specified transport. Takes advantage of the advanced lifecycle management.

```
public SdlProxyALM(Context context,IProxyListenerALM listener,String  
appName,Boolean isMediaApp,Language languageDesired,Language  
hmiDisplayLanguageDesired,String appId,boolean callbackToUiThread,  
boolean preRegister)
```

Parameters

context

- Used to create a multiplexing transport config

listener

Reference to the object in the App listening to callbacks from SDL.

appName

Name of the application displayed on SDL.

isMediaApp

Indicates if the app is a media application.

languageDesired

Indicates the language desired for the SDL interface.

hmiDisplayLanguageDesired

Desired language in HMI.

appId

Identifier of the client application.

callbackToUIThread

If true, all callbacks will occur on the UI thread.

preRegister

Flag that indicates that client should be pre-registered or not

SdlProxyALM(IProxyListenerALM listener, String appName, Boolean isMediaApp, String appId)

```
public SdlProxyALM(IProxyListenerALM listener,String appName,Boolean isMediaApp,String appId)
```

Parameters

listener

Reference to the object in the App listening to callbacks from SDL.

appName

Name of the application displayed on SDL.

isMediaApp

Indicates if the app is a media application.

appId

Identifier of the client application.

SdlProxyALM(Context context, IProxyListenerALM listener, String appName, Boolean isMediaApp, String appID)

Constructor for the SdlProxy object, the proxy for communicating between the App and SDL via specified transport. Takes advantage of the advanced lifecycle management.

```
public SdlProxyALM(Context context,IProxyListenerALM listener,String  
appName,Boolean isMediaApp,String appID)
```

Parameters

context

- Used to create a multiplexing transport config

listener

Reference to the object in the App listening to callbacks from SDL.

appName

Name of the application displayed on SDL.

isMediaApp

Indicates if the app is a media application.

appId

Identifier of the client application.

SdlProxyALM(IProxyListenerALM
listener,
SdlProxyConfigurationResources
sdlProxyConfigurationResources,
String appName, Boolean
isMediaApp, String appId)

```
public SdlProxyALM(IProxyListenerALM listener,  
SdlProxyConfigurationResources sdlProxyConfigurationResources,String  
appName,Boolean isMediaApp,String appId)
```

Parameters

listener

Reference to the object in the App listening to callbacks from SDL.

<code>sdlProxyConfigurationResources</code>
Proxy configuration resources.
<code>appName</code>
Name of the application displayed on SDL.
<code>isMediaApp</code>
Indicates if the app is a media application.
<code>appId</code>
Identifier of the client application.

SdlProxyALM(Context context, IProxyListenerALM listener, SdlProxyConfigurationResources sdlProxyConfigurationResources, String appName, Boolean isMediaApp, String appId)

Constructor for the SdlProxy object, the proxy for communicating between the App and SDL via specified transport. Takes advantage of the advanced lifecycle management.


```
public SdlProxyALM(Context context, IProxyListenerALM listener,  
SdlProxyConfigurationResources sdlProxyConfigurationResources, String  
appName, Boolean isMediaApp, String appId)
```

Parameters

context

- Used to create a multiplexing transport config

listener

Reference to the object in the App listening to callbacks from SDL.

sdlProxyConfigurationResources

Proxy configuration resources.

appName

Name of the application displayed on SDL.

isMediaApp

Indicates if the app is a media application.

appId

Identifier of the client application.

SdlProxyALM(IProxyListenerALM
listener, String appName,
Boolean isMediaApp, String
appID, BaseTransportConfig
transportConfig)

```
public SdlProxyALM(IProxyListenerALM listener,String appName,Boolean  
isMediaApp,String appID,BaseTransportConfig transportConfig)
```

Parameters

listener
appName
isMediaApp
appID

transportConfig

```
SdlProxyALM( IProxyListenerALM  
listener, String appName,  
Boolean isMediaApp, String  
appId, TemplateColorScheme  
dayColorScheme,  
TemplateColorScheme  
nightColorScheme,  
BaseTransportConfig  
transportConfig )
```

```
public SdlProxyALM(IProxyListenerALM listener,String appName,Boolean  
isMediaApp,String appId,TemplateColorScheme dayColorScheme,  
TemplateColorScheme nightColorScheme,BaseTransportConfig  
transportConfig)
```

Parameters

listener

appName
isMediaApp
appID
dayColorScheme
nightColorScheme
transportConfig

SdlProxyALM(IProxyListenerALM
listener, String appName,
Boolean isMediaApp, String
appID, boolean
callbackToUIThread, boolean
preRegister)

```
public SdlProxyALM(IProxyListenerALM listener,String appName,Boolean  
isMediaApp,String appID,boolean callbackToUiThread,boolean  
preRegister)
```

Parameters

listener

Reference to the object in the App listening to callbacks from SDL.

appName

Name of the application displayed on SDL.

isMediaApp

Indicates if the app is a media application.

appID

Identifier of the client application.

callbackToUiThread

If true, all callbacks will occur on the UI thread.

preRegister

Flag that indicates that client should be pre-registered or not

SdlProxyALM(Context context, IProxyListenerALM listener, String appName, Boolean isMediaApp, String appId, boolean callbackToUiThread, boolean preRegister)

Constructor for the SdlProxy object, the proxy for communicating between the App and SDL via specified transport. Takes advantage of the advanced lifecycle management.

```
public SdlProxyALM(Context context,IProxyListenerALM listener,String appName,Boolean isMediaApp,String appId,boolean callbackToUiThread ,boolean preRegister)
```

Parameters

context

- Used to create a multiplexing transport config

listener

Reference to the object in the App listening to callbacks from SDL.

appName

Name of the application displayed on SDL.

isMediaApp

Indicates if the app is a media application.

appID

Identifier of the client application.

callbackToUIThread

If true, all callbacks will occur on the UI thread.

preRegister

Flag that indicates that client should be pre-registered or not

```
SdlProxyALM( Service
appService, IProxyListenerALM
listener,
SdlProxyConfigurationResources
sdlProxyConfigurationResources,
String appName,
java.util.Vector<TTSCChunk>
ttsName, String
ngnMediaScreenAppName,
java.util.Vector<java.lang.String>
vrSynonyms, Boolean
isMediaApp, SdlMsgVersion
sdlMsgVersion, Language
languageDesired, Language
hmiDisplayLanguageDesired,
String appId, String
autoActivateID, boolean
callbackToUIThread, boolean
preRegister )
```

Constructor for the SdlProxy object, the proxy for communicating between the App and SDL via specified transport. Takes advantage of the advanced lifecycle management.


```
public SdlProxyALM(Service appService,IProxyListenerALM listener,  
SdlProxyConfigurationResources sdlProxyConfigurationResources,String  
appName,java.util.Vector<TTSCChunk> ttsName,String  
ngnMediaScreenAppName,java.util.Vector<java.lang.String>  
vrSynonyms,Boolean isMediaApp,SdlMsgVersion sdlMsgVersion,  
Language languageDesired,Language hmiDisplayLanguageDesired,  
String appId,String autoActivateID,boolean callbackToUiThread,boolean  
preRegister)
```

Parameters

appService

Reference to the apps service object.

listener

Reference to the object in the App listening to callbacks from SDL.

sdlProxyConfigurationResources

Proxy configuration resources.

appName

Name of the application displayed on SDL.

ttsName

TTS name.

ngnMediaScreenAppName

Name of the application displayed on SDL for Navigation equipped vehicles. Limited to five characters.

vrSynonyms

A vector of strings, all of which can be used as voice commands too

isMediaApp

Indicates if the app is a media application.

sdIMsgVersion

Indicates the version of SDL SmartDeviceLink Messages desired. Must be less than or equal to the version of SDL SmartDeviceLink running on the vehicle.

languageDesired

Indicates the language desired for the SDL interface.

hmiDisplayLanguageDesired

Desired language in HMI.

appID

Identifier of the client application.

autoActivateID

ID used to re-register previously registered application.

`callbackToUiThread`

If true, all callbacks will occur on the UI thread.

`preRegister`

Flag that indicates that client should be pre-registered or not

```
SdlProxyALM( Service
appService, IProxyListenerALM
listener,
SdlProxyConfigurationResources
sdlProxyConfigurationResources,
String appName,
java.util.Vector<TTSChunk>
ttsName, String
ngnMediaScreenAppName,
java.util.Vector<java.lang.String>
vrSynonyms, Boolean
isMediaApp, SdlMsgVersion
sdlMsgVersion, Language
languageDesired, Language
hmiDisplayLanguageDesired,
String appId, String
autoActivateID, boolean
callbackToUIThread, boolean
preRegister, BaseTransportConfig
transportConfig )
```

```
public SdlProxyALM(Service appService,IProxyListenerALM listener,  
SdlProxyConfigurationResources sdlProxyConfigurationResources,String  
appName,java.util.Vector<TTSCChunk> ttsName,String  
ngnMediaScreenAppName,java.util.Vector<java.lang.String>  
vrSynonyms,Boolean isMediaApp,SdlMsgVersion sdlMsgVersion,  
Language languageDesired,Language hmiDisplayLanguageDesired,  
String appId,String autoActivateId,boolean callbackToUiThread,boolean  
preRegister,BaseTransportConfig transportConfig)
```

Parameters

appService
listener
sdlProxyConfigurationResources
appName
ttsName
ngnMediaScreenAppName

vrSynonyms
isMediaApp
sdIMsgVersion
languageDesired
hmiDisplayLanguageDesired
appID
autoActivateID
callbackToUIThread
preRegister
transportConfig

```
SdlProxyALM( IProxyListenerALM
listener,
SdlProxyConfigurationResources
sdlProxyConfigurationResources,
String appName,
java.util.Vector<TTSCChunk>
ttsName, String
ngnMediaScreenAppName,
java.util.Vector<java.lang.String>
vrSynonyms, Boolean
isMediaApp, SdlMsgVersion
sdlMsgVersion, Language
languageDesired, Language
hmiDisplayLanguageDesired,
String appId, String
autoActivateID, boolean
callbackToUIThread, boolean
preRegister )
```

```
public SdlProxyALM(IProxyListenerALM listener,  
SdlProxyConfigurationResources sdlProxyConfigurationResources,String  
appName,java.util.Vector<TTSCChunk> ttsName,String  
ngnMediaScreenAppName,java.util.Vector<java.lang.String>  
vrSynonyms,Boolean isMediaApp,SdlMsgVersion sdlMsgVersion,  
Language languageDesired,Language hmiDisplayLanguageDesired,  
String appId,String autoActivateID,boolean callbackToUIThread,boolean  
preRegister)
```

Parameters

listener

Reference to the object in the App listening to callbacks from SDL.

sdlProxyConfigurationResources

Proxy configuration resources.

appName

Name of the application displayed on SDL.

ttsName

TTS name.

ngnMediaScreenAppName

Name of the application displayed on SDL for Navigation equipped vehicles. Limited to five characters.

vrSynonyms

A vector of strings, all of which can be used as voice commands too

isMediaApp

Indicates if the app is a media application.

sdIMsgVersion

Indicates the version of SDL SmartDeviceLink Messages desired. Must be less than or equal to the version of SDL SmartDeviceLink running on the vehicle.

languageDesired

Indicates the language desired for the SDL interface.

hmiDisplayLanguageDesired

Desired language in HMI.

appID

Identifier of the client application.

autoActivateID

ID used to re-register previously registered application.

callbackToUIThread

If true, all callbacks will occur on the UI thread.

preRegister

Flag that indicates that client should be pre-registered or not

```
SdlProxyALM( Context context,  
IProxyListenerALM listener,  
SdlProxyConfigurationResources  
sdlProxyConfigurationResources,  
String appName,  
java.util.Vector<TTSChunk>  
ttsName, String  
ngnMediaScreenAppName,  
java.util.Vector<java.lang.String>  
vrSynonyms, Boolean  
isMediaApp, SdlMsgVersion  
sdlMsgVersion, Language  
languageDesired, Language  
hmiDisplayLanguageDesired,  
String appId, String  
autoActivateID, boolean  
callbackToUIThread, boolean  
preRegister )
```

Constructor for the SdlProxy object, the proxy for communicating between the App and SDL via specified transport. Takes advantage of the advanced lifecycle management.

```
public SdlProxyALM(Context context, IProxyListenerALM listener,
SdlProxyConfigurationResources sdlProxyConfigurationResources, String
appName, java.util.Vector<TTSCChunk> ttsName, String
ngnMediaScreenAppName, java.util.Vector<java.lang.String>
vrSynonyms, Boolean isMediaApp, SdlMsgVersion sdlMsgVersion,
Language languageDesired, Language hmiDisplayLanguageDesired,
String appId, String autoActivateID, boolean callbackToUiThread, boolean
preRegister)
```

Parameters

context

- Used to create a multiplexing transport config

listener

Reference to the object in the App listening to callbacks from SDL.

sdlProxyConfigurationResources

Proxy configuration resources.

appName

Name of the application displayed on SDL.

ttsName

TTS name.

ngnMediaScreenAppName

Name of the application displayed on SDL for Navigation equipped vehicles. Limited to five characters.

vrSynonyms

A vector of strings, all of which can be used as voice commands too

isMediaApp

Indicates if the app is a media application.

sdIMsgVersion

Indicates the version of SDL SmartDeviceLink Messages desired. Must be less than or equal to the version of SDL SmartDeviceLink running on the vehicle.

languageDesired

Indicates the language desired for the SDL interface.

hmiDisplayLanguageDesired

Desired language in HMI.

applID

Identifier of the client application.

autoActivateID

ID used to re-register previously registered application.

`callbackToUiThread`

If true, all callbacks will occur on the UI thread.

`preRegister`

Flag that indicates that client should be pre-registered or not

```
SdlProxyALM( IProxyListenerALM
listener,
SdlProxyConfigurationResources
sdlProxyConfigurationResources,
String appName,
java.util.Vector<TTSCChunk>
ttsName, String
ngnMediaScreenAppName,
java.util.Vector<java.lang.String>
vrSynonyms, Boolean
isMediaApp, SdlMsgVersion
sdlMsgVersion, Language
languageDesired, Language
hmiDisplayLanguageDesired,
String appId, String
autoActivateID, boolean
callbackToUIThread, boolean
preRegister, BaseTransportConfig
transportConfig )
```

Constructor for the SdlProxy object, the proxy for communicating between the App and SDL via specified transport. Takes advantage of the advanced lifecycle management.

```
public SdlProxyALM(IProxyListenerALM listener,  
SdlProxyConfigurationResources sdlProxyConfigurationResources,String  
appName,java.util.Vector<TTSCChunk> ttsName,String  
ngnMediaScreenAppName,java.util.Vector<java.lang.String>  
vrSynonyms,Boolean isMediaApp,SdlMsgVersion sdlMsgVersion,  
Language languageDesired,Language hmiDisplayLanguageDesired,  
String appId,String autoActivateID,boolean callbackToUIThread,boolean  
preRegister,BaseTransportConfig transportConfig)
```

Parameters

listener

Reference to the object in the App listening to callbacks from SDL.

sdlProxyConfigurationResources

Proxy configuration resources.

appName

Name of the application displayed on SDL.

ttsName

TTS name.

ngnMediaScreenAppName

Name of the application displayed on SDL for Navigation equipped vehicles. Limited to five characters.

vrSynonyms

A vector of strings, all of which can be used as voice commands too

isMediaApp

Indicates if the app is a media application.

sdIMsgVersion

Indicates the version of SDL SmartDeviceLink Messages desired. Must be less than or equal to the version of SDL SmartDeviceLink running on the vehicle.

languageDesired

Indicates the language desired for the SDL interface.

hmiDisplayLanguageDesired

Desired language in HMI.

appID

Identifier of the client application.

autoActivateID

ID used to re-register previously registered application.

callbackToUIThread

If true, all callbacks will occur on the UI thread.

preRegister

Flag that indicates that client should be pre-registered or not

transportConfig

Initial configuration for transport.

```
SdlProxyALM( IProxyListenerALM
listener,
SdlProxyConfigurationResources
sdlProxyConfigurationResources,
String appName,
java.util.Vector<TTSChunk>
ttsName, String
ngnMediaScreenAppName,
java.util.Vector<java.lang.String>
vrSynonyms, Boolean
isMediaApp, SdlMsgVersion
sdlMsgVersion, Language
languageDesired, Language
hmiDisplayLanguageDesired,
java.util.Vector<AppHMIType>
appType, String appId, String
autoActivateID, boolean
callbackToUIThread, boolean
preRegister )
```

```
public SdlProxyALM(IProxyListenerALM listener,  
SdlProxyConfigurationResources sdlProxyConfigurationResources,String  
appName,java.util.Vector<TTSCheck> ttsName,String  
ngnMediaScreenAppName,java.util.Vector<java.lang.String>  
vrSynonyms,Boolean isMediaApp,SdlMsgVersion sdlMsgVersion,  
Language languageDesired,Language hmiDisplayLanguageDesired,java  
.util.Vector<AppHMType> appType,String appId,String autoActivateID,  
boolean callbackToUIThread,boolean preRegister)
```

Parameters

listener

Reference to the object in the App listening to callbacks from SDL.

sdlProxyConfigurationResources

Proxy configuration resources.

appName

Name of the application displayed on SDL.

ttsName

TTS name.

ngnMediaScreenAppName

Name of the application displayed on SDL for Navigation equipped vehicles. Limited to five characters.

vrSynonyms

A vector of strings, all of which can be used as voice commands too

isMediaApp

Indicates if the app is a media application.

sdIMsgVersion

Indicates the version of SDL SmartDeviceLink Messages desired. Must be less than or equal to the version of SDL SmartDeviceLink running on the vehicle.

languageDesired

Indicates the language desired for the SDL interface.

hmiDisplayLanguageDesired

Desired language in HMI.

appType

Type of application.

appID

Identifier of the client application.

autoActivateID

ID used to re-register previously registered application.

`callbackToUiThread`

If true, all callbacks will occur on the UI thread.

`preRegister`

Flag that indicates that client should be pre-registered or not

```
SdlProxyALM( Context context,  
IProxyListenerALM listener,  
SdlProxyConfigurationResources  
sdlProxyConfigurationResources,  
String appName,  
java.util.Vector<TTSCChunk>  
ttsName, String  
ngnMediaScreenAppName,  
java.util.Vector<java.lang.String>  
vrSynonyms, Boolean  
isMediaApp, SdlMsgVersion  
sdlMsgVersion, Language  
languageDesired, Language  
hmiDisplayLanguageDesired,  
java.util.Vector<AppHMType>  
appType, String appId, String  
autoActivateID, boolean  
callbackToUIThread, boolean  
preRegister )
```

Constructor for the SdlProxy object, the proxy for communicating between the App and SDL via specified transport. Takes advantage of the advanced lifecycle management.

```
public SdlProxyALM(Context context, IProxyListenerALM listener,  
SdlProxyConfigurationResources sdlProxyConfigurationResources, String  
appName, java.util.Vector<TTSCChunk> ttsName, String  
ngnMediaScreenAppName, java.util.Vector<java.lang.String>  
vrSynonyms, Boolean isMediaApp, SdlMsgVersion sdlMsgVersion,  
Language languageDesired, Language hmiDisplayLanguageDesired, java  
.util.Vector<AppHMIMType> appType, String appId, String autoActivateID,  
boolean callbackToUiThread, boolean preRegister)
```

Parameters

context

- Used to create a multiplexing transport config

listener

Reference to the object in the App listening to callbacks from SDL.

sdlProxyConfigurationResources

Proxy configuration resources.

appName

Name of the application displayed on SDL.

ttsName

TTS name.

ngnMediaScreenAppName

Name of the application displayed on SDL for Navigation equipped vehicles. Limited to five characters.

vrSynonyms

A vector of strings, all of which can be used as voice commands too

isMediaApp

Indicates if the app is a media application.

sdIMsgVersion

Indicates the version of SDL SmartDeviceLink Messages desired. Must be less than or equal to the version of SDL SmartDeviceLink running on the vehicle.

languageDesired

Indicates the language desired for the SDL interface.

hmiDisplayLanguageDesired

Desired language in HMI.

appType

Type of application.

appID

Identifier of the client application.

autoActivateID

ID used to re-register previously registered application.

callbackToUIThread

If true, all callbacks will occur on the UI thread.

preRegister

Flag that indicates that client should be pre-registered or not

```
SdlProxyALM( IProxyListenerALM
listener,
SdlProxyConfigurationResources
sdlProxyConfigurationResources,
String appName,
java.util.Vector<TTSChunk>
ttsName, String
ngnMediaScreenAppName,
java.util.Vector<java.lang.String>
vrSynonyms, Boolean
isMediaApp, SdlMsgVersion
sdlMsgVersion, Language
languageDesired, Language
hmiDisplayLanguageDesired,
java.util.Vector<AppHMIType>
appType, String appId, String
autoActivateID, boolean
callbackToUIThread, boolean
preRegister, BaseTransportConfig
transportConfig )
```

Constructor for the SdlProxy object, the proxy for communicating between the App and SDL via specified transport. Takes advantage of the advanced lifecycle management.

```
public SdlProxyALM(IProxyListenerALM listener,  
SdlProxyConfigurationResources sdlProxyConfigurationResources,String  
appName,java.util.Vector<TTSCChunk> ttsName,String  
ngnMediaScreenAppName,java.util.Vector<java.lang.String>  
vrSynonyms,Boolean isMediaApp,SdlMsgVersion sdlMsgVersion,  
Language languageDesired,Language hmiDisplayLanguageDesired,java  
.util.Vector<AppHMType> appType,String appId,String autoActivateID,  
boolean callbackToUiThread,boolean preRegister,BaseTransportConfig  
transportConfig)
```

Parameters

listener

Reference to the object in the App listening to callbacks from SDL.

sdlProxyConfigurationResources

Proxy configuration resources.

appName

Name of the application displayed on SDL.

ttsName

TTS name.

ngnMediaScreenAppName

Name of the application displayed on SDL for Navigation equipped vehicles. Limited to five characters.

vrSynonyms

A vector of strings, all of which can be used as voice commands too

isMediaApp

Indicates if the app is a media application.

sdIMsgVersion

Indicates the version of SDL SmartDeviceLink Messages desired. Must be less than or equal to the version of SDL SmartDeviceLink running on the vehicle.

languageDesired

Indicates the language desired for the SDL interface.

hmiDisplayLanguageDesired

Desired language in HMI.

appType

Type of application.

appID

Identifier of the client application.

autoActivateID

ID used to re-register previously registered application.

callbackToUIThread

If true, all callbacks will occur on the UI thread.

preRegister

Flag that indicates that client should be pre-registered or not

transportConfig

Initial configuration for transport.

```
SdlProxyALM( Service
appService, IProxyListenerALM
listener,
SdlProxyConfigurationResources
sdlProxyConfigurationResources,
String appName,
java.util.Vector<TTSCChunk>
ttsName, String
ngnMediaScreenAppName,
java.util.Vector<java.lang.String>
vrSynonyms, Boolean
isMediaApp, SdlMsgVersion
sdlMsgVersion, Language
languageDesired, Language
hmiDisplayLanguageDesired,
java.util.Vector<AppHMIType>
appType, String appId, String
autoActivateID, boolean
callbackToUIThread, boolean
preRegister, BaseTransportConfig
transportConfig )
```

```
public SdlProxyALM(Service appService,IProxyListenerALM listener,  
SdlProxyConfigurationResources sdlProxyConfigurationResources,String  
appName,java.util.Vector<TTSCChunk> ttsName,String  
ngnMediaScreenAppName,java.util.Vector<java.lang.String>  
vrSynonyms,Boolean isMediaApp,SdlMsgVersion sdlMsgVersion,  
Language languageDesired,Language hmiDisplayLanguageDesired,java  
.util.Vector<AppHMIType> appType,String appId,String autoActivateID,  
boolean callbackToUIThread,boolean preRegister,BaseTransportConfig  
transportConfig)
```

Parameters

appService

listener

sdlProxyConfigurationResources

appName

ttsName

ngnMediaScreenAppName

vrSynonyms
isMediaApp
sdIMsgVersion
languageDesired
hmiDisplayLanguageDesired
appType
appID
autoActivateID
callbackToUIThread
preRegister

transportConfig

```
SdlProxyALM( IProxyListenerALM
listener,
SdlProxyConfigurationResources
sdlProxyConfigurationResources,
String appName,
java.util.Vector<TTSChunk>
ttsName, String
ngnMediaScreenAppName,
java.util.Vector<java.lang.String>
vrSynonyms, Boolean
isMediaApp, SdlMsgVersion
sdlMsgVersion, Language
languageDesired, Language
hmiDisplayLanguageDesired,
java.util.Vector<AppHMIType>
appType, String appId, String
autoActivateID, boolean
callbackToUIThread, boolean
preRegister, String sHashID,
BaseTransportConfig
transportConfig )
```

Constructor for the SdlProxy object, the proxy for communicating between the App and SDL via specified transport. Takes advantage of the advanced lifecycle management.

```
public SdlProxyALM(IProxyListenerALM listener,  
SdlProxyConfigurationResources sdlProxyConfigurationResources,String  
appName,java.util.Vector<TTSCChunk> ttsName,String  
ngnMediaScreenAppName,java.util.Vector<java.lang.String>  
vrSynonyms,Boolean isMediaApp,SdlMsgVersion sdlMsgVersion,  
Language languageDesired,Language hmiDisplayLanguageDesired,java  
.util.Vector<AppHMType> appType,String appId,String autoActivateID,  
boolean callbackToUiThread,boolean preRegister,String sHashID,  
BaseTransportConfig transportConfig)
```

Parameters

listener

Reference to the object in the App listening to callbacks from SDL.

sdlProxyConfigurationResources

Proxy configuration resources.

appName

Name of the application displayed on SDL.

ttsName

TTS name.

ngnMediaScreenAppName

Name of the application displayed on SDL for Navigation equipped vehicles. Limited to five characters.

vrSynonyms

A vector of strings, all of which can be used as voice commands too

isMediaApp

Indicates if the app is a media application.

sdIMsgVersion

Indicates the version of SDL SmartDeviceLink Messages desired. Must be less than or equal to the version of SDL SmartDeviceLink running on the vehicle.

languageDesired

Indicates the language desired for the SDL interface.

hmiDisplayLanguageDesired

Desired language in HMI.

appType

Type of application.

appID

Identifier of the client application.

autoActivateID

ID used to re-register previously registered application.

callbackToUIThread

If true, all callbacks will occur on the UI thread.

preRegister

Flag that indicates that client should be pre-registered or not

sHashID

HashID used for app resumption

transportConfig

Initial configuration for transport.

SdlProxyALM(Service
appService, IProxyListenerALM
listener,
SdlProxyConfigurationResources
sdlProxyConfigurationResources,
String appName,
java.util.Vector<TTSCChunk>
ttsName, String
ngnMediaScreenAppName,
java.util.Vector<java.lang.String>
vrSynonyms, Boolean
isMediaApp, SdlMsgVersion
sdlMsgVersion, Language
languageDesired, Language
hmiDisplayLanguageDesired,
java.util.Vector<AppHMType>
appType, String appId, String
autoActivateID, boolean
callbackToUIThread, boolean
preRegister, String sHashID,

BaseTransportConfig transportConfig)

```
public SdlProxyALM(Service appService,IProxyListenerALM listener,  
SdlProxyConfigurationResources sdlProxyConfigurationResources,String  
appName,java.util.Vector<TTSCChunk> ttsName,String  
ngnMediaScreenAppName,java.util.Vector<java.lang.String>  
vrSynonyms,Boolean isMediaApp,SdlMsgVersion sdlMsgVersion,  
Language languageDesired,Language hmiDisplayLanguageDesired,java  
.util.Vector<AppHMType> appType,String appId,String autoActivateID,  
boolean callbackToUIThread,boolean preRegister,String sHashID,  
BaseTransportConfig transportConfig)
```

Parameters

appService
listener
sdlProxyConfigurationResources
appName
ttsName

ngnMediaScreenAppName
vrSynonyms
isMediaApp
sdIMsgVersion
languageDesired
hmiDisplayLanguageDesired
appType
appID
autoActivateID
callbackToUIThread

preRegister
sHashID
transportConfig

SdlProxyALM(Service
appService, IProxyListenerALM
listener,
SdlProxyConfigurationResources
sdlProxyConfigurationResources,
String appName,
java.util.Vector<TTSCChunk>
ttsName, String
ngnMediaScreenAppName,
java.util.Vector<java.lang.String>
vrSynonyms, Boolean
isMediaApp, SdlMsgVersion
sdlMsgVersion, Language
languageDesired, Language
hmiDisplayLanguageDesired,
java.util.Vector<AppHMIType>
appType, String appId, String
autoActivateID,
TemplateColorScheme
dayColorScheme,
TemplateColorScheme

nightColorScheme, boolean
callbackToUiThread, boolean
preRegister, String sHashID,
BaseTransportConfig
transportConfig)

```
public SdlProxyALM(Service appService,IProxyListenerALM listener,  
SdlProxyConfigurationResources sdlProxyConfigurationResources,String  
appName,java.util.Vector<TTSCChunk> ttsName,String  
ngnMediaScreenAppName,java.util.Vector<java.lang.String>  
vrSynonyms,Boolean isMediaApp,SdlMsgVersion sdlMsgVersion,  
Language languageDesired,Language hmiDisplayLanguageDesired,java  
.util.Vector<AppHMIMType> appType,String appId,String autoActivateID,  
TemplateColorScheme dayColorScheme,TemplateColorScheme  
nightColorScheme,boolean callbackToUiThread,boolean preRegister,  
String sHashID,BaseTransportConfig transportConfig)
```

Parameters

appService

listener

sdlProxyConfigurationResources

appName

ttsName

ngnMediaScreenAppName

vrSynonyms

isMediaApp

sdIMsgVersion

languageDesired

hmiDisplayLanguageDesired

appType

appID

autoActivateID
dayColorScheme
nightColorScheme
callbackToUIThread
preRegister
sHashID
transportConfig

Methods

resetProxy()

Disconnects the application from SDL, then recreates the transport such that the next time a SDL unit discovers applications, this application will be available.

```
public void resetProxy() throws SdlException
```

Throws

- SdlException

getButtonCapabilities()

Deprecated

Gets buttonCapabilities set when application interface is registered.

```
public java.util.List getButtonCapabilities() throws SdlException
```

Throws

- `SdlException`

getSoftButtonCapabilities()

Deprecated

Gets getSoftButtonCapabilities set when application interface is registered.

```
public java.util.List getSoftButtonCapabilities() throws SdlException
```

Throws

- `SdlException`

getPresetBankCapabilities()

Deprecated

Gets getPresetBankCapabilities set when application interface is registered.

```
public PresetBankCapabilities getPresetBankCapabilities() throws  
SdlException
```


Throws

- `SdlException`

getProxyVersionInfo()

Gets the current version information of the proxy.

```
public java.lang.String getProxyVersionInfo() throws SdlException
```

Throws

- `SdlException`

getDisplayCapabilities()

Deprecated

Gets displayCapabilities set when application interface is registered.

```
public DisplayCapabilities getDisplayCapabilities() throws SdlException
```

Throws

- `SdlException`

getHmiZoneCapabilities()

Deprecated

Gets hmiZoneCapabilities set when application interface is registered.

```
public java.util.List getHmiZoneCapabilities() throws SdlException
```

Throws

- `SdlException`

getSpeechCapabilities()

Deprecated

Gets speechCapabilities set when application interface is registered.

```
public java.util.List getSpeechCapabilities() throws SdlException
```

Throws

- `SdlException`

`getPrerecordedSpeech()`

Gets `PrerecordedSpeech` set when application interface is registered.

```
public java.util.List getPrerecordedSpeech() throws SdlException
```

Throws

- `SdlException`

`getSdlLanguage()`

Gets `sdlLanguage` set when application interface is registered.

```
public Language getSdlLanguage() throws SdlException
```

Throws

- `SdlException`

`getHmiDisplayLanguage()`

Gets `getHmiDisplayLanguage` set when application interface is registered.

```
public Language getHmiDisplayLanguage() throws SdlException
```

Throws

- `SdlException`

`getSdlMsgVersion()`

Gets `sdlMsgVersion` set when application interface is registered.

```
public SdlMsgVersion getSdlMsgVersion() throws SdlException
```

Throws

- `SdlException`

`getVrCapabilities()`

Deprecated

Gets vrCapabilities set when application interface is registered.

```
public java.util.List getVrCapabilities() throws SdlException
```

Throws

- `SdlException`

`getVehicleType()`

Gets getVehicleType set when application interface is registered.

```
public VehicleType getVehicleType() throws SdlException
```

Throws

- `SdlException`

getAudioPassThruCapabilities()

Deprecated

Gets AudioPassThruCapabilities set when application interface is registered.

```
public java.util.List getAudioPassThruCapabilities() throws SdlException
```

Throws

- `SdlException`

getSupportedDiagModes()

```
public java.util.List getSupportedDiagModes() throws SdlException
```

Throws

- `SdlException`

`getHmiCapabilities()`

Deprecated

Gets HmiCapabilities when application interface is registered.

```
public HmiCapabilities getHmiCapabilities() throws SdlException
```

Throws

- `SdlException`

`getSystemSoftwareVersion()`

```
public java.lang.String getSystemSoftwareVersion() throws SdlException
```

Throws

- `SdlException`

`isAppResumeSuccess()`

```
public boolean isAppResumeSuccess() throws SdlException
```


Throws

- `SdlException`

Inherited Methods

From Class

Methods

```
com.smartdevicelink.proxy.SdlProxyBase
```

addPutFileResponseListener, remPutFileResponseListener, getIsConnected, getAppInterfaceRegistered, initializeProxy, forceOnConnected, sendTransportBroadcast, enableSiphonDebug, disableSiphonDebug, enableDebugTool, disableDebugTool, isDebugEnabled, isAudioStreamTransportAvailable, isVideoStreamTransportAvailable, close, dispose, cycleProxy, getProtocolVersion, serializeJSON, onPacketProgress, addOnRPCResponseListener, getResponseListeners, onRPCNotificationReceived, onRPCReceived, onRPCRequestReceived, addOnRPCNotificationListener, addOnRPCListener, removeOnRPCListener, addOnRPCRequestListener, removeOnRPCNotificationListener, removeOnRPCNotificationListener, removeOnRPCRequestListener, getSdlMsgVersion, sendSequentialRequests, sendRequests, sendRPC, sendRPCRequest, notifyProxyClosed, startRPCStream, startRPCStream, endRPCStream, createFutureTask, createScheduler, startService, endService, startH264, startH264, endH264, pausePCM, pauseH264, resumePCM, resumeH264, startPCM, startPCM, endPCM, startVideoStream, endVideoStream, pauseVideoStream, resumeVideoStream, createOpenGLInputSurface, startRemoteDisplayStream, stopRemoteDisplayStream, startEncoder, releaseEncoder, drainEncoder, startAudioStream, endAudioStream, pauseAudioStream, resumeAudioStream, setAppService, startProtectedRPCService, getLockScreenIcon, addCommand, addCommand, addCommand, addCommand, addCommand, addCommand, addCommand, addCommand, addCommand, addCommand, addCommand, addCommand, addCommand, addCommand, addSubMenu, addSubMenu, addSubMenu, alert, alert, alert, alert, alert, alert, alert, alert, alert, alert, alert, alert, deleteInteractionChoiceSet, deleteCommand, deleteInteractionChoiceSet, deleteSubMenu, performInteraction, performInteraction, performInteraction, performInteraction, performInteraction, performInteraction

java.lang.Object

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

Class SdlProxyBase

Hierarchy

java.lang.Object

Package

com.smartdevicelink.proxy

All Known Subclasses

com.smartdevicelink.proxy.SdlProxyALM

Quick Jump

Fields

- TAG
- MAX_SUPPORTED_RPC_VERSION
- _appService
- _advancedLifecycleManagementEnabled
- _appInterfaceRegisterd
- _preRegisterd
- _haveReceivedFirstFocusLevel
- _haveReceivedFirstFocusLevelFull
- _proxyDisposed
- _sdlConnectionState
- _sdlIntefaceAvailablity

- `_hmiLevel`
- `lastHmiStatus`
- `_audioStreamingState`
- `_sdIMsgVersion`
- `_autoActivateIdReturned`
- `_sdlLanguage`
- `_hmiDisplayLanguage`
- `_prerecordedSpeech`
- `_vehicleType`
- `_systemSoftwareVersion`
- `_diagModes`
- `firstTimeFull`
- `_proxyVersionInfo`
- `_bResumeSuccess`
- `_secList`
- `_systemCapabilityManager`
- `_iconResumed`
- `raiResponse`
- `protocolVersion`
- `rpcSpecVersion`
- `rpcResponseListeners`
- `rpcNotificationListeners`
- `rpcRequestListeners`
- `rpcListeners`
- `manager`
- `authToken`

Constructors

- `SdlProxyBase(proxyListenerType listener, Context context, String appName, String shortAppName, Boolean isMediaApp, Language languageDesired, Language hmiDisplayLanguageDesired, java.util.Vector<AppHMIType> appType, String appId, BaseTransportConfig transportConfig, java.util.Vector<java.lang.String> vrSynonyms, java.util.Vector<TTSCChunk> ttsName, TemplateColorScheme dayColorScheme, TemplateColorScheme nightColorScheme)`
- `SdlProxyBase(proxyListenerType listener, SdlProxyConfigurationResources sdlProxyConfigurationResources, boolean enableAdvancedLifecycleManagement, String appName, java.util.Vector<TTSCChunk> ttsName, String ngnMediaScreenAppName, java.util.Vector<java.lang.String> vrSynonyms, Boolean isMediaApp, SdlMsgVersion sdlMsgVersion, Language languageDesired, Language hmiDisplayLanguageDesired, java.util.Vector<AppHMIType> appType, String appId, String autoActivateId, boolean callbackToUiThread, BaseTransportConfig transportConfig)`
- `SdlProxyBase(proxyListenerType listener, SdlProxyConfigurationResources sdlProxyConfigurationResources, boolean enableAdvancedLifecycleManagement, String appName, java.util.Vector<TTSCChunk> ttsName, String ngnMediaScreenAppName, java.util.Vector<java.lang.String> vrSynonyms, Boolean isMediaApp, SdlMsgVersion sdlMsgVersion,`

- Language languageDesired, Language hmiDisplayLanguageDesired, java.util.Vector<AppHMIType> appType, String applID, String autoActivateID, TemplateColorScheme dayColorScheme, TemplateColorScheme nightColorScheme, boolean callbackToUiThread, boolean preRegister, String sHashID, Boolean bEnableResume, BaseTransportConfig transportConfig)
- SdlProxyBase(proxyListenerType listener, SdlProxyConfigurationResources sdlProxyConfigurationResources, boolean enableAdvancedLifecycleManagement, String appName, java.util.Vector<TTSCChunk> ttsName, String ngnMediaScreenAppName, java.util.Vector<java.lang.String> vrSynonyms, Boolean isMediaApp, SdlMsgVersion sdlMsgVersion, Language languageDesired, Language hmiDisplayLanguageDesired, java.util.Vector<AppHMIType> appType, String applID, String autoActivateID, boolean callbackToUiThread, boolean preRegister, BaseTransportConfig transportConfig)
 - SdlProxyBase(proxyListenerType listener, SdlProxyConfigurationResources sdlProxyConfigurationResources, boolean enableAdvancedLifecycleManagement, String appName, java.util.Vector<TTSCChunk> ttsName, String ngnMediaScreenAppName, java.util.Vector<java.lang.String> vrSynonyms, Boolean isMediaApp, SdlMsgVersion sdlMsgVersion, Language languageDesired, Language hmiDisplayLanguageDesired, java.util.Vector<AppHMIType> appType, String applID, String autoActivateID, TemplateColorScheme dayColorScheme, TemplateColorScheme nightColorScheme, boolean callbackToUiThread, boolean preRegister, BaseTransportConfig transportConfig)

Methods

- addPutFileResponseListener(IPutFileResponseListener _putFileListener)
- remPutFileResponseListener(IPutFileResponseListener _putFileListener)
- getIsConnected()
- getAppInterfaceRegistered()
- initializeProxy()
- forceOnConnected()
- sendTransportBroadcast()
- enableSiphonDebug()
- disableSiphonDebug()
- enableDebugTool()
- disableDebugTool()
- isDebugEnabled()
- isAudioStreamTransportAvailable()
- isVideoStreamTransportAvailable()
- close() **Deprecated**
- dispose()
- cycleProxy(SdlDisconnectedReason disconnectedReason)
- getProtocolVersion()

- `serializeJSON(RPCMessage msg)`
- `onPacketProgress(int correlationId,long bytesWritten,long totalSize)`
- `addOnRPCResponseListener(OnRPCResponseListener listener,int correlationId,int totalSize)`
- `getResponseListeners()`
- `onRPCNotificationReceived(RPCNotification notification)`
- `onRPCReceived(RPCMessage message)`
- `onRPCRequestReceived(RPCRequest request)`
- `addOnRPCNotificationListener(FunctionID notificationId,OnRPCNotificationListener listener)`
- `addOnRPCListener(FunctionID messageId,OnRPCListener listener)`
- `removeOnRPCListener(FunctionID messageId,OnRPCListener listener)`
- `addOnRPCRequestListener(FunctionID requestId,OnRPCRequestListener listener)`
- `removeOnRPCNotificationListener(FunctionID notificationId)`
- **Deprecated**
- `removeOnRPCNotificationListener(FunctionID notificationId,OnRPCNotificationListener listener)`
- `removeOnRPCRequestListener(FunctionID requestId,OnRPCRequestListener listener)`
- `getSdlMsgVersion()`
- `sendSequentialRequests(java.util.List<? extends RPCMessage> rpcs,OnMultipleRequestListener listener)`
- `sendRequests(java.util.List<? extends RPCMessage> rpcs,OnMultipleRequestListener listener)`
- `sendRPC(RPCMessage message)`
- `sendRPCRequest(RPCRequest request)` **Deprecated**
- `notifyProxyClosed(String info,Exception e,SdlDisconnectedReason reason)`
- `startRPCStream(InputStream is,RPCRequest msg)`
- `startRPCStream(RPCRequest msg)`
- `endRPCStream()`
- `createFutureTask`
(`com.smartdevicelink.proxy.SdlProxyBase<proxyListenerType>.CallableMethod`
`callMethod`)
- `createScheduler()`
- `startService(SessionType serviceType,boolean isEncrypted)`
- `endService(SessionType serviceType)`
- `startH264(InputStream is,boolean isEncrypted)` **Deprecated**
- `startH264(boolean isEncrypted)` **Deprecated**
- `endH264()` **Deprecated**
- `pausePCM()` **Deprecated**
- `pauseH264()` **Deprecated**
- `resumePCM()` **Deprecated**
- `resumeH264()` **Deprecated**
- `startPCM(InputStream is,boolean isEncrypted)` **Deprecated**
- `startPCM(boolean isEncrypted)` **Deprecated**
- `endPCM()` **Deprecated**
- `startVideoStream(boolean isEncrypted,VideoStreamingParameters parameters)`
- `endVideoStream()`
- `pauseVideoStream()`
- `resumeVideoStream()`

- `createOpenGLInputSurface(int frameRate,int iFrameInterval,int width,int height,int bitrate,boolean isEncrypted)`
- `startRemoteDisplayStream(Context context,java.lang.Class<? extends com.smartdevicelink.streaming.video.SdlRemoteDisplay> remoteDisplay,VideoStreamingParameters parameters,boolean encrypted)`
- `stopRemoteDisplayStream()`
- `startEncoder()`
- `releaseEncoder()`
- `drainEncoder(boolean endOfStream)`
- `startAudioStream(boolean isEncrypted,AudioStreamingCodec codec,AudioStreamingParams params)`
- `endAudioStream()`
- `pauseAudioStream()`
- `resumeAudioStream()`
- `setAppService(Service mService)`
- `startProtectedRPCService()`
- `getLockScreenIcon`
(`LockScreenManager.OnLockScreenIconDownloadedListener l`)
- `addCommand(Integer commandID,String menuText,Integer parentID,Integer position,java.util.Vector<java.lang.String> vrCommands,String IconValue,ImageType IconType,Integer correlationID)`
- `addCommand(Integer commandID,String menuText,Integer position,java.util.Vector<java.lang.String> vrCommands,String IconValue,ImageType IconType,Integer correlationID)`
- `addCommand(Integer commandID,String menuText,Integer position,String IconValue,ImageType IconType,Integer correlationID)`
- `addCommand(Integer commandID,String menuText,String IconValue,ImageType IconType,Integer correlationID)`
- `addCommand(Integer commandID,String menuText,java.util.Vector<java.lang.String> vrCommands,String IconValue,ImageType IconType,Integer correlationID)`
- `addCommand(Integer commandID,java.util.Vector<java.lang.String> vrCommands,String IconValue,ImageType IconType,Integer correlationID)`
- `addCommand(Integer commandID,String menuText,Integer parentID,Integer position,java.util.Vector<java.lang.String> vrCommands,Integer correlationID)`
- `addCommand(Integer commandID,String menuText,Integer position,java.util.Vector<java.lang.String> vrCommands,Integer correlationID)`
- `addCommand(Integer commandID,String menuText,Integer position,Integer correlationID)`
- `addCommand(Integer commandID,String menuText,Integer correlationID)`
- `addCommand(Integer commandID,String menuText,java.util.Vector<java.lang.String> vrCommands,Integer correlationID)`
- `addCommand(Integer commandID,java.util.Vector<java.lang.String> vrCommands,Integer correlationID)`

- addSubMenu(Integer menuID,String menuName,Integer position,Image menuIcon,Integer correlationID)
- addSubMenu(Integer menuID,String menuName,Integer position,Integer correlationID) **Deprecated**
- addSubMenu(Integer menuID,String menuName,Integer correlationID) **Deprecated**
- alert(String ttsText,String alertText1,String alertText2,String alertText3,Boolean playTone,Integer duration,java.util.Vector<SoftButton> softButtons,Integer correlationID)
- alert(java.util.Vector<TTSChunk> ttsChunks,String alertText1,String alertText2,String alertText3,Boolean playTone,Integer duration,java.util.Vector<SoftButton> softButtons,Integer correlationID)
- alert(String ttsText,Boolean playTone,java.util.Vector<SoftButton> softButtons,Integer correlationID)
- alert(java.util.Vector<TTSChunk> chunks,Boolean playTone,java.util.Vector<SoftButton> softButtons,Integer correlationID)
- alert(String alertText1,String alertText2,String alertText3,Boolean playTone,Integer duration,java.util.Vector<SoftButton> softButtons,Integer correlationID)
- alert(String ttsText,String alertText1,String alertText2,Boolean playTone,Integer duration,Integer correlationID)
- alert(java.util.Vector<TTSChunk> ttsChunks,String alertText1,String alertText2,Boolean playTone,Integer duration,Integer correlationID)
- alert(String ttsText,Boolean playTone,Integer correlationID)
- alert(java.util.Vector<TTSChunk> chunks,Boolean playTone,Integer correlationID)
- alert(String alertText1,String alertText2,Boolean playTone,Integer duration,Integer correlationID)
- createInteractionChoiceSet(java.util.Vector<Choice> choiceSet,Integer interactionChoiceSetID,Integer correlationID)
- deleteCommand(Integer commandID,Integer correlationID)
- deleteInteractionChoiceSet(Integer interactionChoiceSetID,Integer correlationID)
- deleteSubMenu(Integer menuID,Integer correlationID)
- performInteraction(String initPrompt,String displayText,Integer interactionChoiceSetID,java.util.Vector<VrHelpItem> vrHelp,Integer correlationID)
- performInteraction(String initPrompt,String displayText,Integer interactionChoiceSetID,String helpPrompt,String timeoutPrompt,InteractionMode interactionMode,Integer timeout,java.util.Vector<VrHelpItem> vrHelp,Integer correlationID)
- performInteraction(String initPrompt,String displayText,java.util.Vector<java.lang.Integer> interactionChoiceSetIDList,String helpPrompt,String timeoutPrompt,InteractionMode interactionMode,Integer timeout,java.util.Vector<VrHelpItem> vrHelp,Integer correlationID)
- performInteraction(java.util.Vector<TTSChunk> initChunks,String displayText,java.util.Vector<java.lang.Integer> interactionChoiceSetIDList,java.util.Vector<TTSChunk>

- helpChunks,java.util.Vector<TTSCChunk>
timeoutChunks,InteractionMode interactionMode,Integer
timeout,java.util.Vector<VrHelpItem> vrHelp,Integer
correlationID)
- performInteraction(String initPrompt,String displayText,Integer
interactionChoiceSetID,Integer correlationID)
- performInteraction(String initPrompt,String displayText,Integer
interactionChoiceSetID,String helpPrompt,String
timeoutPrompt,InteractionMode interactionMode,Integer
timeout,Integer correlationID)
- performInteraction(String initPrompt,String
displayText,java.util.Vector<java.lang.Integer>
interactionChoiceSetIDList,String helpPrompt,String
timeoutPrompt,InteractionMode interactionMode,Integer
timeout,Integer correlationID)
- performInteraction(java.util.Vector<TTSCChunk> initChunks,String
displayText,java.util.Vector<java.lang.Integer>
interactionChoiceSetIDList,java.util.Vector<TTSCChunk>
helpChunks,java.util.Vector<TTSCChunk>
timeoutChunks,InteractionMode interactionMode,Integer
timeout,Integer correlationID)
- registerAppInterfacePrivate(SdlMsgVersion sdlMsgVersion,String
appName,java.util.Vector<TTSCChunk> ttsName,String
ngnMediaScreenAppName,java.util.Vector<java.lang.String>
vrSynonyms,Boolean isMediaApp,Language
languageDesired,Language
hmiDisplayLanguageDesired,java.util.Vector<AppHMIType>
appType,String appId,TemplateColorScheme
dayColorScheme,TemplateColorScheme nightColorScheme,Integer
correlationID)
- setGlobalProperties(String helpPrompt,String
timeoutPrompt,String vrHelpTitle,java.util.Vector<VrHelpItem>
vrHelp,Integer correlationID)
- setGlobalProperties(java.util.Vector<TTSCChunk>
helpChunks,java.util.Vector<TTSCChunk> timeoutChunks,String
vrHelpTitle,java.util.Vector<VrHelpItem> vrHelp,Integer
correlationID)
- setGlobalProperties(String helpPrompt,String
timeoutPrompt,Integer correlationID)
- setGlobalProperties(java.util.Vector<TTSCChunk>
helpChunks,java.util.Vector<TTSCChunk> timeoutChunks,Integer
correlationID)
- resetGlobalProperties(java.util.Vector<GlobalProperty>
properties,Integer correlationID)
- setMediaClockTimer(Integer hours,Integer minutes,Integer
seconds,UpdateMode updateMode,Integer correlationID)
- pauseMediaClockTimer(Integer correlationID)
- resumeMediaClockTimer(Integer correlationID)
- clearMediaClockTimer(Integer correlationID)
- show(String mainText1,String mainText2,String mainText3,String
mainText4,String statusBar,String mediaClock,String
mediaTrack,Image graphic,java.util.Vector<SoftButton>
softButtons,java.util.Vector<java.lang.String>
customPresets,TextAlignment alignment,Integer correlationID)

- `show(String mainText1,String mainText2,String mainText3,String mainText4,Image graphic,java.util.Vector<SoftButton> softButtons,java.util.Vector<java.lang.String> customPresets,TextAlignment alignment,Integer correlationID)`
- `show(String mainText1,String mainText2,String statusBar,String mediaClock,String mediaTrack,TextAlignment alignment,Integer correlationID)`
- `show(String mainText1,String mainText2,TextAlignment alignment,Integer correlationID)`
- `speak(String ttsText,Integer correlationID)`
- `speak(java.util.Vector<TTSCChunk> ttsChunks,Integer correlationID)`
- `subscribeButton(ButtonName buttonName,Integer correlationID)`
- `unregisterAppInterfacePrivate(Integer correlationID)`
- `unsubscribeButton(ButtonName buttonName,Integer correlationID)`
- `createChoiceSetChoice(Integer choiceID,String choiceMenuName,java.util.Vector<java.lang.String> choiceVrCommands)`
- `performaudiopassthru(String initialPrompt,String audioPassThruDisplayText1,String audioPassThruDisplayText2,SamplingRate samplingRate,Integer maxDuration,BitsPerSample bitsPerSample,AudioType audioType,Boolean muteAudio,Integer correlationID)`
- `endaudiopassthru(Integer correlationID)`
- `subscribevehicledata(boolean gps,boolean speed,boolean rpm,boolean fuelLevel,boolean fuelLevel_State,boolean instantFuelConsumption,boolean externalTemperature,boolean prndl,boolean tirePressure,boolean odometer,boolean beltStatus,boolean bodyInformation,boolean deviceStatus,boolean driverBraking,Integer correlationID)` **Deprecated**
- `subscribevehicledata(boolean gps,boolean speed,boolean rpm,boolean fuelLevel,boolean fuelLevel_State,boolean instantFuelConsumption,boolean externalTemperature,boolean prndl,boolean tirePressure,boolean engineOilLife,boolean odometer,boolean beltStatus,boolean bodyInformation,boolean deviceStatus,boolean driverBraking,Integer correlationID)`
- `unsubscribevehicledata(boolean gps,boolean speed,boolean rpm,boolean fuelLevel,boolean fuelLevel_State,boolean instantFuelConsumption,boolean externalTemperature,boolean prndl,boolean tirePressure,boolean odometer,boolean beltStatus,boolean bodyInformation,boolean deviceStatus,boolean driverBraking,Integer correlationID)` **Deprecated**
- `unsubscribevehicledata(boolean gps,boolean speed,boolean rpm,boolean fuelLevel,boolean fuelLevel_State,boolean instantFuelConsumption,boolean externalTemperature,boolean prndl,boolean tirePressure,boolean engineOilLife,boolean odometer,boolean beltStatus,boolean bodyInformation,boolean deviceStatus,boolean driverBraking,Integer correlationID)`
- `getvehicledata(boolean gps,boolean speed,boolean rpm,boolean fuelLevel,boolean fuelLevel_State,boolean instantFuelConsumption,boolean externalTemperature,boolean vin,boolean prndl,boolean tirePressure,boolean odometer,boolean beltStatus,boolean bodyInformation,boolean deviceStatus,boolean driverBraking,Integer correlationID)` **Deprecated**

- `getvehicledata(boolean gps,boolean speed,boolean rpm,boolean fuelLevel,boolean fuelLevel_State,boolean instantFuelConsumption,boolean externalTemperature,boolean vin,boolean prndl,boolean tirePressure,boolean engineOilLife,boolean odometer,boolean beltStatus,boolean bodyInformation,boolean deviceStatus,boolean driverBraking,Integer correlationID)`
- `scrollablemessage(String scrollableMessageBody,Integer timeout,java.util.Vector<SoftButton> softButtons,Integer correlationID)`
- `slider(Integer numTicks,Integer position,String sliderHeader,java.util.Vector<java.lang.String> sliderFooter,Integer timeout,Integer correlationID)`
- `changeregistration(Language language,Language hmiDisplayLanguage,Integer correlationID)`
- `putFileStream(InputStream is,String sdlFileName,Integer iOffset,Integer iLength) Deprecated`
- `putFileStream(InputStream inputStream,String fileName,Long offset,Long length)`
- `putFileStream(String sdlFileName,Integer iOffset,Integer iLength) Deprecated`
- `putFileStream(String fileName,Long offset,Long length)`
- `putFileStream(InputStream is,String sdlFileName,Integer iOffset,Integer iLength,FileType fileType,Boolean bPersistentFile,Boolean bSystemFile) Deprecated`
- `putFileStream(InputStream inputStream,String fileName,Long offset,Long length,FileType fileType,Boolean isPersistentFile,Boolean isSystemFile,OnPutFileUpdateListener cb)`
- `putFileStream(String sdlFileName,Integer iOffset,Integer iLength,FileType fileType,Boolean bPersistentFile,Boolean bSystemFile) Deprecated`
- `putFileStream(String fileName,Long offset,Long length,FileType fileType,Boolean isPersistentFile,Boolean isSystemFile,OnPutFileUpdateListener cb)`
- `putFileStream(String sPath,String sdlFileName,Integer iOffset,FileType fileType,Boolean bPersistentFile,Boolean bSystemFile,Integer iCorrelationID) Deprecated`
- `putFileStream(String path,String fileName,Long offset,FileType fileType,Boolean isPersistentFile,Boolean isSystemFile,Boolean isPayloadProtected,Integer correlationId,OnPutFileUpdateListener cb)`
- `putFileStream(InputStream is,String sdlFileName,Integer iOffset,Integer iLength,FileType fileType,Boolean bPersistentFile,Boolean bSystemFile,Integer iCorrelationID) Deprecated`
- `putFileStream(InputStream inputStream,String fileName,Long offset,Long length,FileType fileType,Boolean isPersistentFile,Boolean isSystemFile,Boolean isPayloadProtected,Integer correlationId)`
- `endPutFileStream()`
- `putfile(String sdlFileName,FileType fileType,Boolean persistentFile,byte[] fileData,Integer correlationID)`
- `deletefile(String sdlFileName,Integer correlationID)`
- `listfiles(Integer correlationID)`
- `setappicon(String sdlFileName,Integer correlationID)`

- `setDisplayLayout(String displayLayout,Integer correlationID)`
- `setDisplayLayout(String displayLayout,TemplateColorScheme dayColorScheme,TemplateColorScheme nightColorScheme,Integer correlationID)`
- `getSystemCapabilityManager()`
- `isCapabilitySupported(SystemCapabilityType systemCapabilityType)`
- `getCapability(SystemCapabilityType systemCapabilityType,OnSystemCapabilityListener scListener)`
- `getCapability(SystemCapabilityType systemCapabilityType)`
- `addOnSystemCapabilityListener(SystemCapabilityType systemCapabilityType,OnSystemCapabilityListener listener)`
- `removeOnSystemCapabilityListener(SystemCapabilityType systemCapabilityType,OnSystemCapabilityListener listener)`
- `getCurrentTransportType()`
- `setSdlSecurityClassList(java.util.List<java.lang.Class<? extends com.smartdevicelink.security.SdlSecurityBase>> list)`
- `setMinimumProtocolVersion(Version minimumProtocolVersion)`
- `setMinimumRPCVersion(Version minimumRPCVersion)`
- `isServiceTypeProtected(SessionType sType)`
- `addServiceListener(SessionType serviceType,ISdlServiceListener sdlServiceListener)`
- `removeServiceListener(SessionType serviceType,ISdlServiceListener sdlServiceListener)`
- `getAcceptedVideoParams()`
- `getProxyListener()`
- `getAppName()`
- `getNgnAppName()`
- `getAppID()`
- `getDeviceInfo()`
- `getInstanceDT()`
- `setConnectionDetails(String sDetails)`
- `getConnectionDetails()`
- `setPoliciesURL(String sText)`
- `getPoliciesURL()`
- `getIconResumed()`
- `getRegisterAppInterfaceResponse()`
- `getCurrentHMIStatus()`

- `getAuthToken()`

Fields

TAG

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

Constant Value

"SdlProxy"

MAX_SUPPORTED_RPC_VERSION

```
public static final com.smartdevicelink.util.Version  
MAX_SUPPORTED_RPC_VERSION
```

_appService

```
protected android.app.Service _appService
```

_advancedLifecycleManagementEnabled

```
protected java.lang.Boolean _advancedLifecycleManagementEnabled
```

_appInterfaceRegistered

```
protected java.lang.Boolean _appInterfaceRegistered
```

_preRegistered

```
protected java.lang.Boolean _preRegistered
```

_haveReceivedFirstFocusLevel

```
protected java.lang.Boolean _haveReceivedFirstFocusLevel
```

_haveReceivedFirstFocusLevelFull

```
protected java.lang.Boolean _haveReceivedFirstFocusLevelFull
```

_proxyDisposed

```
protected java.lang.Boolean _proxyDisposed
```

_sdlConnectionState

```
protected SdlConnectionState _sdlConnectionState
```

_sdlInterfaceAvailability

```
protected SdlInterfaceAvailability _sdlInterfaceAvailability
```

_hmiLevel

```
protected HMILevel _hmiLevel
```

lastHmiStatus

```
protected OnHMISStatus lastHmiStatus
```

_audioStreamingState

```
protected AudioStreamingState _audioStreamingState
```

_sdlMsgVersion

```
protected SdlMsgVersion _sdlMsgVersion
```

_autoActivateIdReturned

```
protected java.lang.String _autoActivateIdReturned
```

_sdlLanguage

```
protected Language _sdlLanguage
```

_hmiDisplayLanguage

```
protected Language _hmiDisplayLanguage
```

_prerecordedSpeech

```
protected java.util.List _prerecordedSpeech
```

_vehicleType

```
protected VehicleType _vehicleType
```

_systemSoftwareVersion

```
protected java.lang.String _systemSoftwareVersion
```

_diagModes

```
protected java.util.List _diagModes
```

firstTimeFull

```
protected java.lang.Boolean firstTimeFull
```

_proxyVersionInfo

```
protected java.lang.String _proxyVersionInfo
```

_bResumeSuccess

```
protected java.lang.Boolean _bResumeSuccess
```

_secList

```
protected java.util.List _secList
```

_systemCapabilityManager

```
protected SystemCapabilityManager _systemCapabilityManager
```

_iconResumed

```
protected java.lang.Boolean _iconResumed
```

raiResponse

```
protected RegisterAppInterfaceResponse raiResponse
```

protocolVersion

```
protected com.smartdevicelink.util.Version protocolVersion
```

rpcSpecVersion

```
protected com.smartdevicelink.util.Version rpcSpecVersion
```

rpcResponseListeners

```
protected android.util.SparseArray rpcResponseListeners
```

rpcNotificationListeners

```
protected android.util.SparseArray rpcNotificationListeners
```

rpcRequestListeners

```
protected android.util.SparseArray rpcRequestListeners
```

rpcListeners

```
protected android.util.SparseArray rpcListeners
```

manager

```
protected com.smartdevicelink.proxy.SdlProxyBase.  
VideoStreamingManager manager
```

authToken

```
protected java.lang.String authToken
```

Constructors

```
SdlProxyBase(  
    proxyListenerType listener,  
    Context context, String  
    appName, String shortAppName,  
    Boolean isMediaApp, Language  
    languageDesired, Language  
    hmiDisplayLanguageDesired,  
    java.util.Vector<AppHMIType>  
    appType, String appId,  
    BaseTransportConfig  
    transportConfig,  
    java.util.Vector<java.lang.String>  
    vrSynonyms,  
    java.util.Vector<TTSChunk>  
    ttsName, TemplateColorScheme  
    dayColorScheme,  
    TemplateColorScheme  
    nightColorScheme )
```

Used by the SdlManager

```
public SdlProxyBase(proxyListenerType listener, Context context, String
appName, String shortAppName, Boolean isMediaApp, Language
languageDesired, Language hmiDisplayLanguageDesired, java.util.Vector
<AppHMIType> appType, String appId, BaseTransportConfig
transportConfig, java.util.Vector<java.lang.String> vrSynonyms, java.util
.Vector<TTSCChunk> ttsName, TemplateColorScheme dayColorScheme,
TemplateColorScheme nightColorScheme)
```

Parameters

listener

Type of listener for this proxy base.

context

Application context.

appName

Client application name.

shortAppName

Client short application name.

isMediaApp

Flag that indicates that client application if media application or not.

languageDesired

Desired language.

hmiDisplayLanguageDesired

Desired language for HMI.

appType

Type of application.

appID

Application identifier.

transportConfig

Configuration of transport to be used by underlying connection.

vrSynonyms

List of synonyms.

ttsName

TTS name.

dayColorScheme

TemplateColorScheme for the day

nightColorScheme

TemplateColorScheme for the night

```
SdlProxyBase(  
    proxyListenerType listener,  
    SdlProxyConfigurationResources  
    sdlProxyConfigurationResources,  
    boolean  
    enableAdvancedLifecycleManagement,  
    String appName,  
    java.util.Vector<TTSChunk>  
    ttsName, String  
    ngnMediaScreenAppName,  
    java.util.Vector<java.lang.String>  
    vrSynonyms, Boolean  
    isMediaApp, SdlMsgVersion  
    sdlMsgVersion, Language  
    languageDesired, Language  
    hmiDisplayLanguageDesired,  
    java.util.Vector<AppHMType>  
    appType, String appId, String  
    autoActivateID, boolean  
    callbackToUIThread,
```


BaseTransportConfig transportConfig)

Constructor.

```
protected SdlProxyBase(proxyListenerType listener,  
SdlProxyConfigurationResources sdlProxyConfigurationResources,  
boolean enableAdvancedLifecycleManagement,String appName,java.  
util.Vector<TTSCChunk> ttsName,String ngnMediaScreenAppName,java.  
util.Vector<java.lang.String> vrSynonyms,Boolean isMediaApp,  
SdlMsgVersion sdlMsgVersion,Language languageDesired,Language  
hmiDisplayLanguageDesired,java.util.Vector<AppHMIMType> appType,  
String appId,String autoActivateID,boolean callbackToUIThread,  
BaseTransportConfig transportConfig)
```

Parameters

listener

Type of listener for this proxy base.

sdlProxyConfigurationResources

Configuration resources for this proxy.

enableAdvancedLifecycleManagement

Flag that ALM should be enabled or not.

appName

Client application name.
ttsName TTS name.
ngnMediaScreenAppName Media Screen Application name.
vrSynonyms List of synonyms.
isMediaApp Flag that indicates that client application if media application or not.
sdIMsgVersion Version of Sdl Message.
languageDesired Desired language.
hmiDisplayLanguageDesired Desired language for HMI.
appType Type of application.

appID

Application identifier.

autoActivateID

Auto activation identifier.

callbackToUIThread

Flag that indicates that this proxy should send callback to UI thread or not.

transportConfig

Configuration of transport to be used by underlying connection.

```
SdlProxyBase(  
    proxyListenerType listener,  
    SdlProxyConfigurationResources  
    sdlProxyConfigurationResources,  
    boolean  
    enableAdvancedLifecycleManagement,  
    String appName,  
    java.util.Vector<TTSChunk>  
    ttsName, String  
    ngnMediaScreenAppName,  
    java.util.Vector<java.lang.String>  
    vrSynonyms, Boolean  
    isMediaApp, SdlMsgVersion  
    sdlMsgVersion, Language  
    languageDesired, Language  
    hmiDisplayLanguageDesired,  
    java.util.Vector<AppHMIType>  
    appType, String appId, String  
    autoActivateID,  
    TemplateColorScheme  
    dayColorScheme,
```

TemplateColorScheme
nightColorScheme, boolean
callbackToUiThread, boolean
preRegister, String sHashID,
Boolean bEnableResume,
BaseTransportConfig
transportConfig)

```
protected SdlProxyBase(proxyListenerType listener,  
SdlProxyConfigurationResources sdlProxyConfigurationResources,  
boolean enableAdvancedLifecycleManagement,String appName,java.  
util.Vector<TTSCChunk> ttsName,String ngnMediaScreenAppName,java.  
util.Vector<java.lang.String> vrSynonyms,Boolean isMediaApp,  
SdlMsgVersion sdlMsgVersion,Language languageDesired,Language  
hmiDisplayLanguageDesired,java.util.Vector<AppHMIDType> appType,  
String applID,String autoActivateID,TemplateColorScheme  
dayColorScheme,TemplateColorScheme nightColorScheme,boolean  
callbackToUiThread,boolean preRegister,String sHashID,Boolean  
bEnableResume,BaseTransportConfig transportConfig)
```

Parameters

listener

sdlProxyConfigurationResources

enableAdvancedLifecycleManagement

appName

ttsName

ngnMediaScreenAppName

vrSynonyms

isMediaApp

sdIMsgVersion

languageDesired

hmiDisplayLanguageDesired

appType

appID
autoActivateID
dayColorScheme
nightColorScheme
callbackToUIThread
preRegister
sHashID
bEnableResume
transportConfig

```
SdlProxyBase(  
    proxyListenerType listener,  
    SdlProxyConfigurationResources  
    sdlProxyConfigurationResources,  
    boolean  
    enableAdvancedLifecycleManagement,  
    String appName,  
    java.util.Vector<TTSChunk>  
    ttsName, String  
    ngnMediaScreenAppName,  
    java.util.Vector<java.lang.String>  
    vrSynonyms, Boolean  
    isMediaApp, SdlMsgVersion  
    sdlMsgVersion, Language  
    languageDesired, Language  
    hmiDisplayLanguageDesired,  
    java.util.Vector<AppHMType>  
    appType, String appId, String  
    autoActivateID, boolean  
    callbackToUIThread, boolean
```


preRegister, BaseTransportConfig transportConfig)

Constructor.

```
protected SdlProxyBase(proxyListenerType listener,  
SdlProxyConfigurationResources sdlProxyConfigurationResources,  
boolean enableAdvancedLifecycleManagement,String appName,java.  
util.Vector<TTSCChunk> ttsName,String ngnMediaScreenAppName,java.  
util.Vector<java.lang.String> vrSynonyms,Boolean isMediaApp,  
SdlMsgVersion sdlMsgVersion,Language languageDesired,Language  
hmiDisplayLanguageDesired,java.util.Vector<AppHMIMType> appType,  
String appId,String autoActivateId,boolean callbackToUiThread,boolean  
preRegister,BaseTransportConfig transportConfig)
```

Parameters

listener

Type of listener for this proxy base.

sdlProxyConfigurationResources

Configuration resources for this proxy.

enableAdvancedLifecycleManagement

Flag that ALM should be enabled or not.

appName

Client application name.
ttsName TTS name.
ngnMediaScreenAppName Media Screen Application name.
vrSynonyms List of synonyms.
isMediaApp Flag that indicates that client application if media application or not.
sdIMsgVersion Version of Sdl Message.
languageDesired Desired language.
hmiDisplayLanguageDesired Desired language for HMI.
appType Type of application.

appID

Application identifier.

autoActivateID

Auto activation identifier.

callbackToUIThread

Flag that indicates that this proxy should send callback to UI thread or not.

preRegister

Flag that indicates that this proxy should be pre-registered or not.

transportConfig

Configuration of transport to be used by underlying connection.

```
SdlProxyBase(  
    proxyListenerType listener,  
    SdlProxyConfigurationResources  
    sdlProxyConfigurationResources,  
    boolean  
    enableAdvancedLifecycleManagement,  
    String appName,  
    java.util.Vector<TTSChunk>  
    ttsName, String  
    ngnMediaScreenAppName,  
    java.util.Vector<java.lang.String>  
    vrSynonyms, Boolean  
    isMediaApp, SdlMsgVersion  
    sdlMsgVersion, Language  
    languageDesired, Language  
    hmiDisplayLanguageDesired,  
    java.util.Vector<AppHMIType>  
    appType, String appId, String  
    autoActivateID,  
    TemplateColorScheme  
    dayColorScheme,
```

TemplateColorScheme nightColorScheme, boolean callbackToUiThread, boolean preRegister, BaseTransportConfig transportConfig)

Constructor.

```
protected SdlProxyBase(proxyListenerType listener,  
SdlProxyConfigurationResources sdlProxyConfigurationResources,  
boolean enableAdvancedLifecycleManagement,String appName,java.  
util.Vector<TTSCChunk> ttsName,String ngnMediaScreenAppName,java.  
util.Vector<java.lang.String> vrSynonyms,Boolean isMediaApp,  
SdlMsgVersion sdlMsgVersion,Language languageDesired,Language  
hmiDisplayLanguageDesired,java.util.Vector<AppHMIType> appType,  
String appId,String autoActivateID,TemplateColorScheme  
dayColorScheme,TemplateColorScheme nightColorScheme,boolean  
callbackToUiThread,boolean preRegister,BaseTransportConfig  
transportConfig)
```

Parameters

listener

Type of listener for this proxy base.

sdlProxyConfigurationResources

Configuration resources for this proxy.

enableAdvancedLifecycleManagement

Flag that ALM should be enabled or not.

appName

Client application name.

ttsName

TTS name.

ngnMediaScreenAppName

Media Screen Application name.

vrSynonyms

List of synonyms.

isMediaApp

Flag that indicates that client application if media application or not.

sdIMsgVersion

Version of Sdl Message.

languageDesired

Desired language.

hmiDisplayLanguageDesired

Desired language for HMI.

appType

Type of application.

appID

Application identifier.

autoActivateID

Auto activation identifier.

dayColorScheme

Day color scheme.

nightColorScheme

callbackToUIThread

Flag that indicates that this proxy should send callback to UI thread or not.

preRegister

Flag that indicates that this proxy should be pre-registered or not.

transportConfig

Configuration of transport to be used by underlying connection.

Methods

addPutFileResponseListener(
IPutFileResponseListener
_putFileListener)

```
public void addPutFileResponseListener(IPutFileResponseListener  
_putFileListener)
```

Parameters

_putFileListener

remPutFileResponseListener(
IPutFileResponseListener
_putFileListener)


```
public void remPutFileResponseListener(IPutFileResponseListener  
_putFileListener)
```

Parameters

_putFileListener

getIsConnected()

```
public java.lang.Boolean getIsConnected()
```

getAppInterfaceRegistered()

Returns whether the application is registered in SDL. Note: for testing purposes, it's possible that the connection is established, but the application is not registered.

```
public java.lang.Boolean getAppInterfaceRegistered()
```

initializeProxy()

```
public void initializeProxy() throws SdlException
```

Throws

- SdlException

forceOnConnected()

This method will fake the multiplex connection event

```
public void forceOnConnected()
```

sendTransportBroadcast()

```
public void sendTransportBroadcast()
```

enableSiphonDebug()

Public method to enable the siphon transport

```
public void enableSiphonDebug()
```

disableSiphonDebug()

Public method to disable the Siphon Trace Server

```
public void disableSiphonDebug()
```

enableDebugTool()

Public method to enable the Debug Tool

```
public static void enableDebugTool()
```

disableDebugTool()

Public method to disable the Debug Tool

```
public static void disableDebugTool()
```

isDebugEnabled()

Public method to determine Debug Tool enabled

```
public static boolean isDebugEnabled()
```

isAudioStreamTransportAvailable() ()

Check to see if it a transport is available to perform audio streaming.

NOTE: This is only for the audio streaming service, not regular streaming of media playback.

```
public boolean isAudioStreamTransportAvailable()
```

isVideoStreamTransportAvailable()

Check to see if it a transport is available to perform video streaming.

```
public boolean isVideoStreamTransportAvailable()
```

close()

Deprecated

```
public void close() throws SdlException
```

Throws

- SdlException

dispose()

Terminates the App's Interface Registration, closes the transport connection, ends the protocol session, and frees any resources used by the proxy.

```
public void dispose() throws SdlException
```

Throws

- SdlException

cycleProxy(SdlDisconnectedReason disconnectedReason)

```
protected void cycleProxy(SdlDisconnectedReason disconnectedReason  
)
```

Parameters

disconnectedReason

getProtocolVersion()

Get the SDL protocol spec version being used

```
public com.smartdevicelink.proxy.Version getProtocolVersion()
```

serializeJSON(RPCMessage msg)

```
public java.lang.String serializeJSON(RPCMessage msg)
```

Parameters

msg

onPacketProgress(int correlationId, long bytesWritten, long totalSize)

Only call this method for a PutFile response. It will cause a class cast exception if not.

```
public void onPacketProgress(int correlationId,long bytesWritten,long totalSize)
```

Parameters

correlationId

correlation id of the packet being updated

bytesWritten

how many bytes were written

totalSize

the total size in bytes

addOnRPCResponseListener(OnRPCResponseListener listener, int correlationId, int totalSize)

Add a listener that will receive the response to the specific RPCRequest sent with the corresponding correlation id

```
public void addOnRPCResponseListener(OnRPCResponseListener  
listener,int correlationId,int totalSize)
```

Parameters

listener

that will get called back when a response is received

correlationId

of the RPCRequest that was sent

totalSize

only include if this is an OnPutFileUpdateListener. Otherwise it will be ignored.

getResponseListeners()

```
public android.util.SparseArray getResponseListeners()
```

onRPCNotificationReceived(RPCNotification notification)

```
public boolean onRPCNotificationReceived(RPCNotification notification)
```

Parameters

notification

onRPCReceived(RPCMessage message)

```
public boolean onRPCReceived(RPCMessage message)
```

Parameters

message

onRPCRequestReceived(RPCRequest request)

```
public boolean onRPCRequestReceived(RPCRequest request)
```

Parameters

request

addOnRPCNotificationListener(FunctionID notificationId, OnRPCNotificationListener listener)

This will add a listener for the specific type of notification. As of now it will only allow a single listener per notification function id

```
public void addOnRPCNotificationListener(FunctionID notificationId,  
OnRPCNotificationListener listener)
```

Parameters

notificationId

The notification type that this listener is designated for

listener

The listener that will be called when a notification of the provided type is received

addOnRPCListener(FunctionID messageId, OnRPCListener listener)

This will add a listener for the specific type of message. As of now it will only allow a single listener per request function id

```
public void addOnRPCListener(FunctionID messageId, OnRPCListener  
listener)
```

Parameters

messageId

The message type that this listener is designated for

listener

The listener that will be called when a request of the provided type is received

removeOnRPCListener(FunctionID messageId, OnRPCListener listener)

```
public boolean removeOnRPCListener(FunctionID messageId,  
OnRPCListener listener)
```

Parameters

messageId
listener

addOnRPCRequestListener(FunctionID requestId, OnRPCRequestListener listener)

This will add a listener for the specific type of request. As of now it will only allow a single listener per request function id

```
public void addOnRPCRequestListener(FunctionID requestId,  
OnRPCRequestListener listener)
```

Parameters

requestId

The request type that this listener is designated for

listener

The listener that will be called when a request of the provided type is received

removeOnRPCNotificationListener (FunctionID notificationId)

Deprecated

This method is no longer valid and will not remove the listener for the supplied notification id

```
public void removeOnRPCNotificationListener(FunctionID notificationId)
```

Parameters

notificationId

n/a

removeOnRPCNotificationListener
(FunctionID notificationId,
OnRPCNotificationListener
listener)

```
public boolean removeOnRPCNotificationListener(FunctionID  
notificationId, OnRPCNotificationListener listener)
```

Parameters

notificationId

listener

removeOnRPCRequestListener(FunctionID requestId, OnRPCRequestListener listener)

```
public boolean removeOnRPCRequestListener(FunctionID requestId,  
OnRPCRequestListener listener)
```

Parameters

requestId
listener

getSdlMsgVersion()

Get SDL Message Version

```
public SdlMsgVersion getSdlMsgVersion() throws SdlException
```

Throws

- `SdlException`

```
sendSequentialRequests(  
    java.util.List<? extends  
    RPCMessage> rpcs,  
    OnMultipleRequestListener  
    listener )
```

Takes a list of `RPCMessages` and sends it to SDL in a synchronous fashion. Responses are captured through callback on `OnMultipleRequestListener`. For sending requests asynchronously, use `sendRequests`

NOTE: This will override any listeners on individual RPCs

```
public void sendSequentialRequests(java.util.List<? extends  
RPCMessage> rpcs, OnMultipleRequestListener listener) throws  
SdlException
```

Parameters

`rpcs`

is the list of `RPCMessages` being sent

listener

listener for updates and completions

Throws

- SdlException

sendRequests(java.util.List<? extends RPCMessage> rpcs, OnMultipleRequestListener listener)

Takes a list of RPCMessages and sends it to SDL. Responses are captured through callback on OnMultipleRequestListener. For sending requests synchronously, use sendSequentialRequests

NOTE: This will override any listeners on individual RPCs

```
public void sendRequests(java.util.List<? extends RPCMessage> rpcs,
OnMultipleRequestListener listener) throws SdlException
```

Parameters

`rpcs`

is the list of RPCMessages being sent

`listener`

listener for updates and completions

Throws

- `SdlException`

sendRPC(RPCMessage message)

```
public void sendRPC(RPCMessage message) throws SdlException
```

Parameters

message

Throws

- `SdlException`

sendRPCRequest(RPCRequest request)

Deprecated

Takes an `RPCRequest` and sends it to SDL. Responses are captured through callback on `IProxyListener`.

```
public void sendRPCRequest(RPCRequest request) throws SdlException
```

Parameters

request

is the RPCRequest being sent

Throws

- SdlException

notifyProxyClosed(String info,
Exception e,
SdlDisconnectedReason reason)

```
protected void notifyProxyClosed(String info,Exception e,  
SdlDisconnectedReason reason)
```

Parameters

info
e
reason

startRPCStream(InputStream is,
RPCRequest msg)

```
public boolean startRPCStream(InputStream is,RPCRequest msg)
```

Parameters

is

msg

startRPCStream(RPCRequest
msg)

```
public java.io.OutputStream startRPCStream(RPCRequest msg)
```

Parameters

msg

endRPCStream()

```
public void endRPCStream()
```

createFutureTask(com.smartdevicelink.proxy.SdlProxyBase callMethod)

```
public java.util.concurrent.FutureTask createFutureTask(com.  
smartdevicelink.proxy.SdlProxyBase<proxyListenerType>.  
CallableMethod callMethod)
```


Parameters

callMethod

createScheduler()

```
public java.util.concurrent.ScheduledExecutorService createScheduler()
```

startService(SessionType serviceType, boolean isEncrypted)

```
public void startService(SessionType serviceType,boolean isEncrypted)
```

Parameters

serviceType
isEncrypted

endService(SessionType
serviceType)

```
public void endService(SessionType serviceType)
```

Parameters

serviceType

startH264(InputStream is, boolean isEncrypted)

Deprecated

```
public boolean startH264(InputStream is,boolean isEncrypted)
```

Parameters

is
isEncrypted

startH264(boolean isEncrypted)

Deprecated

```
public java.io.OutputStream startH264(boolean isEncrypted)
```

Parameters

isEncrypted

endH264()

Deprecated

Closes the opened video service (serviceType 11)

```
public boolean endH264()
```

pausePCM()

Deprecated

Pauses the stream for the opened audio service (serviceType 10)

```
public boolean pausePCM()
```

pauseH264()

Deprecated

Pauses the stream for the opened video service (serviceType 11)

```
public boolean pauseH264()
```

resumePCM()

Deprecated

Resumes the stream for the opened audio service (serviceType 10)

```
public boolean resumePCM()
```

resumeH264()

Deprecated

Resumes the stream for the opened video service (serviceType 11)

```
public boolean resumeH264()
```

startPCM(InputStream is, boolean isEncrypted)

Deprecated

Opens the audio service (serviceType 10) and subsequently streams raw PCM audio from an InputStream provided by the app

```
public boolean startPCM(InputStream is,boolean isEncrypted)
```

Parameters

is
isEncrypted

startPCM(boolean isEncrypted)

Deprecated

Opens the audio service (serviceType 10) and subsequently provides an OutputStream to the app

```
public java.io.OutputStream startPCM(boolean isEncrypted)
```

Parameters

isEncrypted

endPCM()

Deprecated

Closes the opened audio service (serviceType 10)

```
public boolean endPCM()
```

startVideoStream(boolean isEncrypted, VideoStreamingParameters parameters)

Opens a video service (service type 11) and subsequently provides an IVideoStreamListener to the app to send video data. The supplied VideoStreamingParameters will be set as desired parameters that will be used to negotiate

NOTE: IF USING SECONDARY TRANSPORTS, THE VIDEO SERVICE MUST BE STARTED BEFORE CALLING THIS METHOD. USE A `ISdlServiceListener` TO BE NOTIFIED THAT IT STARTS THEN CALL THIS METHOD TO START STREAMING. ADD A LISTENER USE {@link #addServiceListener(SessionType, ISdlServiceListener)}.

```
public IVideoStreamListener startVideoStream(boolean isEncrypted,  
VideoStreamingParameters parameters)
```

Parameters

isEncrypted

Specify true if packets on this service have to be encrypted

parameters

Video streaming parameters including: codec which will be used for streaming (currently, only VideoStreamingCodec.H264 is accepted), height and width of the video in pixels.

endVideoStream()

Closes the opened video service (serviceType 11)

```
public boolean endVideoStream()
```

pauseVideoStream()

Pauses the stream for the opened video service (serviceType 11)

```
public boolean pauseVideoStream()
```

resumeVideoStream()

Resumes the stream for the opened video service (serviceType 11)

```
public boolean resumeVideoStream()
```

createOpenGLInputSurface(int frameRate, int iFrameInterval, int width, int height, int bitrate, boolean isEncrypted)

Opens the video service (serviceType 11) and creates a Surface (used for streaming video) with input parameters provided by the app

```
public android.view.Surface createOpenGLInputSurface(int frameRate,  
int iFrameInterval,int width,int height,int bitrate,boolean isEncrypted)
```

Parameters

frameRate

- specified rate of frames to utilize for creation of Surface

iFrameInterval

- specified interval to utilize for creation of Surface

width

- specified width to utilize for creation of Surface

height

- specified height to utilize for creation of Surface

bitrate

- specified bitrate to utilize for creation of Surface

isEncrypted

```
startRemoteDisplayStream(  
Context context,  
java.lang.Class<? extends  
com.smartdevicelink.streaming.video.S  
remoteDisplay,  
VideoStreamingParameters  
parameters, boolean encrypted )
```

Starts streaming a remote display to the module if there is a connected session. This method of streaming requires the device to be on API level 19 or higher

```
public void startRemoteDisplayStream(Context context, java.lang.Class  
<? extends com.smartdevicelink.streaming.video.SdlRemoteDisplay>  
remoteDisplay, VideoStreamingParameters parameters, boolean  
encrypted)
```

Parameters

context

a context that can be used to create the remote display

remoteDisplay

class object of the remote display. This class will be used to create an instance of the remote display and will be projected to the module

parameters

streaming parameters to be used when streaming. If null is sent in, the default/optimized options will be used. If you are unsure about what parameters to be used it is best to just send null and let the system determine what works best for the currently connected module.

encrypted

a flag of if the stream should be encrypted. Only set if you have a supplied encryption library that the module can understand.

stopRemoteDisplayStream()

Stops the remote display stream if one has been started

```
public void stopRemoteDisplayStream()
```

startEncoder()

Starts the MediaCodec encoder utilized in conjunction with the Surface returned via the createOpenGLInputSurface method

```
public void startEncoder()
```

releaseEncoder()

Releases the MediaCodec encoder utilized in conjunction with the Surface returned via the createOpenGLInputSurface method

```
public void releaseEncoder()
```

drainEncoder(boolean endOfStream)

Releases the MediaCodec encoder utilized in conjunction with the Surface returned via the createOpenGLInputSurface method

```
public void drainEncoder(boolean endOfStream)
```

Parameters

endOfStream

startAudioStream(boolean isEncrypted, AudioStreamingCodec codec, AudioStreamingParams params)

Opens a audio service (service type 10) and subsequently provides an IAudioStreamListener to the app to send audio data. Currently information passed by "params" are ignored, since Audio Streaming feature lacks capability negotiation mechanism. App should configure audio stream data to align with head unit's capability by checking (upcoming) pcmCapabilities. The default format is in 16kHz and 16 bits.

```
public IAudioStreamListener startAudioStream(boolean isEncrypted,  
AudioStreamingCodec codec,AudioStreamingParams params)
```

Parameters

isEncrypted

Specify true if packets on this service have to be encrypted

codec

Audio codec which will be used for streaming. Currently, only `AudioStreamingCodec.LPCM` is accepted.

params

(Reserved for future use) Additional configuration information for each codec. If "codec" is `AudioStreamingCodec.LPCM`, "params" must be an instance of `LPCMParams` class.

endAudioStream()

Closes the opened audio service (serviceType 10)

```
public boolean endAudioStream()
```

pauseAudioStream()

Pauses the stream for the opened audio service (serviceType 10)

```
public boolean pauseAudioStream()
```

resumeAudioStream()

Resumes the stream for the opened audio service (serviceType 10)

```
public boolean resumeAudioStream()
```

setAppService(Service mService)

```
public void setAppService(Service mService)
```


Parameters

mService

startProtectedRPCService()

```
public boolean startProtectedRPCService()
```

getLockScreenIcon(LockScreenManager.OnLockScreenIcon l)

```
public void getLockScreenIcon(LockScreenManager.  
OnLockScreenIconDownloadedListener l)
```

Parameters

|

addCommand(Integer
commandID, String menuText,
Integer parentID, Integer
position,
java.util.Vector<java.lang.String>
vrCommands, String IconValue,
ImageType IconType, Integer
correlationID)

Sends an AddCommand RPCRequest to SDL. Responses are captured through callback on IProxyListener.

```
public void addCommand(Integer commandID,String menuText,Integer  
parentID,Integer position,java.util.Vector<java.lang.String>  
vrCommands,String IconValue,ImageType IconType,Integer correlationID  
) throws SdlException
```

Parameters

commandID

-Unique command ID of the command to add.

menuText

-Menu text for optional sub value containing menu parameters.

parentID

-Menu parent ID for optional sub value containing menu parameters.

position

-Menu position for optional sub value containing menu parameters.

vrCommands

-VR synonyms for this AddCommand.

IconValue

-A static hex icon value or the binary image file name identifier (sent by the PutFile RPC).

IconType

-Describes whether the image is static or dynamic

correlationID

-A unique ID that correlates each RPCRequest and RPCResponse

Throws

- SdlException

addCommand(Integer
commandID, String menuText,
Integer position,
java.util.Vector<java.lang.String>
vrCommands, String IconValue,
ImageType IconType, Integer
correlationID)

Sends an AddCommand RPCRequest to SDL. Responses are captured through callback on IProxyListener.

```
public void addCommand(Integer commandID,String menuText,Integer  
position,java.util.Vector<java.lang.String> vrCommands,String  
IconValue,ImageType IconType,Integer correlationID) throws  
SdlException
```

Parameters

commandID

-Unique command ID of the command to add.

menuText

-Menu text for optional sub value containing menu parameters.

position

-Menu position for optional sub value containing menu parameters.

vrCommands

-VR synonyms for this AddCommand.

IconValue

-A static hex icon value or the binary image file name identifier (sent by the PutFile RPC).

IconType

-Describes whether the image is static or dynamic

correlationID

-A unique ID that correlates each RPCRequest and RPCResponse

Throws

- `SdlException`

`addCommand(Integer
commandID, String menuText,
Integer position, String
IconValue, ImageType IconType,
Integer correlationID)`

Sends an AddCommand RPCRequest to SDL. Responses are captured through callback on IProxyListener.

```
public void addCommand(Integer commandID,String menuText,Integer  
position,String IconValue,ImageType IconType,Integer correlationID)  
throws SdlException
```

Parameters

`commandID`

-Unique command ID of the command to add.

menuText

-Menu text for optional sub value containing menu parameters.

position

-Menu position for optional sub value containing menu parameters.

IconValue

-A static hex icon value or the binary image file name identifier (sent by the PutFile RPC).

IconType

-Describes whether the image is static or dynamic

correlationID

-A unique ID that correlates each RPCRequest and RPCResponse

Throws

- SdlException

addCommand(Integer commandID, String menuText, String IconValue, ImageType IconType, Integer correlationID)

Sends an AddCommand RPCRequest to SDL. Responses are captured through callback on IProxyListener.

```
public void addCommand(Integer commandID,String menuText,String  
IconValue,ImageType IconType,Integer correlationID) throws  
SdlException
```

Parameters

commandID

-Unique command ID of the command to add.

menuText

-Menu text for optional sub value containing menu parameters.

IconValue

-A static hex icon value or the binary image file name identifier (sent by the PutFile RPC).

IconType

-Describes whether the image is static or dynamic

correlationID

-A unique ID that correlates each RPCRequest and RPCResponse

Throws

- SdlException

`addCommand(Integer
commandID, String menuText,
java.util.Vector<java.lang.String>
vrCommands, String IconValue,
ImageType IconType, Integer
correlationID)`

Sends an AddCommand RPCRequest to SDL. Responses are captured through callback on IProxyListener.

```
public void addCommand(Integer commandID,String menuText,java.util.  
.Vector<java.lang.String> vrCommands,String IconValue,ImageType  
IconType,Integer correlationID) throws SdlException
```

Parameters

commandID

-Unique command ID of the command to add.

menuText

-Menu text for optional sub value containing menu parameters.

vrCommands

-VR synonyms for this AddCommand.

IconValue

-A static hex icon value or the binary image file name identifier (sent by the PutFile RPC).

IconType

-Describes whether the image is static or dynamic

correlationID

-A unique ID that correlates each RPCRequest and RPCResponse

Throws

- SdlException

addCommand(Integer
commandID,
java.util.Vector<java.lang.String>
vrCommands, String IconValue,
ImageType IconType, Integer
correlationID)

Sends an AddCommand RPCRequest to SDL. Responses are captured through callback on IProxyListener.

```
public void addCommand(Integer commandID,java.util.Vector<java.  
lang.String> vrCommands,String IconValue,ImageType IconType,Integer  
correlationID) throws SdlException
```

Parameters

commandID

-Unique command ID of the command to add.
vrCommands -VR synonyms for this AddCommand.
IconValue -A static hex icon value or the binary image file name identifier (sent by the PutFile RPC).
IconType -Describes whether the image is static or dynamic
correlationID -A unique ID that correlates each RPCRequest and RPCResponse

Throws

- SdlException

addCommand(Integer
commandID, String menuText,
Integer parentID, Integer
position,
java.util.Vector<java.lang.String>
vrCommands, Integer
correlationID)

Sends an AddCommand RPCRequest to SDL. Responses are captured through callback on IProxyListener.

```
public void addCommand(Integer commandID,String menuText,Integer  
parentID,Integer position,java.util.Vector<java.lang.String>  
vrCommands,Integer correlationID) throws SdlException
```

Parameters

commandID

-Unique command ID of the command to add.

menuText

-Menu text for optional sub value containing menu parameters.

parentID

-Menu parent ID for optional sub value containing menu parameters.

position

-Menu position for optional sub value containing menu parameters.

vrCommands

-VR synonyms for this AddCommand.

correlationID

-A unique ID that correlates each RPCRequest and RPCResponse

Throws

- `SdlException`

addCommand(Integer commandID, String menuText, Integer position, java.util.Vector<java.lang.String> vrCommands, Integer correlationID)

Sends an AddCommand RPCRequest to SDL. Responses are captured through callback on IProxyListener.

```
public void addCommand(Integer commandID,String menuText,Integer  
position,java.util.Vector<java.lang.String> vrCommands,Integer  
correlationID) throws SdlException
```

Parameters

commandID

-Unique command ID of the command to add.

menuText

-Menu text for optional sub value containing menu parameters.

position

-Menu position for optional sub value containing menu parameters.

vrCommands

-VR synonyms for this AddCommand.

correlationID

-A unique ID that correlates each RPCRequest and RPCResponse

Throws

- `SdlException`

addCommand(Integer commandID, String menuText, Integer position, Integer correlationID)

Sends an AddCommand RPCRequest to SDL. Responses are captured through callback on IProxyListener.

```
public void addCommand(Integer commandID,String menuText,Integer  
position,Integer correlationID) throws SdlException
```


Parameters

commandID

-Unique command ID of the command to add.

menuText

-Menu text for optional sub value containing menu parameters.

position

-Menu position for optional sub value containing menu parameters.

correlationID

-A unique ID that correlates each RPCRequest and RPCResponse

Throws

- SdlException

addCommand(Integer commandID, String menuText, Integer correlationID)

Sends an AddCommand RPCRequest to SDL. Responses are captured through callback on IProxyListener.

```
public void addCommand(Integer commandID,String menuText,Integer correlationID) throws SdlException
```

Parameters

commandID

-Unique command ID of the command to add.

menuText

-Menu text for optional sub value containing menu parameters.

correlationID

-A unique ID that correlates each RPCRequest and RPCResponse

Throws

- `SdlException`

`addCommand(Integer
commandID, String menuText,
java.util.Vector<java.lang.String>
vrCommands, Integer
correlationID)`

Sends an AddCommand RPCRequest to SDL. Responses are captured through callback on IProxyListener.

```
public void addCommand(Integer commandID,String menuText,java.util.  
.Vector<java.lang.String> vrCommands,Integer correlationID) throws  
SdlException
```

Parameters

`commandID`

-Unique command ID of the command to add.

menuText

-Menu text for optional sub value containing menu parameters.

vrCommands

-VR synonyms for this AddCommand.

correlationID

-A unique ID that correlates each RPCRequest and RPCResponse

Throws

- SdlException

addCommand(Integer
commandID,
java.util.Vector<java.lang.String>
vrCommands, Integer
correlationID)

Sends an AddCommand RPCRequest to SDL. Responses are captured through callback on IProxyListener.

```
public void addCommand(Integer commandID, java.util.Vector<java.  
lang.String> vrCommands, Integer correlationID) throws SdlException
```

Parameters

commandID

-Unique command ID of the command to add.

vrCommands

-VR synonyms for this AddCommand.

correlationID

-A unique ID that correlates each RPCRequest and RPCResponse

Throws

- SdlException

addSubMenu(Integer menuID,
String menuName, Integer
position, Image menuIcon,
Integer correlationID)

Sends an AddSubMenu RPCRequest to SDL. Responses are captured through callback on IProxyListener.

```
public void addSubMenu(Integer menuID,String menuName,Integer position,Image menuIcon,Integer correlationID) throws SdlException
```

Parameters

menuID

-Unique ID of the sub menu to add.

menuName

-Text to show in the menu for this sub menu.

position

-Position within the items that are at top level of the in application menu.

menuIcon

-Image to be shown along with the submenu item

correlationID

-A unique ID that correlates each RPCRequest and RPCResponse

Throws

- `SdlException`

`addSubMenu( Integer menuID, String menuName, Integer position, Integer correlationID )`

Deprecated

Sends an AddSubMenu RPCRequest to SDL. Responses are captured through callback on IProxyListener.

```
public void addSubMenu(Integer menuID,String menuName,Integer position,Integer correlationID) throws SdlException
```

Parameters

menuID

-Unique ID of the sub menu to add.

menuName

-Text to show in the menu for this sub menu.

position

-Position within the items that are at top level of the in application menu.

correlationID

-A unique ID that correlates each RPCRequest and RPCResponse

Throws

- SdlException

addSubMenu(Integer menuID, String menuName, Integer correlationID)

Deprecated

Sends an AddSubMenu RPCRequest to SDL. Responses are captured through callback on IProxyListener.

```
public void addSubMenu(Integer menuID,String menuName,Integer correlationID) throws SdlException
```


Parameters

menuID

-Unique ID of the sub menu to add.

menuName

-Text to show in the menu for this sub menu.

correlationID

-A unique ID that correlates each RPCRequest and RPCResponse

Throws

- SdlException

```
alert( String ttsText, String
alertText1, String alertText2,
String alertText3, Boolean
playTone, Integer duration,
java.util.Vector<SoftButton>
softButtons, Integer correlationID
)
```

Sends an Alert RPCRequest to SDL. Responses are captured through callback on IProxyListener.

```
public void alert(String ttsText,String alertText1,String alertText2,String
alertText3,Boolean playTone,Integer duration,java.util.Vector<
SoftButton> softButtons,Integer correlationID) throws SdlException
```

Parameters

ttsText

-The text to speech message in the form of a string.

alertText1

-The first line of the alert text field.

alertText2

-The second line of the alert text field.

alertText3

-The optional third line of the alert text field.

playTone

-Defines if tone should be played.

duration

-Timeout in milliseconds.

softButtons

-A list of App defined SoftButtons.

correlationID

-A unique ID that correlates each RPCRequest and RPCResponse

Throws

- `SdlException`

```
alert(  
    java.util.Vector<TTSCChunk>  
    ttsChunks, String alertText1,  
    String alertText2, String  
    alertText3, Boolean playTone,  
    Integer duration,  
    java.util.Vector<SoftButton>  
    softButtons, Integer correlationID  
)
```

Sends an Alert RPCRequest to SDL. Responses are captured through callback on IProxyListener.

```
public void alert(java.util.Vector<TTSCChunk> ttsChunks,String  
    alertText1,String alertText2,String alertText3,Boolean playTone,Integer  
    duration,java.util.Vector<SoftButton> softButtons,Integer correlationID)  
    throws SdlException
```

Parameters

ttsChunks

-Text/phonemes to speak in the form of ttsChunks.

alertText1 -The first line of the alert text field.
alertText2 -The second line of the alert text field.
alertText3 -The optional third line of the alert text field.
playTone -Defines if tone should be played.
duration -Timeout in milliseconds.
softButtons -A list of App defined SoftButtons.
correlationID -A unique ID that correlates each RPCRequest and RPCResponse

Throws

- SdlException

alert(String ttsText, Boolean playTone, java.util.Vector<SoftButton> softButtons, Integer correlationID)

Sends an Alert RPCRequest to SDL. Responses are captured through callback on IProxyListener.

```
public void alert(String ttsText, Boolean playTone, java.util.Vector<SoftButton> softButtons, Integer correlationID) throws SdlException
```

Parameters

ttsText

-The text to speech message in the form of a string.

playTone

-Defines if tone should be played.

softButtons

-A list of App defined SoftButtons.

correlationID

-A unique ID that correlates each RPCRequest and RPCResponse

Throws

- SdlException

```
alert(  
    java.util.Vector<TTSCChunk>  
    chunks, Boolean playTone,  
    java.util.Vector<SoftButton>  
    softButtons, Integer correlationID  
)
```

Sends an Alert RPCRequest to SDL. Responses are captured through callback on IProxyListener.

```
public void alert(java.util.Vector<TTSCChunk> chunks, Boolean playTone,  
    java.util.Vector<SoftButton> softButtons, Integer correlationID) throws  
    SdlException
```

Parameters

chunks

-A list of text/phonemes to speak in the form of ttsChunks.

playTone

-Defines if tone should be played.

softButtons

-A list of App defined SoftButtons.

correlationID

-A unique ID that correlates each RPCRequest and RPCResponse

Throws

- SdlException


```
alert( String alertText1, String
alertText2, String alertText3,
Boolean playTone, Integer
duration,
java.util.Vector<SoftButton>
softButtons, Integer correlationID
)
```

Sends an Alert RPCRequest to SDL. Responses are captured through callback on IProxyListener.

```
public void alert(String alertText1,String alertText2,String alertText3,
Boolean playTone,Integer duration,java.util.Vector<SoftButton>
softButtons,Integer correlationID) throws SdlException
```

Parameters

alertText1

-The first line of the alert text field.

alertText2

-The second line of the alert text field.

alertText3

-The optional third line of the alert text field.

playTone

-Defines if tone should be played.

duration

-Timeout in milliseconds.

softButtons

-A list of App defined SoftButtons.

correlationID

-A unique ID that correlates each RPCRequest and RPCResponse

Throws

- SdlException

alert(String ttsText, String alertText1, String alertText2, Boolean playTone, Integer duration, Integer correlationID)

Sends an Alert RPCRequest to SDL. Responses are captured through callback on IProxyListener.

```
public void alert(String ttsText,String alertText1,String alertText2,  
Boolean playTone,Integer duration,Integer correlationID) throws  
SdlException
```

Parameters

ttsText

-The text to speech message in the form of a string.

alertText1

-The first line of the alert text field.

alertText2

-The second line of the alert text field.

playTone

-Defines if tone should be played.

duration

-Timeout in milliseconds.

correlationID

-A unique ID that correlates each RPCRequest and RPCResponse

Throws

- SdlException

alert(
java.util.Vector<TTSCChunk>
ttsChunks, String alertText1,
String alertText2, Boolean
playTone, Integer duration,
Integer correlationID)

Sends an Alert RPCRequest to SDL. Responses are captured through callback on IProxyListener.

```
public void alert(java.util.Vector<TTSCChunk> ttsChunks,String  
alertText1,String alertText2,Boolean playTone,Integer duration,Integer  
correlationID) throws SdlException
```

Parameters

ttsChunks

-A list of text/phonemes to speak in the form of ttsChunks.

alertText1

-The first line of the alert text field.

alertText2

-The second line of the alert text field.

playTone

-Defines if tone should be played.

duration

-Timeout in milliseconds.

correlationID

-A unique ID that correlates each RPCRequest and RPCResponse

Throws

- `SdlException`

`alert( String ttsText, Boolean playTone, Integer correlationID )`

Sends an Alert RPCRequest to SDL. Responses are captured through callback on `IProxyListener`.

```
public void alert(String ttsText, Boolean playTone, Integer correlationID)
throws SdlException
```

Parameters

`ttsText`

-The text to speech message in the form of a string.

`playTone`

-Defines if tone should be played.

`correlationID`

-A unique ID that correlates each RPCRequest and RPCResponse

Throws

- SdlException

alert(
java.util.Vector<TTSCChunk>
chunks, Boolean playTone,
Integer correlationID)

Sends an Alert RPCRequest to SDL. Responses are captured through callback on IProxyListener.

```
public void alert(java.util.Vector<TTSCChunk> chunks, Boolean playTone,  
Integer correlationID) throws SdlException
```

Parameters

chunks

-A list of text/phonemes to speak in the form of ttsChunks.

playTone

-Defines if tone should be played.

correlationID

-A unique ID that correlates each RPCRequest and RPCResponse

Throws

- SdlException

alert(String alertText1, String alertText2, Boolean playTone, Integer duration, Integer correlationID)

Sends an Alert RPCRequest to SDL. Responses are captured through callback on IProxyListener.

```
public void alert(String alertText1,String alertText2,Boolean playTone,
Integer duration,Integer correlationID) throws SdlException
```

Parameters

alertText1

-The first line of the alert text field.
<code>alertText2</code> -The second line of the alert text field.
<code>playTone</code> -Defines if tone should be played.
<code>duration</code> -Timeout in milliseconds.
<code>correlationID</code> -A unique ID that correlates each <code>RPCRequest</code> and <code>RPCResponse</code>

Throws

- `SdlException`

createInteractionChoiceSet(java.util.Vector<Choice> choiceSet, Integer interactionChoiceSetID, Integer correlationID)

Sends a CreateInteractionChoiceSet RPCRequest to SDL. Responses are captured through callback on IProxyListener.

```
public void createInteractionChoiceSet(java.util.Vector<Choice>  
choiceSet,Integer interactionChoiceSetID,Integer correlationID) throws  
SdlException
```

Parameters

choiceSet

to be sent to the module

interactionChoiceSetID

to be used in reference to the supplied choiceSet

correlationID

to be set to the RPCRequest

Throws

- SdlException

deleteCommand(Integer commandID, Integer correlationID)

Sends a DeleteCommand RPCRequest to SDL. Responses are captured through callback on IProxyListener.

```
public void deleteCommand(Integer commandID,Integer correlationID)  
throws SdlException
```

Parameters

commandID

-ID of the command(s) to delete.

correlationID

-A unique ID that correlates each RPCRequest and RPCResponse

Throws

- `SdlException`

`deleteInteractionChoiceSet( Integer interactionChoiceSetID, Integer correlationID )`

Sends a `DeleteInteractionChoiceSet` `RPCRequest` to SDL. Responses are captured through callback on `IProxyListener`.

```
public void deleteInteractionChoiceSet(Integer interactionChoiceSetID,  
Integer correlationID) throws SdlException
```

Parameters

`interactionChoiceSetID`

-ID of the interaction choice set to delete.

`correlationID`

-A unique ID that correlates each `RPCRequest` and `RPCResponse`

Throws

- `SdlException`

`deleteSubMenu( Integer menuID, Integer correlationID )`

Sends a DeleteSubMenu RPCRequest to SDL. Responses are captured through callback on IProxyListener.

```
public void deleteSubMenu(Integer menuID,Integer correlationID)  
throws SdlException
```

Parameters

`menuID`

-The menuID of the submenu to delete.

`correlationID`

-A unique ID that correlates each RPCRequest and RPCResponse

Throws

- `SdlException`

```
performInteraction( String  
initPrompt, String displayText,  
Integer interactionChoiceSetID,  
java.util.Vector<VrHelpItem>  
vrHelp, Integer correlationID )
```

Sends a PerformInteraction RPCRequest to SDL. Responses are captured through callback on IProxyListener.

```
public void performInteraction(String initPrompt,String displayText,  
Integer interactionChoiceSetID,java.util.Vector<VrHelpItem> vrHelp,  
Integer correlationID) throws SdlException
```

Parameters

initPrompt

-Initial prompt spoken to the user at the start of an interaction.

displayText

-Text to be displayed first.

interactionChoiceSetID

-Interaction choice set IDs to use with an interaction.

vrHelp

-Suggested VR Help Items to display on-screen during Perform Interaction.

correlationID

-A unique ID that correlates each RPCRequest and RPCResponse.

Throws

- `SdlException`

```
performInteraction( String  
initPrompt, String displayText,  
Integer interactionChoiceSetID,  
String helpPrompt, String  
timeoutPrompt, InteractionMode  
interactionMode, Integer  
timeout,  
java.util.Vector<VrHelpItem>  
vrHelp, Integer correlationID )
```

Sends a PerformInteraction RPCRequest to SDL. Responses are captured through callback on IProxyListener.

```
public void performInteraction(String initPrompt,String displayText,  
Integer interactionChoiceSetID,String helpPrompt,String timeoutPrompt,  
InteractionMode interactionMode,Integer timeout,java.util.Vector<  
VrHelpItem> vrHelp,Integer correlationID) throws SdlException
```

Parameters

initPrompt

-Initial prompt spoken to the user at the start of an interaction.

displayText

-Text to be displayed first.

interactionChoiceSetID

-Interaction choice set IDs to use with an interaction.

helpPrompt

-Help text that is spoken when a user speaks "help" during the interaction.

timeoutPrompt

-Timeout text that is spoken when a VR interaction times out.

interactionMode

- The method in which the user is notified and uses the interaction (Manual,VR,Both).

timeout

-Timeout in milliseconds.

vrHelp

-Suggested VR Help Items to display on-screen during Perform Interaction.

correlationID

-A unique ID that correlates each RPCRequest and RPCResponse.

Throws

- `SdlException`

```
performInteraction( String  
initPrompt, String displayText,  
java.util.Vector<java.lang.Integer>  
interactionChoiceSetIDList,  
String helpPrompt, String  
timeoutPrompt, InteractionMode  
interactionMode, Integer  
timeout,  
java.util.Vector<VrHelpItem>  
vrHelp, Integer correlationID )
```

Sends a PerformInteraction RPCRequest to SDL. Responses are captured through callback on IProxyListener.

```
public void performInteraction(String initPrompt,String displayText,java.  
util.Vector<java.lang.Integer> interactionChoiceSetIDList,String  
helpPrompt,String timeoutPrompt,InteractionMode interactionMode,  
Integer timeout,java.util.Vector<VrHelpItem> vrHelp,Integer  
correlationID) throws SdlException
```

Parameters

initPrompt

-Initial prompt spoken to the user at the start of an interaction.

displayText

-Text to be displayed first.

interactionChoiceSetIDList

-A list of interaction choice set IDs to use with an interaction.

helpPrompt

-Help text that is spoken when a user speaks "help" during the interaction.

timeoutPrompt

-Timeout text that is spoken when a VR interaction times out.

interactionMode

- The method in which the user is notified and uses the interaction (Manual,VR,Both).

timeout

-Timeout in milliseconds.

vrHelp

-Suggested VR Help Items to display on-screen during Perform Interaction.

correlationID

-A unique ID that correlates each RPCRequest and RPCResponse.

Throws

- `SdlException`

```
performInteraction(  
    java.util.Vector<TTSCChunk>  
    initChunks, String displayText,  
    java.util.Vector<java.lang.Integer>  
    interactionChoiceSetIDList,  
    java.util.Vector<TTSCChunk>  
    helpChunks,  
    java.util.Vector<TTSCChunk>  
    timeoutChunks, InteractionMode  
    interactionMode, Integer  
    timeout,  
    java.util.Vector<VrHelpItem>  
    vrHelp, Integer correlationID )
```

Sends a PerformInteraction RPCRequest to SDL. Responses are captured through callback on IProxyListener.

```
public void performInteraction(java.util.Vector<TTSCChunk> initChunks,  
    String displayText,java.util.Vector<java.lang.Integer>  
    interactionChoiceSetIDList,java.util.Vector<TTSCChunk> helpChunks,  
    java.util.Vector<TTSCChunk> timeoutChunks,InteractionMode  
    interactionMode,Integer timeout,java.util.Vector<VrHelpItem> vrHelp,  
    Integer correlationID) throws SdlException
```

Parameters

initChunks

-A list of text/phonemes to speak for the initial prompt in the form of ttsChunks.

displayText

-Text to be displayed first.

interactionChoiceSetIDList

-A list of interaction choice set IDs to use with an interaction.

helpChunks

-A list of text/phonemes to speak for the help text that is spoken when a user speaks "help" during the interaction.

timeoutChunks

A list of text/phonemes to speak for the timeout text that is spoken when a VR interaction times out.

interactionMode

- The method in which the user is notified and uses the interaction (Manual,VR,Both).

timeout

-Timeout in milliseconds.

vrHelp

-Suggested VR Help Items to display on-screen during Perform Interaction.

correlationID

-A unique ID that correlates each RPCRequest and RPCResponse.

Throws

- `SdlException`

performInteraction(String initPrompt, String displayText, Integer interactionChoiceSetID, Integer correlationID)

Sends a PerformInteraction RPCRequest to SDL. Responses are captured through callback on IProxyListener.

```
public void performInteraction(String initPrompt,String displayText,  
Integer interactionChoiceSetID,Integer correlationID) throws  
SdlException
```

Parameters

initPrompt

-Initial prompt spoken to the user at the start of an interaction.

displayText

-Text to be displayed first.

interactionChoiceSetID

-Interaction choice set IDs to use with an interaction.

correlationID

-A unique ID that correlates each RPCRequest and RPCResponse.

Throws

- SdlException

performInteraction(String initPrompt, String displayText, Integer interactionChoiceSetID, String helpPrompt, String timeoutPrompt, InteractionMode interactionMode, Integer timeout, Integer correlationID)

Sends a PerformInteraction RPCRequest to SDL. Responses are captured through callback on IProxyListener.

```
public void performInteraction(String initPrompt,String displayText,  
Integer interactionChoiceSetID,String helpPrompt,String timeoutPrompt,  
InteractionMode interactionMode,Integer timeout,Integer correlationID)  
throws SdlException
```

Parameters

initPrompt

-Initial prompt spoken to the user at the start of an interaction.

displayText

-Text to be displayed first.

interactionChoiceSetID

-Interaction choice set IDs to use with an interaction.

helpPrompt

-Help text that is spoken when a user speaks "help" during the interaction.

timeoutPrompt

-Timeout text that is spoken when a VR interaction times out.

interactionMode

- The method in which the user is notified and uses the interaction (Manual,VR,Both).

timeout

-Timeout in milliseconds.

correlationID

-A unique ID that correlates each RPCRequest and RPCResponse.

Throws

- SdlException

```
performInteraction( String  
initPrompt, String displayText,  
java.util.Vector<java.lang.Integer>  
interactionChoiceSetIDList,  
String helpPrompt, String  
timeoutPrompt, InteractionMode  
interactionMode, Integer  
timeout, Integer correlationID )
```

Sends a PerformInteraction RPCRequest to SDL. Responses are captured through callback on IProxyListener.

```
public void performInteraction(String initPrompt,String displayText,java.  
util.Vector<java.lang.Integer> interactionChoiceSetIDList,String  
helpPrompt,String timeoutPrompt,InteractionMode interactionMode,  
Integer timeout,Integer correlationID) throws SdlException
```

Parameters

initPrompt

-Initial prompt spoken to the user at the start of an interaction.

displayText

-Text to be displayed first.
interactionChoiceSetIDList -A list of interaction choice set IDs to use with an interaction.
helpPrompt -Help text that is spoken when a user speaks "help" during the interaction.
timeoutPrompt -Timeout text that is spoken when a VR interaction times out.
interactionMode - The method in which the user is notified and uses the interaction (Manual,VR,Both).
timeout -Timeout in milliseconds.
correlationID -A unique ID that correlates each RPCRequest and RPCResponse.

Throws

- SdlException

```
performInteraction(  
    java.util.Vector<TTSCChunk>  
    initChunks, String displayText,  
    java.util.Vector<java.lang.Integer>  
    interactionChoiceSetIDList,  
    java.util.Vector<TTSCChunk>  
    helpChunks,  
    java.util.Vector<TTSCChunk>  
    timeoutChunks, InteractionMode  
    interactionMode, Integer  
    timeout, Integer correlationID )
```

Sends a PerformInteraction RPCRequest to SDL. Responses are captured through callback on IProxyListener.

```
public void performInteraction(java.util.Vector<TTSCChunk> initChunks,  
    String displayText,java.util.Vector<java.lang.Integer>  
    interactionChoiceSetIDList,java.util.Vector<TTSCChunk> helpChunks,  
    java.util.Vector<TTSCChunk> timeoutChunks,InteractionMode  
    interactionMode,Integer timeout,Integer correlationID) throws  
    SdlException
```

Parameters

initChunks

-A list of text/phonemes to speak for the initial prompt in the form of ttsChunks.

displayText

-Text to be displayed first.

interactionChoiceSetIDList

-A list of interaction choice set IDs to use with an interaction.

helpChunks

-A list of text/phonemes to speak for the help text that is spoken when a user speaks "help" during the interaction.

timeoutChunks

A list of text/phonemes to speak for the timeout text that is spoken when a VR interaction times out.

interactionMode

- The method in which the user is notified and uses the interaction (Manual,VR,Both).

timeout

-Timeout in milliseconds.

correlationID

-A unique ID that correlates each RPCRequest and RPCResponse.

Throws

- SdlException

```
registerAppInterfacePrivate(  
    SdlMsgVersion sdlMsgVersion,  
    String appName,  
    java.util.Vector<TTSChunk>  
    ttsName, String  
    ngnMediaScreenAppName,  
    java.util.Vector<java.lang.String>  
    vrSynonyms, Boolean  
    isMediaApp, Language  
    languageDesired, Language  
    hmiDisplayLanguageDesired,  
    java.util.Vector<AppHMIType>  
    appType, String appId,  
    TemplateColorScheme  
    dayColorScheme,  
    TemplateColorScheme  
    nightColorScheme, Integer  
    correlationID )
```



```
protected void registerAppInterfacePrivate(SdIMsgVersion  
sdIMsgVersion,String appName,java.util.Vector<TTSCChunk> ttsName,  
String ngnMediaScreenAppName,java.util.Vector<java.lang.String>  
vrSynonyms,Boolean isMediaApp,Language languageDesired,Language  
hmiDisplayLanguageDesired,java.util.Vector<AppHMIType> appType,  
String appID,TemplateColorScheme dayColorScheme,  
TemplateColorScheme nightColorScheme,Integer correlationID) throws  
SdIException
```

Parameters

sdIMsgVersion
appName
ttsName
ngnMediaScreenAppName
vrSynonyms
isMediaApp

languageDesired
hmiDisplayLanguageDesired
appType
appID
dayColorScheme
nightColorScheme
correlationID

Throws

- `SdlException`

setGlobalProperties(String helpPrompt, String timeoutPrompt, String vrHelpTitle, java.util.Vector<VrHelpItem> vrHelp, Integer correlationID)

Sends a SetGlobalProperties RPCRequest to SDL. Responses are captured through callback on IProxyListener.

```
public void setGlobalProperties(String helpPrompt,String timeoutPrompt  
,String vrHelpTitle,java.util.Vector<VrHelpItem> vrHelp,Integer  
correlationID) throws SdlException
```

Parameters

helpPrompt

that will be used for the VR screen

timeoutPrompt

string to be displayed after timeout

vrHelpTitle

string that may be displayed on VR prompt dialog

vrHelp

a list of VR synonyms that may be displayed to user

correlationID

to be attached to the request

Throws

- `SdlException`

```
setGlobalProperties(  
    java.util.Vector<TTSCChunk>  
    helpChunks,  
    java.util.Vector<TTSCChunk>  
    timeoutChunks, String  
    vrHelpTitle,  
    java.util.Vector<VrHelpItem>  
    vrHelp, Integer correlationID )
```

Sends a SetGlobalProperties RPCRequest to SDL. Responses are captured through callback on IProxyListener.

```
public void setGlobalProperties(java.util.Vector<TTSCChunk>  
    helpChunks,java.util.Vector<TTSCChunk> timeoutChunks,String  
    vrHelpTitle,java.util.Vector<VrHelpItem> vrHelp,Integer correlationID)  
    throws SdlException
```

Parameters

helpChunks

tts chunks that should be used when prompting the user

timeoutChunks

tts chunks that will be used when a timeout occurs
vrHelpTitle string that may be displayed on VR prompt dialog
vrHelp a list of VR synonyms that may be displayed to user
correlationID ID to be attached to the RPCRequest that correlates the RPCResponse

Throws

- `SdlException`

`setGlobalProperties( String helpPrompt, String timeoutPrompt, Integer correlationID )`

Sends a `SetGlobalProperties` `RPCRequest` to SDL. Responses are captured through callback on `IProxyListener`.

```
public void setGlobalProperties(String helpPrompt,String timeoutPrompt  
,Integer correlationID) throws SdlException
```

Parameters

helpPrompt

that will be used for the VR screen

timeoutPrompt

string to be displayed after timeout

correlationID

ID to be attached to the RPCRequest that correlates the RPCResponse

Throws

- `SdlException`

```
setGlobalProperties(  
    java.util.Vector<TTSCChunk>  
    helpChunks,  
    java.util.Vector<TTSCChunk>  
    timeoutChunks, Integer  
    correlationID )
```

Sends a SetGlobalProperties RPCRequest to SDL. Responses are captured through callback on IProxyListener.

```
public void setGlobalProperties(java.util.Vector<TTSCChunk>  
    helpChunks,java.util.Vector<TTSCChunk> timeoutChunks,Integer  
    correlationID) throws SdlException
```

Parameters

helpChunks

tts chunks that should be used when prompting the user

timeoutChunks

tts chunks that will be used when a timeout occurs

correlationID

ID to be attached to the RPCRequest that correlates the RPCResponse

Throws

- SdlException

```
resetGlobalProperties(  
    java.util.Vector<GlobalProperty>  
    properties, Integer correlationID  
)
```

```
public void resetGlobalProperties(java.util.Vector<GlobalProperty>  
    properties,Integer correlationID) throws SdlException
```

Parameters

properties

correlationID

Throws

- `SdlException`

`setMediaClockTimer( Integer hours, Integer minutes, Integer seconds, UpdateMode updateMode, Integer correlationID )`

Sends a `SetMediaClockTimer` `RPCRequest` to `SDL`. Responses are captured through callback on `IProxyListener`.

```
public void setMediaClockTimer(Integer hours,Integer minutes,Integer seconds,UpdateMode updateMode,Integer correlationID) throws SdlException
```

Parameters

hours

integer for hours

minutes

integer for minutes

seconds

integer for seconds

updateMode

mode in which the media clock timer should be updated

correlationID

ID to be attached to the RPCRequest that correlates the RPCResponse

Throws

- `SdlException`

pauseMediaClockTimer(Integer correlationID)

Pauses the media clock. Responses are captured through callback on `IProxyListener`.

```
public void pauseMediaClockTimer(Integer correlationID) throws SdlException
```

Parameters

correlationID

ID to be attached to the RPCRequest that correlates the RPCResponse

Throws

- SdlException

resumeMediaClockTimer(Integer correlationID)

Resumes the media clock. Responses are captured through callback on IProxyListener.

```
public void resumeMediaClockTimer(Integer correlationID) throws  
SdlException
```

Parameters

correlationID

ID to be attached to the RPCRequest that correlates the RPCResponse

Throws

- `SdlException`

`clearMediaClockTimer( Integer correlationID )`

Clears the media clock. Responses are captured through callback on `IProxyListener`.

```
public void clearMediaClockTimer(Integer correlationID) throws SdlException
```

Parameters

`correlationID`

ID to be attached to the RPCRequest that correlates the RPCResponse

Throws

- `SdlException`

```
show( String mainText1, String  
mainText2, String mainText3,  
String mainText4, String  
statusBar, String mediaClock,  
String mediaTrack, Image  
graphic,  
java.util.Vector<SoftButton>  
softButtons,  
java.util.Vector<java.lang.String>  
customPresets, TextAlignment  
alignment, Integer correlationID )
```

Sends a Show RPCRequest to SDL. Responses are captured through callback on `IProxyListener`.

```
public void show(String mainText1,String mainText2,String mainText3,  
String mainText4,String statusBar,String mediaClock,String mediaTrack,  
Image graphic,java.util.Vector<SoftButton> softButtons,java.util.Vector  
<java.lang.String> customPresets,TextAlignment alignment,Integer  
correlationID) throws SdlException
```

Parameters

mainText1

text displayed in a single or upper display line.

mainText2

text displayed on the second display line.

mainText3

text displayed on the second "page" first display line.

mainText4

text displayed on the second "page" second display line.

statusBar

text is placed in the status bar area (Only valid for NAVIGATION apps)

mediaClock

text value for MediaClock field.

mediaTrack

text displayed in the track field.

graphic

image struct determining whether static or dynamic image to display in app.

softButtons

app defined SoftButtons.

customPresets

app labeled on-screen presets.

alignment

specifies how mainText1 and mainText2s texts should be aligned on display.

correlationID

ID to be attached to the RPCRequest that correlates the RPCResponse -
A unique ID that correlates each RPCRequest and RPCResponse.

Throws

- SdlException

`show(String mainText1, String
mainText2, String mainText3,
String mainText4, Image graphic,
java.util.Vector<SoftButton>
softButtons,
java.util.Vector<java.lang.String>
customPresets, TextAlignment
alignment, Integer correlationID)`

Sends a Show RPCRequest to SDL. Responses are captured through callback on
IProxyListener.

```
public void show(String mainText1,String mainText2,String mainText3,  
String mainText4,Image graphic,java.util.Vector<SoftButton>  
softButtons,java.util.Vector<java.lang.String> customPresets,  
TextAlignment alignment,Integer correlationID) throws SdlException
```

Parameters

mainText1

-Text displayed in a single or upper display line.

mainText2

-Text displayed on the second display line.

mainText3

-Text displayed on the second "page" first display line.

mainText4

-Text displayed on the second "page" second display line.

graphic

-Image struct determining whether static or dynamic image to display in app.

softButtons

-App defined SoftButtons.

customPresets

-App labeled on-screen presets.

alignment

-Specifies how mainText1 and mainText2s texts should be aligned on display.

correlationID

-A unique ID that correlates each RPCRequest and RPCResponse.

Throws

- SdlException

show(String mainText1, String mainText2, String statusBar, String mediaClock, String mediaTrack, TextAlignment alignment, Integer correlationID)

Sends a Show RPCRequest to SDL. Responses are captured through callback on IProxyListener.

```
public void show(String mainText1,String mainText2,String statusBar,  
String mediaClock,String mediaTrack,TextAlignment alignment,Integer  
correlationID) throws SdlException
```

Parameters

mainText1

text displayed in a single or upper display line.

mainText2

text displayed on the second display line.

statusBar

text is placed in the status bar area (Only valid for NAVIGATION apps)

mediaClock

text value for MediaClock field.

mediaTrack

text displayed in the track field.

alignment

specifies how mainText1 and mainText2s texts should be aligned on display.

correlationID

unique ID that correlates each RPCRequest and RPCResponse.

Throws

- `SdlException`

`show( String mainText1, String mainText2, TextAlignment alignment, Integer correlationID )`

Sends a Show RPCRequest to SDL. Responses are captured through callback on `IProxyListener`.

```
public void show(String mainText1,String mainText2,TextAlignment alignment,Integer correlationID) throws SdlException
```

Parameters

`mainText1`

-Text displayed in a single or upper display line.

`mainText2`

-Text displayed on the second display line.

alignment

-Specifies how mainText1 and mainText2s texts should be aligned on display.

correlationID

-A unique ID that correlates each RPCRequest and RPCResponse.

Throws

- SdlException

speak(String ttsText, Integer correlationID)

Sends a Speak RPCRequest to SDL. Responses are captured through callback on IProxyListener.

```
public void speak(String ttsText,Integer correlationID) throws  
SdlException
```

Parameters

ttsText

-The text to speech message in the form of a string.

correlationID

-A unique ID that correlates each RPCRequest and RPCResponse.

Throws

- SdlException

Speak(
java.util.Vector<TTSText>
ttsChunks, Integer correlationID)

Sends a Speak RPCRequest to SDL. Responses are captured through callback on IProxyListener.

```
public void speak(java.util.Vector<TTSText> ttsChunks,Integer correlationID) throws SdlException
```

Parameters

ttsChunks

-Text/phonemes to speak in the form of ttsChunks.

correlationID

-A unique ID that correlates each RPCRequest and RPCResponse.

Throws

- SdlException

subscribeButton(ButtonName buttonName, Integer correlationID)

Sends a SubscribeButton RPCRequest to SDL. Responses are captured through callback on IProxyListener.

```
public void subscribeButton(ButtonName buttonName,Integer  
correlationID) throws SdlException
```


Parameters

buttonName

-Name of the button to subscribe.

correlationID

-A unique ID that correlates each RPCRequest and RPCResponse.

Throws

- SdlException

unregisterAppInterfacePrivate(Integer correlationID)

```
protected void unregisterAppInterfacePrivate(Integer correlationID)  
throws SdlException
```

Parameters

correlationID

Throws

- SdlException

unsubscribeButton(ButtonName buttonName, Integer correlationID)

Sends an UnsubscribeButton RPCRequest to SDL. Responses are captured through callback on IProxyListener.

```
public void unsubscribeButton(ButtonName buttonName,Integer  
correlationID) throws SdlException
```

Parameters

buttonName

-Name of the button to unsubscribe.

correlationID

-A unique ID that correlates each RPCRequest and RPCResponse.

Throws

- SdlException

```
createChoiceSetChoice( Integer  
choiceID, String  
choiceMenuName,  
java.util.Vector<java.lang.String>  
choiceVrCommands )
```

Creates a choice to be added to a choiceset. Choice has both a voice and a visual menu component.

```
public Choice createChoiceSetChoice(Integer choiceID,String  
choiceMenuName,java.util.Vector<java.lang.String>  
choiceVrCommands)
```

Parameters

choiceID

-Unique ID used to identify this choice (returned in callback).

choiceMenuName

-Text name displayed for this choice.

choiceVrCommands

-Vector of vrCommands used to select this choice by voice. Must contain at least one non-empty element.

```
performaudiopassthru( String  
initialPrompt, String  
audioPassThruDisplayText1,  
String  
audioPassThruDisplayText2,  
SamplingRate samplingRate,  
Integer maxDuration,  
BitsPerSample bitsPerSample,  
AudioType audioType, Boolean  
muteAudio, Integer correlationID  
)
```

Starts audio pass thru session. Responses are captured through callback on IProxyListener.

```
public void performaudiopassthru(String initialPrompt,String  
audioPassThruDisplayText1,String audioPassThruDisplayText2,  
SamplingRate samplingRate,Integer maxDuration,BitsPerSample  
bitsPerSample,AudioType audioType,Boolean muteAudio,Integer  
correlationID) throws SdlException
```

Parameters

initialPrompt

-SDL will speak this prompt before opening the audio pass thru session.

audioPassThruDisplayText1

-First line of text displayed during audio capture.

audioPassThruDisplayText2

-Second line of text displayed during audio capture.

samplingRate

-Allowable values of 8 khz or 16 or 22 or 44 khz.

maxDuration

-The maximum duration of audio recording in milliseconds.

bitsPerSample

-Specifies the quality the audio is recorded. Currently 8 bit or 16 bit.

audioType

-Specifies the type of audio data being requested.

`muteAudio`

-Defines if the current audio source should be muted during the APT session.

`correlationID`

-A unique ID that correlates each `RPCRequest` and `RPCResponse`.

Throws

- `SdlException`

`endaudiopassthru( Integer correlationID )`

Ends audio pass thru session. Responses are captured through callback on `IProxyListener`.

```
public void endaudiopassthru(Integer correlationID) throws  
SdlException
```

Parameters

correlationID

ID to be attached to the RPCRequest that correlates the RPCResponse

Throws

- SdlException

subscribevehicledata(boolean
gps, boolean speed, boolean
rpm, boolean fuelLevel, boolean
fuelLevel_State, boolean
instantFuelConsumption, boolean
externalTemperature, boolean
prndl, boolean tirePressure,
boolean odometer, boolean
beltStatus, boolean
bodyInformation, boolean
deviceStatus, boolean
driverBraking, Integer
correlationID)

Deprecated

Subscribes for specific published data items. The data will be only sent if it has changed. Responses are captured through callback on IProxyListener.

```
public void subscribevehicledata(boolean gps,boolean speed,boolean  
rpm,boolean fuelLevel,boolean fuelLevel_State,boolean  
instantFuelConsumption,boolean externalTemperature,boolean prndl,  
boolean tirePressure,boolean odometer,boolean beltStatus,boolean  
bodyInformation,boolean deviceStatus,boolean driverBraking,Integer  
correlationID) throws SdlException
```

Parameters

gps

-Subscribes to GPS data.

speed

-Subscribes to vehicle speed data in kilometers per hour.

rpm

-Subscribes to number of revolutions per minute of the engine.

fuelLevel

-Subscribes to fuel level in the tank (percentage).

fuelLevel_State

-Subscribes to fuel level state.

instantFuelConsumption

-Subscribes to instantaneous fuel consumption in microlitres.

externalTemperature

-Subscribes to the external temperature in degrees celsius.

prndl

-Subscribes to PRNDL data that houses the selected gear.

tirePressure

-Subscribes to the TireStatus data containing status and pressure of tires.

odometer

-Subscribes to Odometer data in km.

beltStatus

-Subscribes to status of the seat belts.

bodyInformation

-Subscribes to body information including power modes.

deviceStatus

-Subscribes to device status including signal and battery strength.

driverBraking

-Subscribes to the status of the brake pedal.

correlationID

-A unique ID that correlates each RPCRequest and RPCResponse.

Throws

- SdlException

```
subscribevehicledata( boolean  
gps, boolean speed, boolean  
rpm, boolean fuelLevel, boolean  
fuelLevel_State, boolean  
instantFuelConsumption, boolean  
externalTemperature, boolean  
prndl, boolean tirePressure,  
boolean engineOilLife, boolean  
odometer, boolean beltStatus,  
boolean bodyInformation,  
boolean deviceStatus, boolean  
driverBraking, Integer  
correlationID )
```

Subscribes for specific published data items. The data will be only sent if it has changed. Responses are captured through callback on IProxyListener.

```
public void subscribevehicledata(boolean gps,boolean speed,boolean rpm,boolean fuelLevel,boolean fuelLevel_State,boolean instantFuelConsumption,boolean externalTemperature,boolean prndl,boolean tirePressure,boolean engineOilLife,boolean odometer,boolean beltStatus,boolean bodyInformation,boolean deviceStatus,boolean driverBraking,Integer correlationID) throws SdlException
```

Parameters

gps

-Subscribes to GPS data.

speed

-Subscribes to vehicle speed data in kilometers per hour.

rpm

-Subscribes to number of revolutions per minute of the engine.

fuelLevel

-Subscribes to fuel level in the tank (percentage).

fuelLevel_State

-Subscribes to fuel level state.

instantFuelConsumption

-Subscribes to instantaneous fuel consumption in microlitres.

externalTemperature

-Subscribes to the external temperature in degrees celsius.

prndl

-Subscribes to PRNDL data that houses the selected gear.

tirePressure

-Subscribes to the TireStatus data containing status and pressure of tires.

engineOilLife

-Subscribes to Engine Oil Life data.

odometer

-Subscribes to Odometer data in km.

beltStatus

-Subscribes to status of the seat belts.

bodyInformation

-Subscribes to body information including power modes.

deviceStatus

-Subscribes to device status including signal and battery strength.

driverBraking

-Subscribes to the status of the brake pedal.

correlationID

-A unique ID that correlates each RPCRequest and RPCResponse.

Throws

- SdlException

```
unsubscribevehicledata(  
boolean gps, boolean speed,  
boolean rpm, boolean fuelLevel,  
boolean fuelLevel_State, boolean  
instantFuelConsumption, boolean  
externalTemperature, boolean  
prndl, boolean tirePressure,  
boolean odometer, boolean  
beltStatus, boolean  
bodyInformation, boolean  
deviceStatus, boolean  
driverBraking, Integer  
correlationID )
```

Deprecated

Unsubscribes for specific published data items. Responses are captured through callback on IProxyListener.

```
public void unsubscribevehicledata(boolean gps,boolean speed,boolean  
rpm,boolean fuelLevel,boolean fuelLevel_State,boolean  
instantFuelConsumption,boolean externalTemperature,boolean prndl,  
boolean tirePressure,boolean odometer,boolean beltStatus,boolean  
bodyInformation,boolean deviceStatus,boolean driverBraking,Integer  
correlationID) throws SdlException
```


Parameters

gps

-Unsubscribes to GPS data.

speed

-Unsubscribes to vehicle speed data in kilometers per hour.

rpm

-Unsubscribes to number of revolutions per minute of the engine.

fuelLevel

-Unsubscribes to fuel level in the tank (percentage).

fuelLevel_State

-Unsubscribes to fuel level state.

instantFuelConsumption

-Unsubscribes to instantaneous fuel consumption in microlitres.

externalTemperature

-Unsubscribes to the external temperature in degrees celsius.

prndl

-Unsubscribes to PRNDL data that houses the selected gear.

tirePressure

-Unsubscribes to the TireStatus data containing status and pressure of tires.

odometer

-Unsubscribes to Odometer data in km.

beltStatus

-Unsubscribes to status of the seat belts.

bodyInformation

-Unsubscribes to body information including power modes.

deviceStatus

-Unsubscribes to device status including signal and battery strength.

driverBraking

-Unsubscribes to the status of the brake pedal.

correlationID

-A unique ID that correlates each RPCRequest and RPCResponse.

Throws

- SdlException

```
unsubscribevehicledata(  
boolean gps, boolean speed,  
boolean rpm, boolean fuelLevel,  
boolean fuelLevel_State, boolean  
instantFuelConsumption, boolean  
externalTemperature, boolean  
prndl, boolean tirePressure,  
boolean engineOilLife, boolean  
odometer, boolean beltStatus,  
boolean bodyInformation,  
boolean deviceStatus, boolean  
driverBraking, Integer  
correlationID )
```

Unsubscribes for specific published data items. Responses are captured through callback on IProxyListener.

```
public void unsubscribevehicledata(boolean gps,boolean speed,boolean rpm,boolean fuelLevel,boolean fuelLevel_State,boolean instantFuelConsumption,boolean externalTemperature,boolean prndl,boolean tirePressure,boolean engineOilLife,boolean odometer,boolean beltStatus,boolean bodyInformation,boolean deviceStatus,boolean driverBraking,Integer correlationID) throws SdlException
```

Parameters

gps

-Unsubscribes to GPS data.

speed

-Unsubscribes to vehicle speed data in kilometers per hour.

rpm

-Unsubscribes to number of revolutions per minute of the engine.

fuelLevel

-Unsubscribes to fuel level in the tank (percentage).

fuelLevel_State

-Unsubscribes to fuel level state.

instantFuelConsumption

-Unsubscribes to instantaneous fuel consumption in microlitres.

externalTemperature

-Unsubscribes to the external temperature in degrees celsius.

prndl

-Unsubscribes to PRNDL data that houses the selected gear.

tirePressure

-Unsubscribes to the TireStatus data containing status and pressure of tires.

engineOilLife

-Unsubscribes to Engine Oil Life data.

odometer

-Unsubscribes to Odometer data in km.

beltStatus

-Unsubscribes to status of the seat belts.

bodyInformation

-Unsubscribes to body information including power modes.

deviceStatus

-Unsubscribes to device status including signal and battery strength.

driverBraking

-Unsubscribes to the status of the brake pedal.

correlationID

-A unique ID that correlates each RPCRequest and RPCResponse.

Throws

- SdlException

getvehicledata(boolean gps,
boolean speed, boolean rpm,
boolean fuelLevel, boolean
fuelLevel_State, boolean
instantFuelConsumption, boolean
externalTemperature, boolean
vin, boolean prndl, boolean
tirePressure, boolean odometer,
boolean beltStatus, boolean
bodyInformation, boolean
deviceStatus, boolean
driverBraking, Integer
correlationID)

Deprecated

Performs a Non periodic vehicle data read request. Responses are captured through callback on IProxyListener.

```
public void getvehicledata(boolean gps,boolean speed,boolean rpm,  
boolean fuelLevel,boolean fuelLevel_State,boolean  
instantFuelConsumption,boolean externalTemperature,boolean vin,  
boolean prndl,boolean tirePressure,boolean odometer,boolean  
beltStatus,boolean bodyInformation,boolean deviceStatus,boolean  
driverBraking,Integer correlationID) throws SdlException
```

Parameters

gps

-Performs an ad-hoc request for GPS data.

speed

-Performs an ad-hoc request for vehicle speed data in kilometers per hour.

rpm

-Performs an ad-hoc request for number of revolutions per minute of the engine.

fuelLevel

-Performs an ad-hoc request for fuel level in the tank (percentage).

fuelLevel_State

-Performs an ad-hoc request for fuel level state.

instantFuelConsumption

-Performs an ad-hoc request for instantaneous fuel consumption in microlitres.

externalTemperature

-Performs an ad-hoc request for the external temperature in degrees celsius.

vin

-Performs an ad-hoc request for the Vehicle identification number

prndl

-Performs an ad-hoc request for PRNDL data that houses the selected gear.

tirePressure

-Performs an ad-hoc request for the TireStatus data containing status and pressure of tires.

odometer

-Performs an ad-hoc request for Odometer data in km.

beltStatus

-Performs an ad-hoc request for status of the seat belts.

bodyInformation

-Performs an ad-hoc request for body information including power modes.

deviceStatus

-Performs an ad-hoc request for device status including signal and battery strength.

driverBraking

-Performs an ad-hoc request for the status of the brake pedal.

correlationID

-A unique ID that correlates each RPCRequest and RPCResponse.

Throws

- `SdlException`

getvehicledata(boolean gps,
boolean speed, boolean rpm,
boolean fuelLevel, boolean
fuelLevel_State, boolean
instantFuelConsumption, boolean
externalTemperature, boolean
vin, boolean prndl, boolean
tirePressure, boolean
engineOilLife, boolean odometer,
boolean beltStatus, boolean
bodyInformation, boolean
deviceStatus, boolean
driverBraking, Integer
correlationID)

Performs a Non periodic vehicle data read request. Responses are captured through callback on IProxyListener.

```
public void getvehicledata(boolean gps,boolean speed,boolean rpm,  
boolean fuelLevel,boolean fuelLevel_State,boolean  
instantFuelConsumption,boolean externalTemperature,boolean vin,  
boolean prndl,boolean tirePressure,boolean engineOilLife,boolean  
odometer,boolean beltStatus,boolean bodyInformation,boolean  
deviceStatus,boolean driverBraking,Integer correlationID) throws  
SdlException
```

Parameters

gps

-Performs an ad-hoc request for GPS data.

speed

-Performs an ad-hoc request for vehicle speed data in kilometers per hour.

rpm

-Performs an ad-hoc request for number of revolutions per minute of the engine.

fuelLevel

-Performs an ad-hoc request for fuel level in the tank (percentage).

fuelLevel_State

-Performs an ad-hoc request for fuel level state.

instantFuelConsumption

-Performs an ad-hoc request for instantaneous fuel consumption in microlitres.

externalTemperature

-Performs an ad-hoc request for the external temperature in degrees celsius.

vin

-Performs an ad-hoc request for the Vehicle identification number

prndl

-Performs an ad-hoc request for PRNDL data that houses the selected gear.

tirePressure

-Performs an ad-hoc request for the TireStatus data containing status and pressure of tires.

engineOilLife

-Performs an ad-hoc request for Engine Oil Life data.

odometer

-Performs an ad-hoc request for Odometer data in km.

beltStatus

-Performs an ad-hoc request for status of the seat belts.

bodyInformation

-Performs an ad-hoc request for body information including power modes.

deviceStatus

-Performs an ad-hoc request for device status including signal and battery strength.

driverBraking

-Performs an ad-hoc request for the status of the brake pedal.

correlationID

-A unique ID that correlates each RPCRequest and RPCResponse.

Throws

- SdlException

```
scrollablemessage( String  
scrollableMessageBody, Integer  
timeout,  
java.util.Vector<SoftButton>  
softButtons, Integer correlationID  
)
```

Creates a full screen overlay containing a large block of formatted text that can be scrolled with up to 8 SoftButtons defined. Responses are captured through callback on IProxyListener.

```
public void scrollablemessage(String scrollableMessageBody,Integer
timeout,java.util.Vector<SoftButton> softButtons,Integer correlationID)
throws SdlException
```

Parameters

scrollableMessageBody

-Body of text that can include newlines and tabs.

timeout

-App defined timeout. Indicates how long of a timeout from the last action.

softButtons

-App defined SoftButtons.

correlationID

-A unique ID that correlates each RPCRequest and RPCResponse.

Throws

- SdlException

slider(Integer numTicks, Integer position, String sliderHeader, java.util.Vector<java.lang.String> sliderFooter, Integer timeout, Integer correlationID)

Creates a full screen or pop-up overlay (depending on platform) with a single user controlled slider. Responses are captured through callback on IProxyListener.

```
public void slider(Integer numTicks,Integer position,String sliderHeader,  
java.util.Vector<java.lang.String> sliderFooter,Integer timeout,Integer  
correlationID) throws SdlException
```

Parameters

numTicks

-Number of selectable items on a horizontal axis.

position

-Initial position of slider control (cannot exceed numTicks).

sliderHeader

-Text header to display.
sliderFooter - Text footer to display (meant to display min/max threshold descriptors).
timeout -App defined timeout. Indicates how long of a timeout from the last action.
correlationID -A unique ID that correlates each RPCRequest and RPCResponse.

Throws

- SdlException

changeregistration(Language
language, Language
hmiDisplayLanguage, Integer
correlationID)

Responses are captured through callback on IProxyListener.

```
public void changeregistration(Language language, Language  
hmiDisplayLanguage, Integer correlationID) throws SdlException
```

Parameters

language

requested SDL voice engine (VR+TTS) language registration

hmiDisplayLanguage

request display language registration.

correlationID

ID to be attached to the RPCRequest that correlates the RPCResponse

Throws

- SdlException

putFileStream(InputStream is, String sdlFileName, Integer iOffset, Integer iLength)

Deprecated

Used to push a binary stream of file data onto the module from a mobile device. Responses are captured through callback on IProxyListener.

```
public void putFileStream(InputStream is,String sdlFileName,Integer iOffset,Integer iLength) throws SdlException
```

Parameters

is

- The input stream of byte data that putFileStream will read from

sdlFileName

- The file reference name used by the putFile RPC.

iOffset

- The data offset in bytes, a value of zero is used to indicate data starting from the beginning of the file. A value greater than zero is used for resuming partial data chunks.

iLength

- The total length of the file being sent.

Throws

- SdlException

putFileStream(InputStream inputStream, String fileName, Long offset, Long length)

Used to push a binary stream of file data onto the module from a mobile device. Responses are captured through callback on IProxyListener.

```
public void putFileStream(InputStream inputStream,String fileName,  
Long offset,Long length) throws SdlException
```

Parameters

inputStream

The input stream of byte data that will be read from.

fileName

The SDL file reference name used by the RPC.

offset

The data offset in bytes. A value of zero is used to indicate data starting from the beginning of the file and a value greater than zero is used for resuming partial data chunks.

length

The total length of the file being sent.

Throws

- `SdlException`

putFileStream(String sdlFileName, Integer iOffset, Integer iLength)

Deprecated

Used to push a binary stream of file data onto the module from a mobile device. Responses are captured through callback on `IProxyListener`.

```
public java.io.OutputStream putFileStream(String sdlFileName,Integer  
iOffset,Integer iLength) throws SdlException
```

Parameters

sdIFileName

- The file reference name used by the putFile RPC.

iOffset

- The data offset in bytes, a value of zero is used to indicate data starting from the beginning of a file. A value greater than zero is used for resuming partial data chunks.

iLength

- The total length of the file being sent.

Throws

- SdlException

putFileStream(String fileName, Long offset, Long length)

Used to push a binary stream of file data onto the module from a mobile device. Responses are captured through callback on IProxyListener.

```
public java.io.OutputStream putFileStream(String fileName, Long offset,  
Long length) throws SdlException
```

Parameters

fileName

The SDL file reference name used by the RPC.

offset

The data offset in bytes. A value of zero is used to indicate data starting from the beginning of the file and a value greater than zero is used for resuming partial data chunks.

length

The total length of the file being sent.

Throws

- SdlException

putFileStream(InputStream is, String sdFileName, Integer iOffset, Integer iLength, FileType fileType, Boolean bPersistentFile, Boolean bSystemFile)

Deprecated

Used to push a binary stream of file data onto the module from a mobile device. Responses are captured through callback on IProxyListener.

```
public void putFileStream(InputStream is,String sdFileName,Integer iOffset,Integer iLength,FileType fileType,Boolean bPersistentFile,Boolean bSystemFile) throws SdException
```

Parameters

is

- The input stream of byte data that PutFileStream will read from

sdFileName

- The file reference name used by the putFile RPC.

iOffset

- The data offset in bytes, a value of zero is used to indicate data starting from the beginning of the file. A value greater than zero is used for resuming partial data chunks.

iLength

- The total length of the file being sent.

fileType

- The selected file type -- see the FileType enumeration for details

bPersistentFile

- Indicates if the file is meant to persist between sessions / ignition cycles.

bSystemFile

- Indicates if the file is meant to be passed thru core to elsewhere on the system.

Throws

- SdlException

putFileStream(InputStream inputStream, String fileName, Long offset, Long length, FileType fileType, Boolean isPersistentFile, Boolean isSystemFile, OnPutFileUpdateListener cb)

Used to push a binary stream of file data onto the module from a mobile device. Responses are captured through callback on IProxyListener.

```
public void putFileStream(InputStream inputStream,String fileName,  
Long offset,Long length,FileType fileType,Boolean isPersistentFile,  
Boolean isSystemFile,OnPutFileUpdateListener cb) throws SdlException
```

Parameters

inputStream

The input stream of byte data that will be read from.

fileName

The SDL file reference name used by the RPC.

offset

The data offset in bytes. A value of zero is used to indicate data starting from the beginning of the file and a value greater than zero is used for resuming partial data chunks.

length

The total length of the file being sent.

fileType

The selected file type. See the {@link FileType} enum for details.

isPersistentFile

Indicates if the file is meant to persist between sessions / ignition cycles.

isSystemFile

Indicates if the file is meant to be passed through core to elsewhere in the system.

cb

Throws

- `SdlException`

putFileStream(String sdlFileName, Integer iOffset, Integer iLength, FileType fileType, Boolean bPersistentFile, Boolean bSystemFile)

Deprecated

Used to push a binary stream of file data onto the module from a mobile device. Responses are captured through callback on IProxyListener.

```
public java.io.OutputStream putFileStream(String sdlFileName,Integer  
iOffset,Integer iLength,FileType fileType,Boolean bPersistentFile,Boolean  
bSystemFile) throws SdlException
```

Parameters

sdlFileName

- The file reference name used by the putFile RPC.

iOffset

- The data offset in bytes, a value of zero is used to indicate data starting from the beginning of a file. A value greater than zero is used for resuming partial data chunks.

iLength

- The total length of the file being sent.

fileType

- The selected file type -- see the FileType enumeration for details

bPersistentFile

- Indicates if the file is meant to persist between sessions / ignition cycles.

bSystemFile

- Indicates if the file is meant to be passed thru core to elsewhere on the system.

Throws

- SdlException

putFileStream(String fileName, Long offset, Long length, FileType fileType, Boolean isPersistentFile, Boolean isSystemFile, OnPutFileUpdateListener cb)

Used to push a binary stream of file data onto the module from a mobile device. Responses are captured through callback on IProxyListener.

```
public java.io.OutputStream putFileStream(String fileName, Long offset, Long length, FileType fileType, Boolean isPersistentFile, Boolean isSystemFile, OnPutFileUpdateListener cb) throws SdlException
```

Parameters

fileName

The SDL file reference name used by the RPC.

offset

The data offset in bytes. A value of zero is used to indicate data starting from the beginning of the file and a value greater than zero is used for resuming partial data chunks.

length

The total length of the file being sent.

fileType

The selected file type. See the {@link FileType} enum for details.

isPersistentFile

Indicates if the file is meant to persist between sessions / ignition cycles.

isSystemFile

Indicates if the file is meant to be passed through core to elsewhere in the system.

cb

Throws

- `SdlException`

putFileStream(String sPath, String sdlFileName, Integer iOffset, FileType fileType, Boolean bPersistentFile, Boolean bSystemFile, Integer iCorrelationID)

Deprecated

Used to push a stream of putfile RPC's containing binary data from a mobile device to the module. Responses are captured through callback on IProxyListener.

```
public com.smartdevicelink.proxy.RPCStreamController putFileStream(  
String sPath,String sdlFileName,Integer iOffset,FileType fileType,Boolean  
bPersistentFile,Boolean bSystemFile,Integer iCorrelationID) throws  
SdlException
```

Parameters

sPath

- The physical file path on the mobile device.

sdlFileName

- The file reference name used by the putFile RPC.

iOffset

- The data offset in bytes, a value of zero is used to indicate data starting from the beginning of a file. A value greater than zero is used for resuming partial data chunks.

fileType

- The selected file type -- see the FileType enumeration for details

bPersistentFile

- Indicates if the file is meant to persist between sessions / ignition cycles.

bSystemFile

- Indicates if the file is meant to be passed thru core to elsewhere on the system.

iCorrelationID

- A unique ID that correlates each RPCRequest and RPCResponse.

Throws

- SdlException

```
putFileStream( String path,  
String fileName, Long offset,  
FileType fileType, Boolean  
isPersistentFile, Boolean  
isSystemFile, Boolean  
isPayloadProtected, Integer  
correlationId,  
OnPutFileUpdateListener cb )
```

Used to push a binary stream of file data onto the module from a mobile device. Responses are captured through callback on IProxyListener.

```
public com.smartdevicelink.proxy.RPCStreamController putFileStream(  
String path,String fileName,Long offset,FileType fileType,Boolean  
isPersistentFile,Boolean isSystemFile,Boolean isPayloadProtected,  
Integer correlationId,OnPutFileUpdateListener cb) throws SdlException
```

Parameters

path

The physical file path on the mobile device.

fileName

The SDL file reference name used by the RPC.
offset The data offset in bytes. A value of zero is used to indicate data starting from the beginning of the file and a value greater than zero is used for resuming partial data chunks.
fileType The selected file type. See the {@link FileType} enum for details.
isPersistentFile Indicates if the file is meant to persist between sessions / ignition cycles.
isSystemFile Indicates if the file is meant to be passed through core to elsewhere in the system.
isPayloadProtected
correlationId A unique id that correlates each RPCRequest and RPCResponse.
cb

Throws

- SdlException

putFileStream(InputStream is, String sdlFileName, Integer iOffset, Integer iLength, FileType fileType, Boolean bPersistentFile, Boolean bSystemFile, Integer iCorrelationID)

Deprecated

Used to push a stream of putfile RPC's containing binary data from a mobile device to the module. Responses are captured through callback on IProxyListener.

```
public com.smartdevicelink.proxy.RPCStreamController putFileStream(
    InputStream is,String sdlFileName,Integer iOffset,Integer iLength,
    FileType fileType,Boolean bPersistentFile,Boolean bSystemFile,Integer
    iCorrelationID) throws SdlException
```

Parameters

is

- The input stream of byte data that putFileStream will read from.

sdIFileName

- The file reference name used by the putFile RPC.

iOffset

- The data offset in bytes, a value of zero is used to indicate data starting from the beginning of a file. A value greater than zero is used for resuming partial data chunks.

iLength

fileType

- The selected file type -- see the FileType enumeration for details

bPersistentFile

- Indicates if the file is meant to persist between sessions / ignition cycles.

bSystemFile

- Indicates if the file is meant to be passed thru core to elsewhere on the system.

iCorrelationID

- A unique ID that correlates each RPCRequest and RPCResponse.

Throws

- SdlException

putFileStream(InputStream
inputStream, String fileName,
Long offset, Long length, FileType
fileType, Boolean isPersistentFile,
Boolean isSystemFile, Boolean
isPayloadProtected, Integer
correlationId)

Used to push a binary stream of file data onto the module from a mobile device. Responses are captured through callback on IProxyListener.

```
public com.smartdevicelink.proxy.RPCStreamController putFileStream(  
    InputStream inputStream, String fileName, Long offset, Long length,  
    FileType fileType, Boolean isPersistentFile, Boolean isSystemFile, Boolean  
    isPayloadProtected, Integer correlationId) throws SdlException
```

Parameters

`inputStream`

The input stream of byte data that will be read from.

`fileName`

The SDL file reference name used by the RPC.

`offset`

The data offset in bytes. A value of zero is used to indicate data starting from the beginning of the file and a value greater than zero is used for resuming partial data chunks.

`length`

The total length of the file being sent.

`fileType`

The selected file type. See the `{@link FileType}` enum for details.

`isPersistentFile`

Indicates if the file is meant to persist between sessions / ignition cycles.

`isSystemFile`

Indicates if the file is meant to be passed through core to elsewhere in the system.

isPayloadProtected

correlationId

A unique id that correlates each RPCRequest and RPCResponse.

Throws

- SdlException

endPutFileStream()

Used to end an existing putFileStream that was previously initiated with any putFileStream method.

```
public void endPutFileStream()
```


putfile(String sdlFileName, FileType fileType, Boolean persistentFile, byte[] fileData, Integer correlationID)

Used to push a binary data onto the SDL module from a mobile device, such as icons and album art. Not supported on first generation SDL vehicles. Responses are captured through callback on IProxyListener.

```
public void putfile(String sdlFileName,FileType fileType,Boolean persistentFile,byte[] fileData,Integer correlationID) throws SdlException
```

Parameters

sdlFileName

-File reference name.

fileType

-Selected file type.

persistentFile

-Indicates if the file is meant to persist between sessions / ignition cycles.

fileData

byte array of data of the file that is to be sent

correlationID

-A unique ID that correlates each RPCRequest and RPCResponse.

Throws

- SdlException

deletefile(String sdlFileName, Integer correlationID)

Used to delete a file resident on the SDL module in the app's local cache. Not supported on first generation SDL vehicles. Responses are captured through callback on IProxyListener.

```
public void deletefile(String sdlFileName,Integer correlationID) throws SdlException
```

Parameters

sdIFileName

-File reference name.

correlationID

-A unique ID that correlates each RPCRequest and RPCResponse.

Throws

- SdlException

listfiles(Integer correlationID)

Requests the current list of resident filenames for the registered app. Not supported on first generation SDL vehicles. Responses are captured through callback on IProxyListener.

```
public void listfiles(Integer correlationID) throws SdlException
```

Parameters

correlationID

-A unique ID that correlates each RPCRequest and RPCResponse.

Throws

- SdlException

setappicon(String sdlFileName, Integer correlationID)

Used to set existing local file on SDL as the app's icon. Not supported on first generation SDL vehicles. Responses are captured through callback on IProxyListener.

```
public void setappicon(String sdlFileName,Integer correlationID) throws SdlException
```

Parameters

sdlFileName

-File reference name.

correlationID

-A unique ID that correlates each RPCRequest and RPCResponse.

Throws

- SdlException

setDisplaylayout(String displayLayout, Integer correlationID)

Set an alternate display layout. If not sent, default screen for given platform will be shown. Responses are captured through callback on IProxyListener.

```
public void setdisplaylayout(String displayLayout,Integer correlationID)  
throws SdlException
```

Parameters

displayLayout

-Predefined or dynamically created screen layout.

correlationID

-A unique ID that correlates each RPCRequest and RPCResponse.

Throws

- SdlException

```
setDisplayLayout( String  
displayLayout,  
TemplateColorScheme  
dayColorScheme,  
TemplateColorScheme  
nightColorScheme, Integer  
correlationID )
```

Set an alternate display layout. If not sent, default screen for given platform will be shown. Responses are captured through callback on IProxyListener.

```
public void setdisplaylayout(String displayLayout,TemplateColorScheme
dayColorScheme,TemplateColorScheme nightColorScheme,Integer
correlationID) throws SdlException
```

Parameters

displayLayout

-Predefined or dynamically created screen layout.

dayColorScheme

a TemplateColorScheme object representing the colors that will be used for day color scheme

nightColorScheme

a TemplateColorScheme object representing the colors that will be used for night color scheme

correlationID

-A unique ID that correlates each RPCRequest and RPCResponse.

Throws

- SdlException

getSystemCapabilityManager()

Gets the SystemCapabilityManager.

```
public SystemCapabilityManager getSystemCapabilityManager()
```

isCapabilitySupported(SystemCapabilityType systemCapabilityType)

```
public boolean isCapabilitySupported(SystemCapabilityType  
systemCapabilityType)
```

Parameters

systemCapabilityType

getCapability(
SystemCapabilityType
systemCapabilityType,
OnSystemCapabilityListener
scListener)

```
public void getCapability(SystemCapabilityType systemCapabilityType,  
OnSystemCapabilityListener scListener)
```

Parameters

systemCapabilityType
scListener

getCapability(
SystemCapabilityType
systemCapabilityType)

```
public java.lang.Object getCapability(SystemCapabilityType  
systemCapabilityType)
```

Parameters

systemCapabilityType

```
addOnSystemCapabilityListener(  
SystemCapabilityType  
systemCapabilityType,  
OnSystemCapabilityListener  
listener )
```

Add a listener to be called whenever a new capability is retrieved

```
public void addOnSystemCapabilityListener(SystemCapabilityType  
systemCapabilityType, OnSystemCapabilityListener listener)
```

Parameters

systemCapabilityType

Type of capability desired

listener

callback to execute upon retrieving capability

removeOnSystemCapabilityListener (SystemCapabilityType systemCapabilityType, OnSystemCapabilityListener listener)

Remove an OnSystemCapabilityListener that was previously added

```
public boolean removeOnSystemCapabilityListener(  
    SystemCapabilityType systemCapabilityType,  
    OnSystemCapabilityListener listener)
```

Parameters

systemCapabilityType

Type of capability

listener

the listener that should be removed

getCurrentTransportType()

Gets type of transport currently used by this SdlProxy.

```
public TransportType getCurrentTransportType() throws java.lang.  
IllegalStateException
```

Throws

- java.lang.IllegalStateException

```
setSdlSecurityClassList(  
java.util.List<java.lang.Class<?  
extends  
com.smartdevicelink.security.SdlSecurity  
list )
```

```
public void setSdlSecurityClassList(java.util.List<java.lang.Class<?  
extends com.smartdevicelink.security.SdlSecurityBase>> list)
```

Parameters

list

```
setMinimumProtocolVersion(  
Version minimumProtocolVersion  
)
```

Sets the minimum protocol version that will be permitted to connect. If the protocol version of the head unit connected is below this version, the app will disconnect with an EndService protocol message and will not register.

```
public void setMinimumProtocolVersion(Version  
minimumProtocolVersion)
```

Parameters

minimumProtocolVersion

setMinimumRPCVersion(Version minimumRPCVersion)

The minimum RPC version that will be permitted to connect. If the RPC version of the head unit connected is below this version, an UnregisterAppInterface will be sent.

```
public void setMinimumRPCVersion(Version minimumRPCVersion)
```

Parameters

minimumRPCVersion

isServiceTypeProtected(SessionType sType)

```
public boolean isServiceTypeProtected(SessionType sType)
```

Parameters

sType

addServiceListener(SessionType serviceType, ISdlServiceListener sdlServiceListener)

```
public void addServiceListener(SessionType serviceType,  
ISdlServiceListener sdlServiceListener)
```

Parameters

serviceType

sdlServiceListener

removeServiceListener(
SessionType serviceType,
ISdlServiceListener
sdlServiceListener)

```
public void removeServiceListener(SessionType serviceType,  
ISdlServiceListener sdlServiceListener)
```

Parameters

serviceType


```
sdIServiceListener
```

getAcceptedVideoParams()

```
public VideoStreamingParameters getAcceptedVideoParams()
```

getProxyListener()

```
public IProxyListenerBase getProxyListener()
```

getAppName()

```
public java.lang.String getAppName()
```

getNgnAppName()

```
public java.lang.String getNgnAppName()
```

getAppID()

```
public java.lang.String getAppID()
```

getDeviceInfo()

```
public DeviceInfo getDeviceInfo()
```

getInstanceDT()

```
public long getInstanceDT()
```

setConnectionDetails(String sDetails)

```
public void setConnectionDetails(String sDetails)
```

Parameters

sDetails

getConnectionDetails()

```
public java.lang.String getConnectionDetails()
```

setPoliciesURL(String sText)

```
public void setPoliciesURL(String sText)
```

Parameters

sText

getPoliciesURL()

```
public java.lang.String getPoliciesURL()
```

getIconResumed()

Tells developer whether or not their app icon has been resumed on core.

```
public boolean getIconResumed() throws SdlException
```

Throws

- SdlException

getRegisterAppInterfaceResponse()

Method to retrieve the RegisterAppInterface Response message that was sent back from the module. It contains various attributes about the connected module and can be used to adapt to different module types and their supported features.

```
public RegisterAppInterfaceResponse getRegisterAppInterfaceResponse()  
(
```

getCurrentHMIStatus()

Get the current OnHMISStatus

```
public OnHMISStatus getCurrentHMIStatus()  
(
```

getAuthToken()

Retrieves the auth token, if any, that was attached to the StartServiceACK for the RPC service from the module. For example, this should be used to login to a user account.

```
public java.lang.String getAuthToken()
```

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class SdlProxyBuilder

Hierarchy

java.lang.Object

Package

com.smartdevicelink.proxy

Quick Jump

Methods

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class SdlProxyBuilder.Builder

Hierarchy

java.lang.Object

Package

com.smartdevicelink.proxy

Quick Jump

Constructors

- `Builder( IProxyListenerALM listener, String appId, String appName, Boolean isMediaApp )`
- `Builder( IProxyListenerALM listener, String appId, String appName, Boolean isMediaApp, Context context )`

Methods

- `setService(Service val)`
- `setSdlProxyConfigurationResources (SdlProxyConfigurationResources val)`
- `setTtsName(java.util.Vector<TTSCheck> val)`
- `setShortAppName(String val)`
- `setVrSynonyms(java.util.Vector<java.lang.String> val)`
- `setSdlMessageVersion(SdlMsgVersion val)`
- `setLangDesired(Language val)`
- `setHMI LangDesired(Language val)`
- `setVrAppHMISypes(java.util.Vector<AppHMIType> val)`
- `setAutoActivateID(String val)`
- `setCallbackToUIThread(boolean val)`
- `setPreRegister(boolean val)`
- `setAppResumeDataHash(String val)`
- `setTransportType(BaseTransportConfig val)`
- `setSdlSecurity(java.util.List<java.lang.Class<? extends com.smartdevicelink.security.SdlSecurityBase>> val)`
- `setDayColorScheme(TemplateColorScheme val)`
- `setNightColorScheme(TemplateColorScheme val)`
- `build()`

Constructors

Builder(IProxyListenerALM listener, String appId, String appName, Boolean isMediaApp)

```
public Builder(IProxyListenerALM listener,String appId,String appName,  
Boolean isMediaApp)
```

Parameters

listener
appId
appName
isMediaApp

Builder(IProxyListenerALM listener, String appId, String appName, Boolean isMediaApp, Context context)

```
public Builder(IProxyListenerALM listener,String appId,String appName,  
Boolean isMediaApp,Context context)
```

Parameters

listener
appId
appName
isMediaApp
context

Methods

setService(Service val)

```
public com.smartdevicelink.proxy.SdlProxyBuilder.Builder setService(  
Service val)
```

Parameters

val

setSdlProxyConfigurationResources
(
SdlProxyConfigurationResources
val)

```
public com.smartdevicelink.proxy.SdlProxyBuilder.Builder  
setSdlProxyConfigurationResources(SdlProxyConfigurationResources val  
)
```

Parameters

```
val
```

setTtsName(
java.util.Vector<TTSCChunk> val)

```
public com.smartdevicelink.proxy.SdlProxyBuilder.Builder setTtsName(  
java.util.Vector<TTSCChunk> val)
```

Parameters

```
val
```

setShortAppName(String val)

```
public com.smartdevicelink.proxy.SdlProxyBuilder.Builder  
setShortAppName(String val)
```

Parameters

val

setVrSynonyms(
java.util.Vector<java.lang.String>
val)

```
public com.smartdevicelink.proxy.SdlProxyBuilder.Builder  
setVrSynonyms(java.util.Vector<java.lang.String> val)
```

Parameters

val

setSdlMessageVersion(SdlMsgVersion val)

```
public com.smartdevicelink.proxy.SdlProxyBuilder.Builder  
setSdlMessageVersion(SdlMsgVersion val)
```

Parameters

val

setLangDesired(Language val)

```
public com.smartdevicelink.proxy.SdlProxyBuilder.Builder  
setLangDesired(Language val)
```

Parameters

```
val
```

setHMI LangDesired(Language
val)

```
public com.smartdevicelink.proxy.SdlProxyBuilder.Builder  
setHMI LangDesired(Language val)
```

Parameters

```
val
```

setVrAppHMIMTypes(
java.util.Vector<AppHMIMType>
val)

```
public com.smartdevicelink.proxy.SdlProxyBuilder.Builder  
setVrAppHMIMTypes(java.util.Vector<AppHMIMType> val)
```

Parameters

val

setAutoActivateID(String val)

```
public com.smartdevicelink.proxy.SdlProxyBuilder.Builder  
setAutoActivateID(String val)
```

Parameters

val

setCallbackToUiThread(boolean val)

```
public com.smartdevicelink.proxy.SdlProxyBuilder.Builder  
setCallbackToUiThread(boolean val)
```

Parameters

val

setPreRegister(boolean val)

```
public com.smartdevicelink.proxy.SdlProxyBuilder.Builder  
setPreRegister(boolean val)
```

Parameters

```
val
```

setAppResumeDataHash(String
val)

```
public com.smartdevicelink.proxy.SdlProxyBuilder.Builder  
setAppResumeDataHash(String val)
```

Parameters

```
val
```

setTransportType(
BaseTransportConfig val)

```
public com.smartdevicelink.proxy.SdlProxyBuilder.Builder  
setTransportType(BaseTransportConfig val)
```

Parameters

```
val
```

```
setSdlSecurity(  
java.util.List<java.lang.Class<?  
extends  
com.smartdevicelink.security.SdlSecurityBase>>  
val )
```

```
public com.smartdevicelink.proxy.SdlProxyBuilder.Builder  
setSdlSecurity(java.util.List<java.lang.Class<? extends com.  
smartdevicelink.security.SdlSecurityBase>> val)
```

Parameters

```
val
```

setDayColorScheme(
TemplateColorScheme val)

```
public com.smartdevicelink.proxy.SdlProxyBuilder.Builder  
setDayColorScheme(TemplateColorScheme val)
```

Parameters

```
val
```

setNightColorScheme(
TemplateColorScheme val)

```
public com.smartdevicelink.proxy.SdlProxyBuilder.Builder  
setNightColorScheme(TemplateColorScheme val)
```

Parameters

```
val
```

build()

```
public com.smartdevicelink.proxy.SdlProxyALM build() throws  
SdlException
```

Throws

- `SdlException`

Inherited Methods

From Class	Methods
<code>java.lang.Object</code>	<code>getClass</code> , <code>hashCode</code> , <code>equals</code> , <code>clone</code> , <code>toString</code> , <code>notify</code> , <code>notifyAll</code> , <code>wait</code> , <code>wait</code> , <code>wait</code> , <code>finalize</code>

Class SdlProxyConfigurationResources

Hierarchy

`java.lang.Object`

Package

`com.smartdevicelink.proxy`

Quick Jump

Constructors

- `SdlProxyConfigurationResources()`
- `SdlProxyConfigurationResources( String sdlConfigurationFilePath, TelephonyManager telephonyManager )`

Methods

- `setSdlConfigurationFilePath(String sdlConfigurationFilePath)`
- `getSdlConfigurationFilePath()`
- `setTelephonyManager(TelephonyManager telephonyManager)`
- `getTelephonyManager()`

Constructors

SdlProxyConfigurationResources ()

```
public SdlProxyConfigurationResources()
```

SdlProxyConfigurationResources(String sdlConfigurationFilePath, TelephonyManager telephonyManager)

```
public SdlProxyConfigurationResources(String sdlConfigurationFilePath,  
TelephonyManager telephonyManager)
```

Parameters

sdlConfigurationFilePath
telephonyManager

Methods

setSdlConfigurationFilePath(String sdlConfigurationFilePath)


```
public void setSdlConfigurationFilePath(String sdlConfigurationFilePath)
```

Parameters

```
sdlConfigurationFilePath
```

getSdlConfigurationFilePath()

```
public java.lang.String getSdlConfigurationFilePath()
```

setTelephonyManager(TelephonyManager telephonyManager)

```
public void setTelephonyManager(TelephonyManager  
telephonyManager)
```

Parameters

telephonyManager

getTelephonyManager()

```
public android.telephony.TelephonyManager getTelephonyManager()
```

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class SdlRemoteDisplay

Hierarchy

java.lang.Object

android.app.Dialog

android.app.Presentation

Package

com.smartdevicelink.streaming.video

Overview

SdlRemoteDisplay is an abstract class that should be extended by developers to create their remote displays. All logic for UI events can be stored in their extension.

NOTE: When the UI changes (buttons appear, layouts change, etc) the developer should call {@link #invalidate()} to alert any other interfaces that are listening for those types of events.

Quick Jump

Fields

- `w`
- `mainView`
- `handler`

- `uiHandler`
- `callback`
- `mStartRefreshTaskCallback`

Constructors

- `SdlRemoteDisplay( Context context, Display display )`

Methods

- `onCreate(Bundle savedInstanceState)`
- `startRefreshTask()`
- `stopRefreshTask()`
- `getMainView()`
- `invalidate()`
- `handleMotionEvent(MotionEvent motionEvent)`
- `stop()`
- `dismissPresentation()`

Fields

W

```
protected android.view.Window w
```

mainView

```
protected android.view.View mainView
```

handler

```
protected android.os.Handler handler
```

uiHandler

```
protected android.os.Handler uiHandler
```

callback

```
protected com.smartdevicelink.streaming.video.SdlRemoteDisplay.  
Callback callback
```

mStartRefreshTaskCallback

```
protected java.lang.Runnable mStartRefreshTaskCallback
```

Constructors

SdlRemoteDisplay(Context
context, Display display)

```
public SdlRemoteDisplay(Context context,Display display)
```

Parameters

context
display

Methods

onCreate(Bundle
savedInstanceState)

```
protected void onCreate(Bundle savedInstanceState)
```

Parameters

savedInstanceState

startRefreshTask()

```
protected void startRefreshTask()
```

stopRefreshTask()

```
protected void stopRefreshTask()
```

getMainView()

```
public android.view.View getMainView()
```

invalidate()

```
public void invalidate()
```


handleMotionEvent(MotionEvent motionEvent)

```
public void handleMotionEvent(MotionEvent motionEvent)
```

Parameters

motionEvent

stop()

```
public void stop()
```

dismissPresentation()

```
public void dismissPresentation()
```

Inherited Methods

From Class	Methods
android.app.Presentation	getDisplay, getResources, onStart, onStop, show, onDisplayRemoved, onDisplayChanged

android.app.Dialog

getContext, getActionBar,
setOwnerActivity, getOwnerActivity,
isShowing, create, show, hide, dismiss,
onCreate, onStart, onStop,
onSaveInstanceState,
onRestoreInstanceState, getWindow,
getCurrentFocus, findViewById,
setContentView, setContentView,
setContentView, addContentView,
setTitle, setTitle, onKeyDown,
onKeyLongPress, onKeyUp,
onKeyMultiple, onBackPressed,
onKeyShortcut, onTouchEvent,
onTrackballEvent, onGenericMotionEvent,
onWindowAttributesChanged,
onContentChanged,
onWindowFocusChanged,
onAttachedToWindow,
onDetachedFromWindow,
dispatchKeyEvent,
dispatchKeyShortcutEvent,
dispatchTouchEvent,
dispatchTrackballEvent,
dispatchGenericMotionEvent,
dispatchPopulateAccessibilityEvent,
onCreatePanelView, onCreatePanelMenu,
onPreparePanel, onMenuOpened,
onMenuItemSelected, onPanelClosed,
onCreateOptionsMenu,
onPrepareOptionsMenu,
onOptionsItemSelected,
onOptionsMenuClosed,
openOptionsMenu, closeOptionsMenu,
invalidateOptionsMenu,
onCreateContextMenu,
registerForContextMenu,
unregisterForContextMenu,
openContextMenu,
onContextItemSelected,
onContextMenuClosed,
onSearchRequested,
onSearchRequested, getSearchEvent,
onWindowStartingActionMode,
onWindowStartingActionMode,
onActionModeStarted,
onActionModeFinished, takeKeyEvents,
requestWindowFeature

java.lang.Object

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

Interface SdlRemoteDisplay.Callback

Package

com.smartdevicelink.streaming.video

Quick Jump

Methods

- [onCreated\(SdlRemoteDisplay remoteDisplay\)](#)
- [onInvalidated\(SdlRemoteDisplay remoteDisplay\)](#)

Methods

onCreated(SdlRemoteDisplay remoteDisplay)

```
public void onCreate(SdlRemoteDisplay remoteDisplay)
```

Parameters

remoteDisplay

onInvalidated(
SdlRemoteDisplay remoteDisplay
)

```
public void onInvalidated(SdlRemoteDisplay remoteDisplay)
```

Parameters

remoteDisplay

Class

SdlRemoteDisplay.Creator

Hierarchy

java.lang.Object

Package

com.smartdevicelink.streaming.video

All Implemented Interfaces

java.util.concurrent.Callable

Quick Jump

Constructors

- [Creator\(Context context, Display display, SdlRemoteDisplay remoteDisplay, java.lang.Class<? extends com.smartdevicelink.streaming.video.SdlRemoteDisplay> remoteDisplayClass, SdlRemoteDisplay.Callback callback \)](#)

Methods

- [call\(\)](#)

Constructors

Creator(Context context, Display display, SdlRemoteDisplay remoteDisplay, java.lang.Class<? extends com.smartdevicelink.streaming.video.SdlRemoteDisplayClass, SdlRemoteDisplay.Callback callback)

```
public Creator(Context context, Display display, SdlRemoteDisplay remoteDisplay, java.lang.Class<? extends com.smartdevicelink.streaming.video.SdlRemoteDisplayClass, SdlRemoteDisplay.Callback callback)
```

Parameters

context
display

remoteDisplay
remoteDisplayClass
callback

Methods

call()

```
public java.lang.Boolean call()
```

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class SdlRouterService

Hierarchy

java.lang.Object

android.content.Context

android.content.ContextWrapper

android.app.Service

Package

com.smartdevicelink.transport

Overview

This class should not be modified by anyone outside of the approved contributors of the SmartDeviceLink project. This service is a central point of communication between hardware and the registered clients. It will multiplex a single transport to provide a connection for a theoretical infinite amount of SDL sessions.

Quick Jump

Fields

- ROUTER_SERVICE_VERSION_NUMBER
- REGISTER_NEWER_SERVER_INSTANCE_ACTION
- SDL_NOTIFICATION_FAQS_PAGE
- START_SERVICE_ACTION

- REGISTER_WITH_ROUTER_ACTION
- MESSAGE_STATE_CHANGE
- MESSAGE_READ
- MESSAGE_WRITE
- MESSAGE_DEVICE_NAME
- MESSAGE_LOG
- SDL_DEVICE_STATUS_SHARED_PREFS
- registeredApps

Constructors

- SdlRouterService()

Methods

- onBind(Intent intent)
- onUnbind(Intent intent)
- formatMessage(SdlRouterService.RegisteredApp app, Message message)
- onCreate()
- deployNextRouterService()
- startUpSequence()
- onStartCommand(Intent intent, int flags, int startId)
- onDestroy()
- resetForegroundTimeout(long delay)
- cancelForegroundTimeout()
- getConnectedDeviceName() **Deprecated**
- shouldServiceRemainOpen(Intent intent)
- closeSelf()
- onTransportConnected(TransportType type) **Deprecated**
- onTransportConnected(TransportRecord record)
- onTransportDisconnected(TransportType type) **Deprecated**
- onTransportDisconnected(TransportRecord record)
- onTransportError(TransportType transportType) **Deprecated**
- onTransportError(TransportRecord transport)
- onPacketRead(SdlPacket packet)
- writeBytesToTransport(Bundle bundle)
- sendPacketToRegisteredApp(SdlPacket packet)
- bluetoothQueryAndConnect()
- setBluetoothPrefs(int level, String prefLocation) **Deprecated**
- getBluetoothPrefs(String prefLocation) **Deprecated**
- setSDLConnectedStatus(String address, boolean hasSDLConnected)
- isFirstStatusCheck(String address)
- hasSDLConnected(String address)
- getLocalRouterService(Intent launchIntent, ComponentName name) **Deprecated**
- getNextTask(TransportType transportType)

Fields

ROUTER_SERVICE_VERSION_NUMBER

```
protected static final int ROUTER_SERVICE_VERSION_NUMBER
```

Constant Value

9

REGISTER_NEWER_SERVER_INSTANCE_ACTION

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

Constant Value

"com.sdl.android.newservice"

SDL_NOTIFICATION_FAQS_PAGE

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

Constant Value

"<https://smartdevicelink.com/en/guides/android/frequently-asked-questions/sdl-notifications/>"

START_SERVICE_ACTION

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

Constant Value

"sdl.router.startservice"

REGISTER_WITH_ROUTER_ACTION

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

Constant Value

"com.sdl.android.register"

MESSAGE_STATE_CHANGE

```
public static final int MESSAGE_STATE_CHANGE
```

Constant Value

1

MESSAGE_READ

```
public static final int MESSAGE_READ
```

Constant Value

2

MESSAGE_WRITE

```
public static final int MESSAGE_WRITE
```

Constant Value

3

MESSAGE_DEVICE_NAME

```
public static final int MESSAGE_DEVICE_NAME
```

Constant Value

4

MESSAGE_LOG

```
public static final int MESSAGE_LOG
```

Constant Value

5

SDL_DEVICE_STATUS_SHARED_PREFS

```
protected static final java.lang.String  
SDL_DEVICE_STATUS_SHARED_PREFS
```

Constant Value

"sdl.device.status"

registeredApps

```
public static java.util.HashMap registeredApps
```


Inherited Fields

From Class	Fields
android.app.Service	START_CONTINUATION_MASK, START_FLAG_REDELIVERY, START_FLAG_RETRY, START_NOT_STICKY, START_REDELIVER_INTENT, START_STICKY, START_STICKY_COMPATIBILITY

android.content.Context

ACCESSIBILITY_SERVICE,
ACCOUNT_SERVICE, ACTIVITY_SERVICE,
ALARM_SERVICE, APPWIDGET_SERVICE,
APP_OPS_SERVICE, AUDIO_SERVICE,
BATTERY_SERVICE, BIND_ABOVE_CLIENT,
BIND_ADJUST_WITH_ACTIVITY,
BIND_ALLOW_OOM_MANAGEMENT,
BIND_AUTO_CREATE,
BIND_DEBUG_UNBIND, BIND_IMPORTANT,
BIND_NOT_FOREGROUND,
BIND_WAIVE_PRIORITY,
BLUETOOTH_SERVICE, CAMERA_SERVICE,
CAPTIONING_SERVICE,
CARRIER_CONFIG_SERVICE,
CLIPBOARD_SERVICE,
CONNECTIVITY_SERVICE,
CONSUMER_IR_SERVICE,
CONTEXT_IGNORE_SECURITY,
CONTEXT_INCLUDE_CODE,
CONTEXT_RESTRICTED,
DEVICE_POLICY_SERVICE,
DISPLAY_SERVICE, DOWNLOAD_SERVICE,
DROPBOX_SERVICE,
FINGERPRINT_SERVICE,
INPUT_METHOD_SERVICE,
INPUT_SERVICE,
JOB_SCHEDULER_SERVICE,
KEYGUARD_SERVICE,
LAUNCHER_APPS_SERVICE,
LAYOUT_INFLATER_SERVICE,
LOCATION_SERVICE,
MEDIA_PROJECTION_SERVICE,
MEDIA_ROUTER_SERVICE,
MEDIA_SESSION_SERVICE, MIDI_SERVICE,
MODE_APPEND,
MODE_ENABLE_WRITE_AHEAD_LOGGING,
MODE_MULTI_PROCESS, MODE_PRIVATE,
MODE_WORLD_READABLE,
MODE_WORLD_WRITEABLE,
NETWORK_STATS_SERVICE,
NFC_SERVICE, NOTIFICATION_SERVICE,
NSD_SERVICE, POWER_SERVICE,
PRINT_SERVICE, RESTRICTIONS_SERVICE,
SEARCH_SERVICE, SENSOR_SERVICE,
STORAGE_SERVICE, TELECOM_SERVICE,
TELEPHONY_SERVICE,
TELEPHONY_SUBSCRIPTION_SERVICE

Constructors

SdlRouterService()

```
public SdlRouterService()
```

Methods

onBind(Intent intent)

```
public android.os.IBinder onBind(Intent intent)
```

Parameters

intent

onUnbind(Intent intent)

```
public boolean onUnbind(Intent intent)
```

Parameters

intent

formatMessage(SdIRouterService.RegisteredApp app, Message message)

Formats the message for the app that is to receive it

```
protected boolean formatMessage(SdIRouterService.RegisteredApp app  
,Message message)
```

Parameters

app
message

onCreate()

```
public void onCreate()
```

deployNextRouterService()

The method will attempt to start up the next router service in line based on the sorting criteria of best router service.

```
protected void deployNextRouterService()
```


startUpSequence()

```
public void startUpSequence()
```

onStartCommand(Intent intent, int flags, int startId)

```
public int onStartCommand(Intent intent,int flags,int startId)
```

Parameters

intent
flags
startId

onDestroy()

```
public void onDestroy()
```

resetForegroundTimeOut(long delay)

```
public void resetForegroundTimeOut(long delay)
```

Parameters

delay

cancelForegroundTimeOut()

```
public void cancelForegroundTimeOut()
```

getConnectedDeviceName()

Deprecated

```
public java.lang.String getConnectedDeviceName()
```

shouldServiceRemainOpen(Intent intent)

1. If the app has SDL shut off, shut down 2. if The app has an Alt Transport address or was started by one, stay open 3. If Bluetooth is off/NA shut down 4. Anything else

```
public boolean shouldServiceRemainOpen(Intent intent)
```

Parameters

intent

closeSelf()

This method is needed so that apps that choose not to implement this as a service as defined by Android, but rather just a simple class we have to know how to shut down.

```
public void closeSelf()
```

onTransportConnected(TransportType type)

Deprecated

```
public void onTransportConnected(TransportType type)
```

Parameters

type

onTransportConnected(TransportRecord record)

```
public void onTransportConnected(TransportRecord record)
```

Parameters

record

onTransportDisconnected(TransportType type)

Deprecated

```
public void onTransportDisconnected(TransportType type)
```

Parameters

type

onTransportDisconnected(TransportRecord record)

```
public void onTransportDisconnected(TransportRecord record)
```

Parameters

record

onTransportError(TransportType transportType)

Deprecated

```
public void onTransportError(TransportType transportType)
```

Parameters

transportType

onTransportError(TransportRecord transport)

```
public void onTransportError(TransportRecord transport)
```

Parameters

transport

onPacketRead(SdIPacket packet)

```
public void onPacketRead(SdIPacket packet)
```

Parameters

packet

writeBytesToTransport(Bundle bundle)

```
public boolean writeBytesToTransport(Bundle bundle)
```


Parameters

bundle

sendPacketToRegisteredApp(SdIPacket packet)

This will send the received packet to the registered service. It will default to the single registered "foreground" app. This can be overridden to provide more specific functionality.

```
public boolean sendPacketToRegisteredApp(SdIPacket packet)
```

Parameters

packet

the packet that is received

bluetoothQueryAndConnect()

This function looks through the phones currently paired bluetooth devices. If one of the devices' names contain "sync", or livio it will attempt to connect the RFCOMM And start SDL

```
public synchronized boolean bluetoothQueryAndConnect()
```

setBluetoothPrefs(int level, String prefLocation)

Deprecated

```
public static void setBluetoothPrefs(int level,String prefLocation)
```

Parameters

level

The level of bluetooth connecting method that last worked

prefLocation

Where the preference should be stored

getBluetoothPrefs(String prefLocation)

Deprecated

```
public static int getBluetoothPrefs(String prefLocation)
```

Parameters

prefLocation

setSDLConnectedStatus(String address, boolean hasSDLConnected)

Set the connection establishment status of the particular device

```
protected void setSDLConnectedStatus(String address,boolean  
hasSDLConnected)
```

Parameters

address

address of the device in quesiton

hasSDLConnected

true if a connection has been established, false if not

isFirstStatusCheck(String address)

Checks to see if a device address has connected to SDL before.

```
protected boolean isFirstStatusCheck(String address)
```

Parameters

address

the mac address of the device in quesiton

hasSDLConnected(String address)

Checks to see if a device address has connected to SDL before.

```
protected boolean hasSDLConnected(String address)
```

Parameters

address

the mac address of the device in quesiton

getLocalRouterService(Intent launchIntent, ComponentName name)

Deprecated

```
protected static com.smartdevicelink.transport.SdlRouterService.  
LocalRouterService getLocalRouterService(Intent launchIntent,  
ComponentName name)
```

Parameters

launchIntent
name

getNextTask(TransportType transportType)

Method for finding the next, highest priority write task from all connected apps.

```
protected com.smartdevicelink.transport.SdlRouterService.  
PacketWriteTask getNextTask(TransportType transportType)
```

Parameters

transportType

Inherited Methods

From Class	Methods
android.app.Service	getApplication, onCreate, onStart, onStartCommand, onDestroy, onConfigurationChanged, onLowMemory, onTrimMemory, onBind, onBind, onUnbind, onRebind, onTaskRemoved, stopSelf, stopSelf, stopSelfResult, startForeground, stopForeground, dump

android.content.ContextWrapper

attachBaseContext, getBaseContext,
getAssets, getResources,
getPackageManager,
getContentResolver, getMainLooper,
getApplicationContext, setTheme,
getTheme, getClassLoader,
getPackageName, getApplicationInfo,
getPackageResourcePath,
getPackageCodePath,
getSharedPreferences, openFileInput,
openFileOutput, deleteFile,
getFileStreamPath, fileList, getFilesDir,
getNoBackupFilesDir, getExternalFilesDir,
getExternalFilesDirs, getObbDir,
getObbDirs, getCacheDir,
getCodeCacheDir, getExternalCacheDir,
getExternalCacheDirs,
getExternalMediaDirs, getDir,
openOrCreateDatabase,
openOrCreateDatabase, deleteDatabase,
getDatabasePath, databaseList,
getWallpaper, peekWallpaper,
getWallpaperDesiredMinimumWidth,
getWallpaperDesiredMinimumHeight,
setWallpaper, setWallpaper,
clearWallpaper, startActivity,
startActivity, startActivities,
startActivities, startIntentSender,
startIntentSender, sendBroadcast,
sendBroadcast, sendOrderedBroadcast,
sendOrderedBroadcast,
sendBroadcastAsUser,
sendBroadcastAsUser,
sendOrderedBroadcastAsUser,
sendStickyBroadcast,
sendStickyOrderedBroadcast,
removeStickyBroadcast,
sendStickyBroadcastAsUser,
sendStickyOrderedBroadcastAsUser,
removeStickyBroadcastAsUser,
registerReceiver, registerReceiver,
unregisterReceiver, startService,
stopService, bindService, unbindService,
startInstrumentation, getSystemService,
getSystemServiceName,
checkPermission,
checkCallingPermission

android.content.Context

getAssets, getResources,
getPackageManager,
getContentResolver, getMainLooper,
getApplicationContext,
registerComponentCallbacks,
unregisterComponentCallbacks, getText,
getString, getString, getColor,
getDrawable, getColorStateList,
setTheme, getTheme,
obtainStyledAttributes,
obtainStyledAttributes,
obtainStyledAttributes,
obtainStyledAttributes, getClassLoader,
getPackageName, getApplicationInfo,
getPackageResourcePath,
getPackageCodePath,
getSharedPreferences, openFileInput,
openFileOutput, deleteFile,
getFileStreamPath, getFilesDir,
getNoBackupFilesDir, getExternalFilesDir,
getExternalFilesDirs, getObbDir,
getObbDirs, getCacheDir,
getCodeCacheDir, getExternalCacheDir,
getExternalCacheDirs,
getExternalMediaDirs, fileList, getDir,
openOrCreateDatabase,
openOrCreateDatabase, deleteDatabase,
getDatabasePath, databaseList,
getWallpaper, peekWallpaper,
getWallpaperDesiredMinimumWidth,
getWallpaperDesiredMinimumHeight,
setWallpaper, setWallpaper,
clearWallpaper, startActivity,
startActivity, startActivities,
startActivities, startIntentSender,
startIntentSender, sendBroadcast,
sendBroadcast, sendOrderedBroadcast,
sendOrderedBroadcast,
sendBroadcastAsUser,
sendBroadcastAsUser,
sendOrderedBroadcastAsUser,
sendStickyBroadcast,
sendStickyOrderedBroadcast,
removeStickyBroadcast,
sendStickyBroadcastAsUser,
sendStickyOrderedBroadcastAsUser,
removeStickyBroadcastAsUser

java.lang.Object

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

Class

SdlRouterService.PacketWriteTask

Hierarchy

java.lang.Object

Package

com.smartdevicelink.transport

All Implemented Interfaces

java.lang.Runnable

Overview

A runnable task for writing out packets.

Quick Jump

Constructors

- `PacketWriteTask( byte[] bytes, int offset, int size, int priorityCoefficient )`
- `PacketWriteTask( byte[] bytes, int offset, int size, int priorityCoefficient, TransportType transportType )`
- `PacketWriteTask( Bundle bundle )`

Methods

- `setTransportType(TransportType transportType)`
- `run()`

Constructors

`PacketWriteTask(byte[] bytes,
int offset, int size, int
priorityCoefficient)`

```
public PacketWriteTask(byte[] bytes,int offset,int size,int  
priorityCoefficient)
```

Parameters

bytes
offset
size
priorityCoefficient

PacketWriteTask(byte[] bytes,
int offset, int size, int
priorityCoefficient, TransportType
transportType)

```
public PacketWriteTask(byte[] bytes,int offset,int size,int  
priorityCoefficient,TransportType transportType)
```

Parameters

bytes
offset
size
priorityCoefficient
transportType

PacketWriteTask(Bundle bundle)

```
public PacketWriteTask(Bundle bundle)
```


Parameters

bundle

Methods

setTransportType(TransportType
transportType)

```
protected void setTransportType(TransportType transportType)
```

Parameters

transportType

run()

```
public void run()
```

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class SdlRouterStatusProvider

Hierarchy

java.lang.Object

Package

com.smartdevicelink.transport

Quick Jump

Constructors

- `SdlRouterStatusProvider( Context context, ComponentName service, SdlRouterStatusProvider.ConnectedStatusCallback callback )`

Methods

- `setFlags(int flags)`
- `checkIsConnected()`
- `cancel()`

Constructors

`SdlRouterStatusProvider(
Context context,
ComponentName service,
SdlRouterStatusProvider.ConnectedSta
callback)`

```
public SdlRouterStatusProvider(Context context,ComponentName  
service,SdlRouterStatusProvider.ConnectedStatusCallback callback)
```

Parameters

context

service

callback

Methods

setFlags(int flags)

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

Parameters

flags

checkIsConnected()

```
public void checkIsConnected()
```

cancel()

```
public void cancel()
```

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Interface SdlRouterServiceProvider.Connected

Package

com.smartdevicelink.transport

Quick Jump

Methods

- [onConnectionStatusUpdate\(boolean connected,ComponentName service,Context context\)](#)

Methods

**onConnectionStatusUpdate(
boolean connected,
ComponentName service,
Context context)**

```
public void onConnectionStatusUpdate(boolean connected,  
ComponentName service,Context context)
```

Parameters

connected
service
context

Class SdlSecurityBase

Hierarchy

java.lang.Object

AbstractSdlSecurityBase

Package

Quick Jump

Fields

- `appService`
- `context`

Constructors

- `SdlSecurityBase()`

Methods

- `getAppService()` **Deprecated**
- `setAppService(Service val)` **Deprecated**
- `getContext()`
- `setContext(Context val)`

Fields

appService

```
protected static android.app.Service appService
```


context

```
protected static android.content.Context context
```

Constructors

SdlSecurityBase()

```
public SdlSecurityBase()
```

Methods

getAppService()

Deprecated

```
public static android.app.Service getAppService()
```

setAppService(Service val)

Deprecated

```
public static void setAppService(Service val)
```

Parameters

```
val
```

getContext()

```
public static android.content.Context getContext()
```

setContext(Context val)

```
public static void setContext(Context val)
```

Parameters

val

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class SdlSession

Hierarchy

java.lang.Object

Package

com.smartdevicelink.SdlConnection

All Implemented Interfaces

[com.smartdevicelink.protocol.heartbeat.IHeartbeatMonitorListener](#)

All Known Subclasses

[com.smartdevicelink.SdlConnection.SdlSession2](#)

Quick Jump

Fields

- [BUFF_READ_SIZE](#)
- [transportConfig](#)
- [sessionListener](#)
- [lockScreenMan](#)
- [sdlSecurity](#)
- [desiredVideoParams](#)
- [acceptedVideoParams](#)
- [sessionId](#)
- [sessionHashId](#)
- [serviceListeners](#)
- [encryptedServices](#)

Constructors

- [SdlSession\(\)](#)

Methods

- [createSession\(byte wiproVersion, ISdlConnectionListener listener, BaseTransportConfig btConfig\)](#)
- [getTransportConfig\(\)](#)
- [getLockScreenMan\(\)](#)
- [getOutgoingHeartbeatMonitor\(\)](#)
- [getIncomingHeartbeatMonitor\(\)](#)
- [setOutgoingHeartbeatMonitor\(IHeartbeatMonitor outgoingHeartbeatMonitor\)](#)
- [setIncomingHeartbeatMonitor\(IHeartbeatMonitor incomingHeartbeatMonitor\)](#)
- [getSessionHashId\(\)](#)
- [getSessionId\(\)](#)
- [getSdlConnection\(\)](#)
- [getMtu\(\)](#)

- getMtu(SessionType type)
- close()
- resetSession()
- startStream(InputStream is,SessionType sType,byte rpcSessionID)
- startStream(SessionType sType,byte rpcSessionID)
- startVideoStream()
- startAudioStream()
- startRPCStream(InputStream is,RPCRequest request,SessionType sType,byte rpcSessionID,byte wiproVersion)
- startRPCStream(RPCRequest request,SessionType sType,byte rpcSessionID,byte wiproVersion)
- pauseRPCStream()
- resumeRPCStream()
- stopRPCStream()
- stopAudioStream()
- stopVideoStream()
- pauseAudioStream()
- pauseVideoStream()
- resumeAudioStream()
- resumeVideoStream()
- createOpenGLInputSurface(int frameRate,int iFrameInterval,int width,int height,int bitrate,SessionType sType,byte rpcSessionID)
- startEncoder()
- releaseEncoder()
- drainEncoder(boolean endOfStream)
- sendStreamPacket(ProtocolMessage pm)
- setSdlSecurity(SdlSecurityBase sec)
- getSdlSecurity()
- startService(SessionType serviceType,byte sessionID,boolean isEncrypted)
- endService(SessionType serviceType,byte sessionID)
- processControlService(ProtocolMessage msg)
- getBroadcastComment(BaseTransportConfig myTransport)
- startSession()
- initialiseSession()
- sendMessage(ProtocolMessage msg)
- getCurrentTransportType()
- getIsConnected()
- isServiceProtected(SessionType sType)
- onTransportDisconnected(String info)
- onTransportDisconnected(String info,boolean availablePrimary,BaseTransportConfig transportConfig)
- onTransportError(String info,Exception e)
- onProtocolMessageReceived(ProtocolMessage msg)
- onHeartbeatTimedOut(byte sessionID)
- onProtocolSessionStarted(SessionType sessionType,byte sessionID,byte version,String correlationID,int hashID,boolean isEncrypted)
- onProtocolSessionEnded(SessionType sessionType,byte sessionID,String correlationID)
- onProtocolError(String info,Exception e)
- sendHeartbeat(IHeartbeatMonitor monitor)
- heartbeatTimedOut(IHeartbeatMonitor monitor)

- onProtocolSessionStartedNACKed(SessionType sessionType,byte sessionID,byte version,String correlationID,java.util.List<java.lang.String> rejectedParams)
- onProtocolSessionEndedNACKed(SessionType sessionType,byte sessionID,String correlationID)
- onProtocolServiceDataACK(SessionType sessionType,int dataSize,byte sessionID)
- onAuthTokenReceived(String authToken,byte sessionID)
- onSecurityInitialized()
- clearConnection()
- checkForOpenMultiplexConnection(SdlConnection connection)
- removeConnection(SdlConnection connection)
- addServiceListener(SessionType serviceType,ISdlServiceListener sdlServiceListener)
- removeServiceListener(SessionType serviceType,ISdlServiceListener sdlServiceListener)
- getServiceListeners()
- setDesiredVideoParams(VideoStreamingParameters params)
- getDesiredVideoParams()
- setAcceptedVideoParams(VideoStreamingParameters params)
- getAcceptedVideoParams()
- getProtocolVersion()
- isTransportForServiceAvailable(SessionType sessionType)

Fields

BUFF_READ_SIZE

```
protected static final int BUFF_READ_SIZE
```

Constant Value

1024

transportConfig

```
protected BaseTransportConfig transportConfig
```

sessionListener

```
protected ISdlConnectionListener sessionListener
```

lockScreenMan

```
protected com.smartdevicelink.proxy.LockScreenManager  
lockScreenMan
```

sdlSecurity

```
protected com.smartdevicelink.security.SdlSecurityBase sdlSecurity
```

desiredVideoParams

```
protected VideoStreamingParameters desiredVideoParams
```

acceptedVideoParams

```
protected VideoStreamingParameters acceptedVideoParams
```

sessionId

```
protected byte sessionId
```

sessionHashId

```
protected int sessionHashId
```

serviceListeners

```
protected java.util.HashMap serviceListeners
```

encryptedServices

```
protected java.util.concurrent.CopyOnWriteArrayList encryptedServices
```

Constructors

SdlSession()

```
public SdlSession()
```

Methods

```
createSession( byte  
wiproVersion,  
ISdlConnectionListener listener,  
BaseTransportConfig btConfig )
```

```
public static com.smartdevicelink.SdlConnection.SdlSession  
createSession(byte wiproVersion, ISdlConnectionListener listener,  
BaseTransportConfig btConfig)
```

Parameters

wiproVersion
listener
btConfig

getTransportConfig()

```
public BaseTransportConfig getTransportConfig()
```

getLockScreenMan()

```
public com.smartdevicelink.proxy.LockScreenManager  
getLockScreenMan()
```

getOutgoingHeartbeatMonitor()

```
public com.smartdevicelink.protocol.heartbeat.IHeartbeatMonitor  
getOutgoingHeartbeatMonitor()
```

getIncomingHeartbeatMonitor()

```
public com.smartdevicelink.protocol.heartbeat.IHeartbeatMonitor  
getIncomingHeartbeatMonitor()
```

setOutgoingHeartbeatMonitor(IHeartbeatMonitor outgoingHeartbeatMonitor)

```
public void setOutgoingHeartbeatMonitor(IHeartbeatMonitor  
outgoingHeartbeatMonitor)
```

Parameters

outgoingHeartbeatMonitor

setIncomingHeartbeatMonitor(
IHeartbeatMonitor
incomingHeartbeatMonitor)

```
public void setIncomingHeartbeatMonitor(IHeartbeatMonitor  
incomingHeartbeatMonitor)
```

Parameters

incomingHeartbeatMonitor

getSessionHashId()

```
public int getSessionHashId()
```

getSessionId()

```
public byte getSessionId()
```

getSdlConnection()

```
public com.smartdevicelink.SdlConnection.SdlConnection  
getSdlConnection()
```

getMtu()

```
public int getMtu()
```

getMtu(SessionType type)

```
public long getMtu(SessionType type)
```

Parameters

type

close()

```
public void close()
```

resetSession()

```
public void resetSession()
```

startStream(InputStream is, SessionType sType, byte rpcSessionID)

```
public void startStream(InputStream is,SessionType sType,byte  
rpcSessionID) throws java.io.IOException
```

Parameters

is
sType
rpcSessionID

Throws

- java.io.IOException

startStream(SessionType sType, byte rpcSessionID)

```
public java.io.OutputStream startStream(SessionType sType,byte  
rpcSessionID) throws java.io.IOException
```

Parameters

sType
rpcSessionID

Throws

- java.io.IOException

startVideoStream()

```
public IVideoStreamListener startVideoStream()
```

startAudioStream()

```
public IAudioStreamListener startAudioStream()
```

startRPCStream(InputStream is, RPCRequest request, SessionType sType, byte rpcSessionID, byte wiproVersion)

```
public void startRPCStream(InputStream is,RPCRequest request,  
SessionType sType,byte rpcSessionID,byte wiproVersion)
```

Parameters

is
request
sType
rpcSessionID
wiproVersion

startRPCStream(RPCRequest
request, SessionType sType, byte
rpcSessionID, byte wiproVersion
)

```
public java.io.OutputStream startRPCStream(RPCRequest request,  
SessionType sType,byte rpcSessionID,byte wiproVersion)
```

Parameters

request
sType
rpcSessionID
wiproVersion

pauseRPCStream()

```
public void pauseRPCStream()
```

resumeRPCStream()

```
public void resumeRPCStream()
```

stopRPCStream()

```
public void stopRPCStream()
```

stopAudioStream()

```
public boolean stopAudioStream()
```

stopVideoStream()

```
public boolean stopVideoStream()
```

pauseAudioStream()

```
public boolean pauseAudioStream()
```

pauseVideoStream()

```
public boolean pauseVideoStream()
```

resumeAudioStream()

```
public boolean resumeAudioStream()
```

resumeVideoStream()

```
public boolean resumeVideoStream()
```

createOpenGLInputSurface(int
frameRate, int iFrameInterval, int
width, int height, int bitrate,
SessionType sType, byte
rpcSessionID)

```
public android.view.Surface createOpenGLInputSurface(int frameRate,  
int iFrameInterval,int width,int height,int bitrate,SessionType sType,byte  
rpcSessionID)
```

Parameters

frameRate
iFrameInterval
width
height

bitrate
sType
rpcSessionID

startEncoder()

```
public void startEncoder()
```

releaseEncoder()

```
public void releaseEncoder()
```

drainEncoder(boolean endOfStream)


```
public void drainEncoder(boolean endOfStream)
```

Parameters

endOfStream

sendStreamPacket(ProtocolMessage pm)

```
public void sendStreamPacket(ProtocolMessage pm)
```

Parameters

pm

setSdlSecurity(SdlSecurityBase sec)

```
public void setSdlSecurity(SdlSecurityBase sec)
```

Parameters

sec

getSdlSecurity()

```
public com.smartdevicelink.security.SdlSecurityBase getSdlSecurity()
```

startService(SessionType serviceType, byte sessionID, boolean isEncrypted)

```
public void startService(SessionType serviceType,byte sessionID,  
boolean isEncrypted)
```

Parameters

serviceType
sessionID
isEncrypted

endService(SessionType serviceType, byte sessionID)

```
public void endService(SessionType serviceType,byte sessionID)
```

Parameters

serviceType

sessionID

processControlService(ProtocolMessage msg)

```
protected void processControlService(ProtocolMessage msg)
```

Parameters

msg

getBroadcastComment(BaseTransportConfig myTransport)

```
public java.lang.String getBroadcastComment(BaseTransportConfig  
myTransport)
```

Parameters

myTransport

startSession()

```
public void startSession() throws SdlException
```

Throws

- SdlException

initialiseSession()

```
protected void initialiseSession()
```

sendMessage(ProtocolMessage msg)

```
public void sendMessage(ProtocolMessage msg)
```

Parameters

msg

getCurrentTransportType()

```
public TransportType getCurrentTransportType()
```

getIsConnected()

```
public boolean getIsConnected()
```

isServiceProtected(SessionType sType)

```
public boolean isServiceProtected(SessionType sType)
```

Parameters

sType

onTransportDisconnected(String info)

```
public void onTransportDisconnected(String info)
```

Parameters

info

onTransportDisconnected(String
info, boolean availablePrimary,
BaseTransportConfig
transportConfig)

```
public void onTransportDisconnected(String info,boolean  
availablePrimary,BaseTransportConfig transportConfig)
```

Parameters

info

availablePrimary
transportConfig

onTransportError(String info, Exception e)

```
public void onTransportError(String info,Exception e)
```

Parameters

info
e

onProtocolMessageReceived(ProtocolMessage msg)

```
public void onProtocolMessageReceived(ProtocolMessage msg)
```

Parameters

msg

onHeartbeatTimedOut(byte sessionId)

```
public void onHeartbeatTimedOut(byte sessionId)
```

Parameters

sessionID

onProtocolSessionStarted(
SessionType sessionType, byte
sessionID, byte version, String
correlationID, int hashID,
boolean isEncrypted)

```
public void onProtocolSessionStarted(SessionType sessionType,byte  
sessionID,byte version,String correlationID,int hashID,boolean  
isEncrypted)
```

Parameters

sessionType

sessionID

version
correlationID
hashID
isEncrypted

onProtocolSessionEnded(
SessionType sessionType, byte
sessionID, String correlationID)

```
public void onProtocolSessionEnded(SessionType sessionType,byte  
sessionID,String correlationID)
```

Parameters

sessionType

sessionID
correlationID

onProtocolError(String info, Exception e)

```
public void onProtocolError(String info,Exception e)
```

Parameters

info
e

sendHeartbeat(IHeartbeatMonitor monitor)

```
public void sendHeartbeat(IHeartbeatMonitor monitor)
```

Parameters

monitor

heartbeatTimedOut(IHeartbeatMonitor monitor)

```
public void heartbeatTimedOut(IHeartbeatMonitor monitor)
```

Parameters

monitor

onProtocolSessionStartedNACKed
(SessionType sessionType, byte
sessionID, byte version, String
correlationID,
java.util.List<java.lang.String>
rejectedParams)

```
public void onProtocolSessionStartedNACKed(SessionType sessionType,  
byte sessionID,byte version,String correlationID,java.util.List<java.lang.  
String> rejectedParams)
```

Parameters

sessionType

sessionID
version
correlationID
rejectedParams

onProtocolSessionEndedNACKed(
SessionType sessionType, byte
sessionID, String correlationID)

```
public void onProtocolSessionEndedNACKed(SessionType sessionType,  
byte sessionID,String correlationID)
```

Parameters

sessionType

sessionID
correlationID

onProtocolServiceDataACK(SessionType sessionType, int dataSize, byte sessionID)

```
public void onProtocolServiceDataACK(SessionType sessionType,int  
dataSize,byte sessionID)
```

Parameters

sessionType
dataSize
sessionID

onAuthTokenReceived(String authToken, byte sessionId)

```
public void onAuthTokenReceived(String authToken,byte sessionId)
```

Parameters

authToken
sessionId

onSecurityInitialized()

```
public void onSecurityInitialized()
```

clearConnection()

```
public void clearConnection()
```

checkForOpenMultiplexConnection (SdlConnection connection)

```
public void checkForOpenMultiplexConnection(SdlConnection  
connection)
```

Parameters

connection

removeConnection(SdlConnection connection)

```
public static boolean removeConnection(SdlConnection connection)
```

Parameters

connection

addServiceListener(SessionType
serviceType, ISdlServiceListener
sdlServiceListener)

```
public void addServiceListener(SessionType serviceType,  
ISdlServiceListener sdlServiceListener)
```

Parameters

serviceType

sdlServiceListener

removeServiceListener(SessionType serviceType, ISdlServiceListener sdlServiceListener)

```
public boolean removeServiceListener(SessionType serviceType,  
ISdlServiceListener sdlServiceListener)
```

Parameters

serviceType
sdlServiceListener

getServiceListeners()

```
public java.util.HashMap getServiceListeners()
```

setDesiredVideoParams(VideoStreamingParameters params)

```
public void setDesiredVideoParams(VideoStreamingParameters params)
```

Parameters

params

getDesiredVideoParams()

Returns the currently set desired video streaming parameters. If there haven't been any set, the default options will be returned and set for this instance.

```
public VideoStreamingParameters getDesiredVideoParams()
```

setAcceptedVideoParams(VideoStreamingParameters params)

```
public void setAcceptedVideoParams(VideoStreamingParameters  
params)
```

Parameters

params

getAcceptedVideoParams()

```
public VideoStreamingParameters getAcceptedVideoParams()
```

getProtocolVersion()

```
public Version getProtocolVersion()
```

isTransportForServiceAvailable(SessionType sessionType)

Check to see if a transport is available to start/use the supplied service.

```
public boolean isTransportForServiceAvailable(SessionType sessionType)
```

Parameters

sessionType

the session that should be checked for transport availability

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class SdlSession2

Hierarchy

java.lang.Object

[com.smartdevicelink.SdlConnection.SdlSession](#)

Package

[com.smartdevicelink.SdlConnection](#)

Quick Jump

Fields

- [sdlProtocol](#)

Constructors

- `SdlSession2( ISdlConnectionListener listener, MultiplexTransportConfig config )`

Methods

- `createSession(byte protocolVersion,ISdlConnectionListener listener,BaseTransportConfig btConfig)` **Deprecated**
- `getSdlConnection()` **Deprecated**
- `getMtu()`
- `getMtu(SessionType type)`
- `close()`
- `resetSession()`
- `startService(SessionType serviceType,byte sessionID,boolean isEncrypted)`
- `endService(SessionType serviceType,byte sessionID)`
- `getBroadcastComment(BaseTransportConfig myTransport)`
- `startSession()`
- `sendMessage(ProtocolMessage msg)`
- `getCurrentTransportType()`
- `getIsConnected()`
- `shutdown(String info)`
- `onTransportDisconnected(String info,boolean altTransportAvailable,BaseTransportConfig transportConfig)`
- `getProtocolVersion()`
- `onProtocolMessageBytesToSend(SdlPacket packet)`
- `onProtocolSessionStartedNACKed(SessionType sessionType,byte sessionID,byte version,String correlationID,java.util.List<java.lang.String> rejectedParams)`
- `onProtocolSessionNACKed(SessionType sessionType,byte sessionID,byte version,String correlationID,java.util.List<java.lang.String> rejectedParams)`
- `sendHeartbeat(IHeartbeatMonitor monitor)`
- `heartbeatTimedOut(IHeartbeatMonitor monitor)`
- `onHeartbeatTimedOut(byte sessionID)`
- `onProtocolHeartbeat(SessionType sessionType,byte sessionID)`
- `onProtocolHeartbeatACK(SessionType sessionType,byte sessionID)`
- `onResetOutgoingHeartbeat(SessionType sessionType,byte sessionID)`
- `onResetIncomingHeartbeat(SessionType sessionType,byte sessionID)`
- `onAuthTokenReceived(String authToken,byte sessionID)`
- `onSecurityInitialized()`
- `stopStream(SessionType serviceType)`
- `onAuthTokenReceived(String authToken)`
- `isTransportForServiceAvailable(SessionType sessionType)`
- `clearConnection()` **Deprecated**
- `removeConnection(SdlConnection connection)` **Deprecated**
- `checkForOpenMultiplexConnection(SdlConnection connection)` **Deprecated**

Fields

sdlProtocol

```
protected final com.smartdevicelink.protocol.SdlProtocol sdlProtocol
```

Inherited Fields

From Class	Fields
com.smartdevicelink.SdlConnection.SdlSession	BUFF_READ_SIZE, transportConfig, sessionListener, lockScreenMan, sdlSecurity, desiredVideoParams, acceptedVideoParams, sessionId, sessionHashId, serviceListeners, encryptedServices

Constructors

SdlSession2(
ISdlConnectionListener listener,
MultiplexTransportConfig config)

```
public SdlSession2(ISdlConnectionListener listener,  
MultiplexTransportConfig config)
```

Parameters

listener
config

Methods

createSession(byte
protocolVersion,
ISdlConnectionListener listener,
BaseTransportConfig btConfig)

Deprecated

```
public static com.smartdevicelink.SdlConnection.SdlSession2  
createSession(byte protocolVersion, ISdlConnectionListener listener,  
BaseTransportConfig btConfig)
```

Parameters

protocolVersion
listener
btConfig

getSdlConnection()

Deprecated

```
public com.smartdevicelink.SdlConnection.SdlConnection  
getSdlConnection()
```

getMtu()

```
public int getMtu()
```

getMtu(SessionType type)

```
public long getMtu(SessionType type)
```

Parameters

type

close()

```
public void close()
```

resetSession()

```
public void resetSession()
```

startService(SessionType serviceType, byte sessionID, boolean isEncrypted)

```
public void startService(SessionType serviceType,byte sessionID,  
boolean isEncrypted)
```

Parameters

serviceType
sessionID
isEncrypted

endService(SessionType serviceType, byte sessionID)

```
public void endService(SessionType serviceType,byte sessionID)
```

Parameters

serviceType
sessionID

getBroadcastComment(BaseTransportConfig myTransport)

```
public java.lang.String getBroadcastComment(BaseTransportConfig  
myTransport)
```


Parameters

```
myTransport
```

startSession()

```
public void startSession() throws SdlException
```

Throws

- SdlException

sendMessage(ProtocolMessage msg)

```
public void sendMessage(ProtocolMessage msg)
```

Parameters

msg

getCurrentTransportType()

```
public TransportType getCurrentTransportType()
```

getIsConnected()

```
public boolean getIsConnected()
```

shutdown(String info)

```
public void shutdown(String info)
```

Parameters

info

onTransportDisconnected(String
info, boolean
altTransportAvailable,
BaseTransportConfig
transportConfig)

```
public void onTransportDisconnected(String info,boolean  
altTransportAvailable,BaseTransportConfig transportConfig)
```

Parameters

info

altTransportAvailable

transportConfig

getProtocolVersion()

Get the current protocol version used by this session

```
public Version getProtocolVersion()
```

onProtocolMessageBytesToSend(SdIPacket packet)

```
public void onProtocolMessageBytesToSend(SdIPacket packet)
```

Parameters

packet

onProtocolSessionStartedNACKed
(SessionType sessionType, byte
sessionID, byte version, String
correlationID,
java.util.List<java.lang.String>
rejectedParams)

```
public void onProtocolSessionStartedNACKed(SessionType sessionType,  
byte sessionID,byte version,String correlationID,java.util.List<java.lang.  
String> rejectedParams)
```

Parameters

sessionType
sessionID
version
correlationID

rejectedParams

onProtocolSessionNACKed(
SessionType sessionType, byte
sessionID, byte version, String
correlationID,
java.util.List<java.lang.String>
rejectedParams)

```
public void onProtocolSessionNACKed(SessionType sessionType,byte  
sessionID,byte version,String correlationID,java.util.List<java.lang.  
String> rejectedParams)
```

Parameters

sessionType

sessionID

version

correlationID

rejectedParams

sendHeartbeat(IHeartbeatMonitor monitor)

```
public void sendHeartbeat(IHeartbeatMonitor monitor)
```

Parameters

monitor

heartbeatTimedOut(IHeartbeatMonitor monitor)

```
public void heartbeatTimedOut(IHeartbeatMonitor monitor)
```

Parameters

monitor

onHeartbeatTimedOut(byte sessionId)

```
public void onHeartbeatTimedOut(byte sessionId)
```

Parameters

sessionId

onProtocolHeartbeat(
SessionType sessionType, byte
sessionID)

```
public void onProtocolHeartbeat(SessionType sessionType,byte  
sessionID)
```

Parameters

sessionType
sessionID

onProtocolHeartbeatACK(
SessionType sessionType, byte
sessionID)

```
public void onProtocolHeartbeatACK(SessionType sessionType,byte  
sessionID)
```

Parameters

sessionType
sessionID

onResetOutgoingHeartbeat(
SessionType sessionType, byte
sessionID)

```
public void onResetOutgoingHeartbeat(SessionType sessionType,byte  
sessionID)
```

Parameters

sessionType
sessionID

onResetIncomingHeartbeat(SessionType sessionType, byte sessionID)

```
public void onResetIncomingHeartbeat(SessionType sessionType,byte  
sessionID)
```

Parameters

sessionType
sessionID

onAuthTokenReceived(String authToken, byte sessionID)

```
public void onAuthTokenReceived(String authToken,byte sessionID)
```

Parameters

authToken
sessionID

onSecurityInitialized()

```
public void onSecurityInitialized()
```

stopStream(SessionType serviceType)

```
public void stopStream(SessionType serviceType)
```

Parameters

serviceType

onAuthTokenReceived(String authToken)

```
public void onAuthTokenReceived(String authToken)
```

Parameters

authToken

isTransportForServiceAvailable(SessionType sessionType)

Check to see if a transport is available to start/use the supplied service.

```
public boolean isTransportForServiceAvailable(SessionType sessionType)
```

Parameters

sessionType

the session that should be checked for transport availability

clearConnection()

Deprecated

```
public void clearConnection()
```

removeConnection(SdlConnection connection)

Deprecated

```
public static boolean removeConnection(SdlConnection connection)
```

Parameters

connection

checkForOpenMultiplexConnection (SdlConnection connection)

Deprecated

```
public void checkForOpenMultiplexConnection(SdlConnection  
connection)
```

Parameters

connection

Inherited Methods

From Class	Methods
com.smartdevicelink.SdlConnection.SdlSession	createSession, getTransportConfig, getLockScreenMan, getOutgoingHeartbeatMonitor, getIncomingHeartbeatMonitor, setOutgoingHeartbeatMonitor, setIncomingHeartbeatMonitor, getSessionHashId, getSessionId, getSdlConnection, getMtu, getMtu, close, resetSession, startStream, startStream, startVideoStream, startAudioStream, startRPCStream, startRPCStream, pauseRPCStream, resumeRPCStream, stopRPCStream, stopAudioStream, stopVideoStream, pauseAudioStream, pauseVideoStream, resumeAudioStream, resumeVideoStream, createOpenGLInputSurface, startEncoder, releaseEncoder, drainEncoder, sendStreamPacket, setSdlSecurity, getSdlSecurity, startService, endService, processControlService, getBroadcastComment, startSession, initialiseSession, sendMessage, getCurrentTransportType, getIsConnected, isServiceProtected, onTransportDisconnected, onTransportDisconnected, onTransportError, onProtocolMessageReceived, onHeartbeatTimedOut, onProtocolSessionStarted, onProtocolSessionEnded, onProtocolError, sendHeartbeat, heartbeatTimedOut, onProtocolSessionStartedNACKed, onProtocolSessionEndedNACKed, onProtocolServiceDataACK, onAuthTokenReceived, onSecurityInitialized, clearConnection, checkForOpenMultiplexConnection, removeConnection, addServiceListener, removeServiceListener, getServiceListeners, setDesiredVideoParams, getDesiredVideoParams, setAcceptedVideoParams, getAcceptedVideoParams, getProtocolVersion, isTransportForServiceAvailable
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class SdlTrace

Hierarchy

java.lang.Object

Package

com.smartdevicelink.trace

Quick Jump

Fields

- [SYSTEM_LOG_TAG](#)
- [m_appTraceListener](#)

Constructors

- [SdlTrace\(\)](#)

Methods

- [setAcceptAPITraceAdjustments\(Boolean APITraceAdjustmentsAccepted\)](#)
- [getAcceptAPITraceAdjustments\(\)](#)
- [setAppTraceListener\(ISTListener listener\)](#)
- [setAppTraceLevel\(DetailLevel dt\)](#)
- [setProxyTraceLevel\(DetailLevel dt\)](#)
- [setRpcTraceLevel\(DetailLevel dt\)](#)
- [setMarshallingTraceLevel\(DetailLevel dt\)](#)

- setProtocolTraceLevel(DetailLevel dt)
- setTransportTraceLevel(DetailLevel dt)
- logProxyEvent(String eventText,String token)
- logAppEvent(String eventText)
- logRPCEvent(InterfaceActivityDirection msgDirection,RPCMessage rpcMsg,String token)
- logMarshallingEvent(InterfaceActivityDirection msgDirection,byte [] marshalledMessage,String token)
- logProtocolEvent(InterfaceActivityDirection frameDirection,SdIPacket packet,int frameDataOffset,int frameDataLength,String token)
- getBTDeviceInfo(BluetoothDevice btDevice)
- logTransportEvent(String preamble,String transportSpecificInfoXml,InterfaceActivityDirection msgDirection,byte[] buf,int byteLength,String token)
- logTransportEvent(String preamble,String transportSpecificInfoXml,InterfaceActivityDirection msgDirection,byte[] buf,int offset,int byteLength,String token)
- writeMessageToSiphonServer(String info)
- getLogHeader(String dumpReason,int seqNo)

Fields

SYSTEM_LOG_TAG

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

Constant Value

"SdlTrace"

m_appTraceListener

```
protected static ISTListener m_appTraceListener
```

Constructors

SdlTrace()

```
public SdlTrace()
```

Methods

setAcceptAPITraceAdjustments(
Boolean
APITraceAdjustmentsAccepted)

```
public static void setAcceptAPITraceAdjustments(Boolean  
APITraceAdjustmentsAccepted)
```

Parameters

APITraceAdjustmentsAccepted

getAcceptAPITraceAdjustments()

```
public static boolean getAcceptAPITraceAdjustments()
```

setAppTraceListener(ISTListener listener)

```
public static void setAppTraceListener(ISTListener listener)
```

Parameters

listener

setAppTraceLevel(DetailLevel dt)

```
public static void setAppTraceLevel(DetailLevel dt)
```

Parameters

dt

setProxyTraceLevel(DetailLevel dt)

```
public static void setProxyTraceLevel(DetailLevel dt)
```

Parameters

dt

setRpcTraceLevel(DetailLevel dt)

```
public static void setRpcTraceLevel(DetailLevel dt)
```

Parameters

dt

setMarshallingTraceLevel(DetailLevel dt)

```
public static void setMarshallingTraceLevel(DetailLevel dt)
```

Parameters

dt

setProtocolTraceLevel(DetailLevel dt)

```
public static void setProtocolTraceLevel(DetailLevel dt)
```


Parameters

dt

setTransportTraceLevel(
DetailLevel dt)

```
public static void setTransportTraceLevel(DetailLevel dt)
```

Parameters

dt

logProxyEvent(String eventText,
String token)

```
public static boolean logProxyEvent(String eventText,String token)
```

Parameters

eventText
token

logAppEvent(String eventText)

```
public static boolean logAppEvent(String eventText)
```

Parameters

eventText

logRPCEvent(InterfaceActivityDirection msgDirection, RPCMessage rpcMsg, String token)

```
public static boolean logRPCEvent(InterfaceActivityDirection  
msgDirection,RPCMessage rpcMsg,String token)
```

Parameters

msgDirection
rpcMsg
token

logMarshallingEvent(InterfaceActivityDirection msgDirection, byte[] marshalledMessage, String token)

```
public static boolean logMarshallingEvent(InterfaceActivityDirection  
msgDirection,byte[] marshalledMessage,String token)
```

Parameters

msgDirection
marshalledMessage
token

logProtocolEvent(InterfaceActivityDirection frameDirection, SdlPacket packet, int frameDataOffset, int frameDataLength, String token)

```
public static boolean logProtocolEvent(InterfaceActivityDirection  
frameDirection,SdlPacket packet,int frameDataOffset,int  
frameDataLength,String token)
```

Parameters

frameDirection
packet
frameDataOffset
frameDataLength

token

getBTDeviceInfo(BluetoothDevice btDevice)

```
public static java.lang.String getBTDeviceInfo(BluetoothDevice  
btDevice)
```

Parameters

btDevice

logTransportEvent(String
preamble, String
transportSpecificInfoXml,
InterfaceActivityDirection
msgDirection, byte[] buf, int
byteLength, String token)

```
public static boolean logTransportEvent(String preamble,String
transportSpecificInfoXml,InterfaceActivityDirection msgDirection,byte[]
buf,int byteLength,String token)
```

Parameters

preamble
transportSpecificInfoXml
msgDirection
buf
byteLength
token

logTransportEvent(String
preamble, String
transportSpecificInfoXml,
InterfaceActivityDirection
msgDirection, byte[] buf, int
offset, int byteLength, String
token)

```
public static boolean logTransportEvent(String preamble,String  
transportSpecificInfoXml,InterfaceActivityDirection msgDirection,byte[]  
buf,int offset,int byteLength,String token)
```

Parameters

preamble
transportSpecificInfoXml
msgDirection

buf
offset
byteLength
token

writeMessageToSiphonServer(String info)

```
public static java.lang.Boolean writeMessageToSiphonServer(String info )
```

Parameters

info

getLogHeader(String dumpReason, int seqNo)

```
public static java.lang.String getLogHeader(String dumpReason,int  
seqNo)
```

Parameters

dumpReason

seqNo

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class SdlTransport

Hierarchy

java.lang.Object

Package

com.smartdevicelink.transport

All Known Subclasses

com.smartdevicelink.transport.BTTransport,
com.smartdevicelink.transport.MultiplexTransport,
com.smartdevicelink.transport.TCPTransport,
com.smartdevicelink.transport.USBTransport

Quick Jump

Constructors

- [SdlTransport\(ITransportListener transportListener \)](#)

Methods

- [getIsConnected\(\)](#)
- [handleReceivedPacket\(SdlPacket packet\)](#)
- [sendBytesOverTransport\(SdlPacket packet\)](#)
- [sendBytes\(SdlPacket packet\)](#)
- [handleTransportConnected\(\)](#)
- [handleTransportDisconnected\(String info\)](#)
- [handleTransportError\(String message,Exception ex\)](#)

- `openConnection()`
- `disconnect()`
- `getTransportType()`
- `getBroadcastComment()`

Constructors

SdlTransport(ITransportListener transportListener)

```
protected SdlTransport(ITransportListener transportListener)
```

Parameters

transportListener

Methods

getIsConnected()

```
public java.lang.Boolean getIsConnected()
```

handleReceivedPacket(SdlPacket packet)

```
protected void handleReceivedPacket(SdlPacket packet)
```

Parameters

packet

sendBytesOverTransport(SdlPacket packet)

```
protected abstract boolean sendBytesOverTransport(SdlPacket packet)
```

Parameters

packet

sendBytes(SdIPacket packet)

```
public boolean sendBytes(SdIPacket packet)
```

Parameters

packet

handleTransportConnected()

```
protected void handleTransportConnected()
```

handleTransportDisconnected(String info)

```
protected void handleTransportDisconnected(String info)
```

Parameters

info

handleTransportError(String message, Exception ex)

```
protected void handleTransportError(String message,Exception ex)
```

Parameters

message
ex

openConnection()

```
public abstract void openConnection() throws SdlException
```

Throws

- SdlException

disconnect()

```
public abstract void disconnect()
```


getTransportType()

Abstract method which should be implemented by subclasses in order to return actual type of the transport.

```
public abstract TransportType getTransportType()
```

getBroadcastComment()

```
public abstract java.lang.String getBroadcastComment()
```

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class ServiceFinder

Hierarchy

java.lang.Object

Package

com.smartdevicelink.util

Overview

Created by Joey Grover on 8/18/17.

Quick Jump

Fields

- [TAG](#)

Constructors

- [ServiceFinder\(Context context, String packageName, ServiceFinder.ServiceFinderCallback callback \)](#)

Fields

TAG

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

Constructors

ServiceFinder(Context context,
String packageName,
ServiceFinder.ServiceFinderCallback
callback)

```
public ServiceFinder(Context context,String packageName,  
ServiceFinder.ServiceFinderCallback callback)
```

Parameters

context
packageName
callback

Methods

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Interface ServiceFinder.ServiceFinderCallbac

Package

com.smartdevicelink.util

Quick Jump

Methods

- `onComplete(java.util.Vector<android.content.ComponentName> routerServices)`

Methods

`onComplete(
java.util.Vector<android.content.Comp
routerServices)`

```
public void onComplete(java.util.Vector<android.content.  
ComponentName> routerServices)
```

Parameters

routerServices

Class StreamPacketizer

Hierarchy

java.lang.Object

AbstractPacketizer

Package

com.smartdevicelink.streaming

All Implemented Interfaces

java.lang.Runnable

Quick Jump

Fields

- TAG
- sdlConnection

Constructors

- StreamPacketizer(IStreamListener streamListener, InputStream is, SessionType sType, byte rpcSessionID, SdlSession session)

Methods

- start()
- stop()
- run()
- pause()
- resume()
- sendFrame(byte[] data,int offset,int length,long presentationTimeUs)
- sendFrame(ByteBuffer data,long presentationTimeUs)
- sendAudio(byte[] data,int offset,int length,long presentationTimeUs)

- `sendAudio(ByteBuffer data,long presentationTimeUs)`

Fields

TAG

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

Constant Value

"StreamPacketizer"

sdlConnection

```
public com.smartdevicelink.SdlConnection.SdlConnection sdlConnection
```

Constructors

StreamPacketizer(
IStreamListener streamListener,
InputStream is, SessionType
sType, byte rpcSessionID,
SdlSession session)

```
public StreamPacketizer(IStreamListener streamListener,InputStream is,  
SessionType sType,byte rpcSessionID,SdlSession session)
```

Parameters

streamListener
is
sType
rpcSessionID

session

Methods

start()

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

Throws

- java.io.IOException

stop()

```
public void stop()
```

run()

```
public void run()
```

pause()

```
public void pause()
```

resume()

```
public void resume()
```

sendFrame(byte[] data, int offset, int length, long presentationTimeUs)

Called by the app.

```
public void setFrame(byte[] data,int offset,int length,long  
presentationTimeUs) throws java.lang.  
ArrayIndexOutOfBoundsException
```

Parameters

data
offset
length
presentationTimeUs

Throws

- java.lang.ArrayIndexOutOfBoundsException

sendFrame(ByteBuffer data, long presentationTimeUs)

Called by the app.

```
public void sendFrame(ByteBuffer data,long presentationTimeUs)
```

Parameters

data
presentationTimeUs

sendAudio(byte[] data, int offset, int length, long presentationTimeUs)

Called by the app.

```
public void sendAudio(byte[] data,int offset,int length,long  
presentationTimeUs) throws java.lang.  
ArrayIndexOutOfBoundsException
```

Parameters

data
offset
length
presentationTimeUs

Throws

- java.lang.ArrayIndexOutOfBoundsException

sendAudio(ByteBuffer data, long presentationTimeUs)

Called by the app.

```
public void sendAudio(ByteBuffer data,long presentationTimeUs)
```

Parameters

data

presentationTimeUs

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class StreamRPCPacketizer

Hierarchy

java.lang.Object

AbstractPacketizer

Package

com.smartdevicelink.streaming

All Implemented Interfaces

java.lang.Runnable

Quick Jump

Constructors

- [StreamRPCPacketizer\(
com.smartdevicelink.proxy.SdlProxyBase<IProxyListenerBase>
proxy, IStreamListener streamListener, InputStream is,
RPCRequest request, SessionType sType, byte rpcSessionID, byte
wiproVersion, long lLength, SdlSession session \)](#)
- [StreamRPCPacketizer\(
com.smartdevicelink.proxy.SdlProxyBase<IProxyListenerBase>
proxy, IStreamListener streamListener, InputStream is,
RPCRequest request, SessionType sType, byte rpcSessionID,
Version wiproVersion, Version rpcSpecVersion, long lLength,
SdlSession session \)](#)

Methods

- start()
- stop()
- pause()
- resume()
- run()
- onPutFileResponse(PutFileResponse response)
- onPutFileStreamError(Exception e,String info)

Constructors

StreamRPCPacketizer(
com.smartdevicelink.proxy.SdlProxyBase
proxy, IStreamListener
streamListener, InputStream is,
RPCRequest request,
SessionType sType, byte
rpcSessionID, byte wiproVersion,
long lLength, SdlSession session
)

```
public StreamRPCPacketizer(com.smartdevicelink.proxy.SdlProxyBase<
IProxyListenerBase> proxy,IStreamListener streamListener,InputStream
is,RPCRequest request,SessionType sType,byte rpcSessionID,byte
wiproVersion,long lLength,SdlSession session)
```

Parameters

proxy
streamListener
is
request
sType
rpcSessionID
wiproVersion
lLength

session

StreamRPCPacketizer(
com.smartdevicelink.proxy.SdlProxyBase
proxy, IStreamListener
streamListener, InputStream is,
RPCRequest request,
SessionType sType, byte
rpcSessionID, Version
wiproVersion, Version
rpcSpecVersion, long lLength,
SdlSession session)

```
public StreamRPCPacketizer(com.smartdevicelink.proxy.SdlProxyBase<  
IProxyListenerBase> proxy,IStreamListener streamListener,InputStream  
is,RPCRequest request,SessionType sType,byte rpcSessionID,Version  
wiproVersion,Version rpcSpecVersion,long lLength,SdlSession session)
```

Parameters

proxy

streamListener
is
request
sType
rpcSessionID
wiproVersion
rpcSpecVersion
lLength
session

Methods

start()

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

Throws

- java.io.IOException

stop()

```
public void stop()
```

pause()

```
public void pause()
```

resume()

```
public void resume()
```

run()

```
public void run()
```

onPutFileResponse(PutFileResponse response)

```
public void onPutFileResponse(PutFileResponse response)
```

Parameters

response

onPutFileStreamError(Exception e, String info)

```
public void onPutFileStreamError(Exception e,String info)
```

Parameters

e

info

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class TCPTransport

Hierarchy

java.lang.Object

[com.smartdevicelink.transport.SdlTransport](#)

Package

com.smartdevicelink.transport

Overview

Class that implements TCP transport

Quick Jump

Constructors

- [TCPTransport\(TCPTransportConfig tcpTransportConfig, ITransportListener transportListener \)](#)

Methods

- [sendBytesOverTransport\(SdlPacket packet\)](#)
- [openConnection\(\)](#)
- [disconnect\(\)](#)
- [getTransportType\(\)](#)
- [logInfo\(String message\)](#)

- `logError(String message)`
- `logWarning(String message)`
- `logError(String message,Throwable throwable)`
- `getBroadcastComment()`

Constructors

`TCPTransport(
TCPTransportConfig
tcpTransportConfig,
ITransportListener
transportListener)`

Constructs TCP transport component instance

```
public TCPTransport(TCPTransportConfig tcpTransportConfig,  
ITransportListener transportListener)
```

Parameters

`tcpTransportConfig`

Instance of the TCP transport configuration

transportListener

Listener that will be notified on different TCP transport activities

Methods

sendBytesOverTransport(SdIPacket packet)

Performs actual work of sending array of bytes over the transport

```
protected boolean sendBytesOverTransport(SdIPacket packet)
```

Parameters

packet

openConnection()

Tries to open connection to SmartDeviceLinkCore. Actual try will be performed only if no actual connection is available

```
public void openConnection() throws SdlException
```

Throws

- SdlException

disconnect()

Tries to disconnect from SmartDeviceLinkCore. Actual try will be performed only if connection is available

```
public void disconnect()
```

getTransportType()

Overridden abstract method which returns specific type of this transport.

```
public TransportType getTransportType()
```

logInfo(String message)

Internal method for logging information messages

```
protected void logInfo(String message)
```

Parameters

message

Message to log

logError(String message)

Internal method for logging error messages

```
protected void logError(String message)
```

Parameters

message

Message to log

logWarning(String message)

Internal method for logging warning messages

```
protected void logWarning(String message)
```

Parameters

message

Message to log

logError(String message, Throwable throwable)

Internal method for logging error message together with information about exception that was the reason of it

```
protected void logError(String message,Throwable throwable)
```

Parameters

message

Message to log

throwable

Exception, that was the main reason for logged error message

getBroadcastComment()

```
public java.lang.String getBroadcastComment()
```

Inherited Methods

From Class	Methods
<code>com.smartdevicelink.transport.SdlTransport</code>	<code>getIsConnected</code> , <code>handleReceivedPacket</code> , <code>sendBytesOverTransport</code> , <code>sendBytes</code> , <code>handleTransportConnected</code> , <code>handleTransportDisconnected</code> , <code>handleTransportError</code> , <code>openConnection</code> , <code>disconnect</code> , <code>getTransportType</code> , <code>getBroadcastComment</code>
<code>java.lang.Object</code>	<code>getClass</code> , <code>hashCode</code> , <code>equals</code> , <code>clone</code> , <code>toString</code> , <code>notify</code> , <code>notifyAll</code> , <code>wait</code> , <code>wait</code> , <code>wait</code> , <code>finalize</code>

Class TCPTransportConfig

Hierarchy

`java.lang.Object`

`BaseTransportConfig`

Package

`com.smartdevicelink.transport`

Overview

Container of TCP transport specific configuration.

Quick Jump

Constructors

- `TCPTransportConfig( int port, String ipAddress, boolean autoReconnect )`

Methods

- `getPort()`
- `getIPAddress()`
- `getAutoReconnect()`
- `getTransportType()`
- `toString()`

Constructors

TCPTransportConfig(int port, String ipAddress, boolean autoReconnect)

Constructor. Objects of this class must be created for known port and IP address value.

```
public TCPTransportConfig(int port,String ipAddress,boolean autoReconnect)
```


Parameters

port

Port for TCP connection.

ipAddress

IP address for TCP connection.

autoReconnect

Flag which must be set to true if tcp connection must be automatically reestablished in case of disconnection

Methods

getPort()

Gets value of Port.

```
public int getPort()
```

getIPAddress()

Gets value of IP address.

```
public java.lang.String getIPAddress()
```

getAutoReconnect()

Gets value of AutoReconnect

```
public boolean getAutoReconnect()
```

getTransportType()

Overridden abstract method which returns specific type of this transport configuration.

```
public TransportType getTransportType()
```

toString()

```
public java.lang.String toString()
```

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class TraceDeviceInfo

Hierarchy

java.lang.Object

Package

com.smartdevicelink.trace

Quick Jump

Constructors

- `TraceDeviceInfo( TelephonyManager telephonyManager )`

Methods

- `setTelephonyManager(TelephonyManager telephonyManager)`
- `getTelephonyManager()`

Constructors

`TraceDeviceInfo(
TelephonyManager
telephonyManager)`

```
public TraceDeviceInfo(TelephonyManager telephonyManager)
```

Parameters

telephonyManager

Methods

setTelephonyManager(
TelephonyManager
telephonyManager)

```
public static void setTelephonyManager(TelephonyManager  
telephonyManager)
```

Parameters

telephonyManager

getTelephonyManager()

```
public static android.telephony.TelephonyManager  
getTelephonyManager()
```

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class TransportBroker

Hierarchy

java.lang.Object

Package com.smartdevicelink.transport
All Known Subclasses

Quick Jump

Constructors

- `TransportBroker( Context context, String appld, ComponentName service )`

Methods

- `sendMessageToRouterService(Message message)`
- `sendMessageToRouterService(Message message,int retryCount)`
- `start()`
- `resetSession()`
- `stop()`
- `onServiceUnregsiteredFromRouterService(int unregisterCode)`
- `onHardwareDisconnected(TransportType type) Deprecated`
- `onHardwareDisconnected(TransportRecord record,java.util.List<TransportRecord> connectedTransports)`
- `onHardwareConnected(TransportType type) Deprecated`
- `onHardwareConnected(java.util.List<TransportRecord> transports)`
- `onPacketReceived(Parcelable packet)`
- `onLegacyModeEnabled()`
- `getRouterServiceVersion()`
- `sendPacketToRouterService(SdIPacket packet)`
- `getRouterService()`
- `convertAppld(String appld)`
- `enableLegacyMode(boolean enable)`
- `isLegacyModeEnabled()`
- `requestNewSession() Deprecated`
- `requestNewSession(TransportRecord transportRecord)`
- `requestSecondaryTransportConnection(byte sessionId,Bundle bundle)`
- `removeSession(long sessionId)`

Constructors

TransportBroker(Context context, String appld, ComponentName service)

Life Cycle *****

```
public TransportBroker(Context context,String appld,ComponentName  
service)
```

Parameters

context
appld
service

Methods

sendMessageToRouterService(Message message)

```
protected synchronized boolean sendMessageToRouterService(Message  
message)
```

Parameters

message

sendMessageToRouterService(Message message, int retryCount)

```
protected synchronized boolean sendMessageToRouterService(Message  
message,int retryCount)
```

Parameters

message
retryCount

start()

This beings the initial connection with the router service.

```
public boolean start()
```

resetSession()

```
public void resetSession()
```

stop()

This method will end our communication with the router service.

```
public void stop()
```

onServiceUnregisteredFromRouterService (int unregisterCode)

Event Callbacks *****

```
public void onServiceUnregisteredFromRouterService(int  
unregisterCode)
```

Parameters

unregisterCode

onHardwareDisconnected(TransportType type)

Deprecated

```
public void onHardwareDisconnected(TransportType type)
```

Parameters

type

onHardwareDisconnected(TransportRecord record, java.util.List<TransportRecord> connectedTransports)

```
public void onHardwareDisconnected(TransportRecord record,java.util.  
List<TransportRecord> connectedTransports)
```

Parameters

record

connectedTransports

onHardwareConnected(TransportType type)

Deprecated

WILL NO LONGER BE CALLED

```
public boolean onHardwareConnected(TransportType type)
```

Parameters

type

onHardwareConnected(java.util.List<TransportRecord> transports)

```
public boolean onHardwareConnected(java.util.List<TransportRecord>  
transports)
```

Parameters

transports

onPacketReceived(Parcelable packet)

```
public void onPacketReceived(Parcelable packet)
```

Parameters

packet

onLegacyModeEnabled()

```
public void onLegacyModeEnabled()
```

getRouterServiceVersion()

```
protected int getRouterServiceVersion()
```

sendPacketToRouterService(SdIPacket packet)

```
public boolean sendPacketToRouterService(SdIPacket packet)
```

Parameters

packet

getRouterService()

```
protected android.content.ComponentName getRouterService()
```

convertAppId(String appId)

```
public static java.lang.Long convertAppId(String appId)
```

Parameters

appId

enableLegacyMode(boolean enable)

```
protected void enableLegacyMode(boolean enable)
```

Parameters

enable

isLegacyModeEnabled()

```
protected static boolean isLegacyModeEnabled()
```

requestNewSession()

Deprecated

Use this method to let the router service know that you are requesting a new session from the head unit.

```
public void requestNewSession()
```

requestNewSession(TransportRecord transportRecord)

```
public void requestNewSession(TransportRecord transportRecord)
```

Parameters

transportRecord

requestSecondaryTransportConnection (byte sessionId, Bundle bundle)

Request secondary transport and communicate details to router service

```
public void requestSecondaryTransportConnection(byte sessionId,  
Bundle bundle)
```

Parameters

sessionId
bundle

removeSession(long sessionId)

```
public void removeSession(long sessionId)
```

Parameters

sessionId

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class TransportManager

Hierarchy

java.lang.Object

TransportManagerBase

Package

com.smartdevicelink.transport

Quick Jump

Constructors

- [TransportManager\(MultiplexTransportConfig config, TransportEventListener listener \)](#)

Methods

- `start()`
- `close(long sessionId)`
- `resetSession()` **Deprecated**
- `isConnected(TransportType transportType,String address)`
- `getTransportRecord(TransportType transportType,String address)`
- `getConnectedTransports()`
- `isHighBandwidthAvailable()`
- `updateTransportConfig(BaseTransportConfig config)`
- `getRouterService()` **Deprecated**
- `sendPacket(SdlPacket packet)`
- `requestNewSession(TransportRecord transportRecord)`
- `requestSecondaryTransportConnection(byte sessionId,Bundle params)` **Deprecated**
- `requestSecondaryTransportConnection(byte sessionId,TransportRecord transportRecord)`

Constructors

TransportManager(MultiplexTransportConfig config, TransportEventListener listener)

Managing transports List for status of all transports If transport is not connected. Request Router service connect to it. Get connected message

```
public TransportManager(MultiplexTransportConfig config,  
    TransportEventListener listener)
```

Parameters

config
listener

Methods

start()

```
public void start()
```

close(long sessionId)

```
public void close(long sessionId)
```

Parameters

sessionId

resetSession()

Deprecated

```
public void resetSession()
```

isConnected(TransportType transportType, String address)

Check to see if a transport is connected.

```
public boolean isConnected(TransportType transportType,String address  
)
```

Parameters

transportType

the transport to have its connection status returned. If `null` is passed in, all transports will be checked and if any are connected a true value will be returned.

address

the address associated with the transport type. If null, the first transport of supplied type will be used to return if connected.

getTransportRecord(TransportType transportType, String address)

Retrieve a transport record with the supplied params

```
public TransportRecord getTransportRecord(TransportType transportType  
,String address)
```


Parameters

transportType

the transport to have its connection status returned.

address

the address associated with the transport type. If null, the first transport of supplied type will be returned.

getConnectedTransports()

Retrieves the currently connected transports

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

isHighBandwidthAvailable()

```
public boolean isHighBandwidthAvailable()
```

updateTransportConfig(BaseTransportConfig config)

```
public BaseTransportConfig updateTransportConfig(BaseTransportConfig config)
```

Parameters

config

getRouterService()

Deprecated

```
public android.content.ComponentName getRouterService()
```

sendPacket(SdlPacket packet)

```
public void sendPacket(SdlPacket packet)
```

Parameters

packet

requestNewSession(TransportRecord transportRecord)

```
public void requestNewSession(TransportRecord transportRecord)
```

Parameters

transportRecord

requestSecondaryTransportConnection (byte sessionId, Bundle params)

Deprecated

```
public void requestSecondaryTransportConnection(byte sessionId,  
Bundle params)
```

Parameters

sessionId
params

requestSecondaryTransportConnection (byte sessionId, TransportRecord transportRecord)

```
public void requestSecondaryTransportConnection(byte sessionId,  
TransportRecord transportRecord)
```

Parameters

sessionId

transportRecord

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class

TransportManager.LegacyBluetooth

Hierarchy

java.lang.Object

android.os.Handler

Package

com.smartdevicelink.transport

Quick Jump

Constructors

- [LegacyBluetoothHandler\(TransportManager provider \)](#)

Methods

- [handleMessage\(Message msg\)](#)

Constructors

LegacyBluetoothHandler(TransportManager provider)

```
public LegacyBluetoothHandler(TransportManager provider)
```

Parameters

provider

Methods

handleMessage(Message msg)

```
public void handleMessage(Message msg)
```

Parameters

msg

Inherited Methods

From Class	Methods
android.os.Handler	handleMessage, dispatchMessage, getMessageName, obtainMessage, obtainMessage, obtainMessage, obtainMessage, post, postAtTime, postAtTime, postDelayed, postAtFrontOfQueue, removeCallbacks, removeCallbacks, sendMessage, sendEmptyMessage, sendEmptyMessageDelayed, sendEmptyMessageAtTime, sendMessageDelayed, sendMessageAtTime, sendMessageAtFrontOfQueue, removeMessages, removeMessages, removeCallbacksAndMessages, hasMessages, hasMessages, getLooper, dump, toString
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class

TransportManager.TransportBrokerI

Hierarchy

java.lang.Object

com.smartdevicelink.transport.TransportBroker

Package

com.smartdevicelink.transport

Quick Jump

Constructors

- [TransportBrokerImpl\(Context context, String appld, ComponentName routerService \)](#)

Methods

- [onHardwareConnected\(TransportType transportType\)](#) **Deprecated**
- [onHardwareConnected\(java.util.List<TransportRecord> transports\)](#)
- [onHardwareDisconnected\(TransportRecord record,java.util.List<TransportRecord> connectedTransports\)](#)
- [onPacketReceived\(Parcelable packet\)](#)

Constructors

TransportBrokerImpl(Context
context, String appld,
ComponentName routerService)

```
public TransportBrokerImpl(Context context,String appld,  
ComponentName routerService)
```

Parameters

context
appld
routerService

Methods

onHardwareConnected(TransportType transportType)

Deprecated

```
public boolean onHardwareConnected(TransportType transportType)
```

Parameters

transportType

onHardwareConnected(java.util.List<TransportRecord> transports)

```
public boolean onHardwareConnected(java.util.List<TransportRecord>  
transports)
```

Parameters

transports

onHardwareDisconnected(
TransportRecord record,
java.util.List<TransportRecord>
connectedTransports)

```
public void onHardwareDisconnected(TransportRecord record,java.util.  
List<TransportRecord> connectedTransports)
```

Parameters

record

connectedTransports

onPacketReceived(Parcelable packet)

```
public void onPacketReceived(Parcelable packet)
```

Parameters

packet

Inherited Methods

From Class	Methods
com.smartdevicelink.transport.TransportBroker	sendMessageToRouterService, sendMessageToRouterService, start, resetSession, stop, onServiceUnregisteredFromRouterService, onHardwareDisconnected, onHardwareDisconnected, onHardwareConnected, onHardwareConnected, onPacketReceived, onLegacyModeEnabled, getRouterServiceVersion, sendPacketToRouterService, getRouterService, convertAppId, enableLegacyMode, isLegacyModeEnabled, requestNewSession, requestNewSession, requestSecondaryTransportConnection, removeSession
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class USBAccessoryAttachmentActivity

Hierarchy

java.lang.Object

android.content.Context

android.content.ContextWrapper

android.view.ContextThemeWrapper

android.app.Activity

Package

com.smartdevicelink.transport

Overview

The USBAccessoryAttachmentActivity is a proxy to listen for USB_ACCESSORY_ATTACHED intents.

Unfortunately, the USB_ACCESSORY_ATTACHED intent can only be sent to an activity. So this class is a workaround to get that intent.

Some reference: <http://stackoverflow.com/questions/6981736/android-3-1-usb-host-broadcastreceiver-does-not-receive-usb-device-attached/9814826#9814826>

Inspired by OpenXC-Android: <https://github.com/openxc/openxc-android>

NOTE: An application that wants to use USB transport must make the following changes to AndroidManifest.xml:

1. Add these lines to the {@literal ...} scope:

```
{@code
```

```
}
```

2. Add these lines to the {@literal ...} scope:

```
{@code
```

```
}
```

3. Set minimum SDK version to 12:

```
{@code }
```

Quick Jump

Constructors

- [USBAccessoryAttachmentActivity\(\)](#)

Methods

- [onCreate\(Bundle savedInstanceState\)](#)
- [onResume\(\)](#)
- [onNewIntent\(Intent intent\)](#)

- onDestroy()

Fields

Inherited Fields

From Class	Fields
android.app.Activity	DEFAULT_KEYS_DIALER, DEFAULT_KEYS_DISABLE, DEFAULT_KEYS_SEARCH_GLOBAL, DEFAULT_KEYS_SEARCH_LOCAL, DEFAULT_KEYS_SHORTCUT, FOCUSED_STATE_SET, RESULT_CANCELED, RESULT_FIRST_USER, RESULT_OK

android.content.Context

ACCESSIBILITY_SERVICE,
ACCOUNT_SERVICE, ACTIVITY_SERVICE,
ALARM_SERVICE, APPWIDGET_SERVICE,
APP_OPS_SERVICE, AUDIO_SERVICE,
BATTERY_SERVICE, BIND_ABOVE_CLIENT,
BIND_ADJUST_WITH_ACTIVITY,
BIND_ALLOW_OOM_MANAGEMENT,
BIND_AUTO_CREATE,
BIND_DEBUG_UNBIND, BIND_IMPORTANT,
BIND_NOT_FOREGROUND,
BIND_WAIVE_PRIORITY,
BLUETOOTH_SERVICE, CAMERA_SERVICE,
CAPTIONING_SERVICE,
CARRIER_CONFIG_SERVICE,
CLIPBOARD_SERVICE,
CONNECTIVITY_SERVICE,
CONSUMER_IR_SERVICE,
CONTEXT_IGNORE_SECURITY,
CONTEXT_INCLUDE_CODE,
CONTEXT_RESTRICTED,
DEVICE_POLICY_SERVICE,
DISPLAY_SERVICE, DOWNLOAD_SERVICE,
DROPBOX_SERVICE,
FINGERPRINT_SERVICE,
INPUT_METHOD_SERVICE,
INPUT_SERVICE,
JOB_SCHEDULER_SERVICE,
KEYGUARD_SERVICE,
LAUNCHER_APPS_SERVICE,
LAYOUT_INFLATER_SERVICE,
LOCATION_SERVICE,
MEDIA_PROJECTION_SERVICE,
MEDIA_ROUTER_SERVICE,
MEDIA_SESSION_SERVICE, MIDI_SERVICE,
MODE_APPEND,
MODE_ENABLE_WRITE_AHEAD_LOGGING,
MODE_MULTI_PROCESS, MODE_PRIVATE,
MODE_WORLD_READABLE,
MODE_WORLD_WRITEABLE,
NETWORK_STATS_SERVICE,
NFC_SERVICE, NOTIFICATION_SERVICE,
NSD_SERVICE, POWER_SERVICE,
PRINT_SERVICE, RESTRICTIONS_SERVICE,
SEARCH_SERVICE, SENSOR_SERVICE,
STORAGE_SERVICE, TELECOM_SERVICE,
TELEPHONY_SERVICE,
TELEPHONY_SUBSCRIPTION_SERVICE

Constructors

USBAccessoryAttachmentActivity ()

```
public USBAccessoryAttachmentActivity()
```

Methods

onCreate(Bundle savedInstanceState)

```
protected void onCreate(Bundle savedInstanceState)
```

Parameters

savedInstanceState

onResume()

```
protected void onResume()
```

onNewIntent(Intent intent)

```
protected void onNewIntent(Intent intent)
```

Parameters

intent

onDestroy()

```
protected void onDestroy()
```

Inherited Methods

From Class

Methods

getIntent, setIntent, getApplication,
isChild, getParent, getWindowManager,
getWindow, getLoaderManager,
getCurrentFocus, onCreate, onCreate,
onRestoreInstanceState,
onRestoreInstanceState, onPostCreate,
onPostCreate, onStart, onRestart,
onStateNotSaved, onResume,
onPostResume, isVoiceInteraction,
isVoiceInteractionRoot,
getVoiceInteractor, onNewIntent,
onSaveInstanceState,
onSaveInstanceState, onPause,
onUserLeaveHint, onCreateThumbnail,
onCreateDescription,
onProvideAssistData,
onProvideAssistContent, showAssist,
onStop, onDestroy, reportFullyDrawn,
onConfigurationChanged,
getChangingConfigurations,
getLastNonConfigurationInstance,
onRetainNonConfigurationInstance,
onLowMemory, onTrimMemory,
getFragmentManager,
onAttachFragment, managedQuery,
startManagingCursor,
stopManagingCursor, findViewById,
getActionBar, setActionBar,
setContentView, setContentView,
setContentView, addContentView,
getContentTransitionManager,
setContentTransitionManager,
getContentScene,
setFinishOnTouchOutside,
setDefaultKeyMode, onKeyDown,
onKeyLongPress, onKeyUp,
onKeyMultiple, onBackPressed,
onKeyShortcut, onTouchEvent,
onTrackballEvent, onGenericMotionEvent,
onUserInteraction,
onWindowAttributesChanged,
onContentChanged,
onWindowFocusChanged,
onAttachedToWindow,
onDetachedFromWindow,
hasWindowFocus, dispatchKeyEvent,
dispatchKeyShortcutEvent

android.view.ContextThemeWrapper

attachBaseContext,
applyOverrideConfiguration,
getResources, setTheme, getTheme,
getSystemService,
onApplyThemeResource

android.content.ContextWrapper

attachBaseContext, getBaseContext,
getAssets, getResources,
getPackageManager,
getContentResolver, getMainLooper,
getApplicationContext, setTheme,
getTheme, getClassLoader,
getPackageName, getApplicationInfo,
getPackageResourcePath,
getPackageCodePath,
getSharedPreferences, openFileInput,
openFileOutput, deleteFile,
getFileStreamPath, fileList, getFilesDir,
getNoBackupFilesDir, getExternalFilesDir,
getExternalFilesDirs, getObbDir,
getObbDirs, getCacheDir,
getCodeCacheDir, getExternalCacheDir,
getExternalCacheDirs,
getExternalMediaDirs, getDir,
openOrCreateDatabase,
openOrCreateDatabase, deleteDatabase,
getDatabasePath, databaseList,
getWallpaper, peekWallpaper,
getWallpaperDesiredMinimumWidth,
getWallpaperDesiredMinimumHeight,
setWallpaper, setWallpaper,
clearWallpaper, startActivity,
startActivity, startActivities,
startActivities, startIntentSender,
startIntentSender, sendBroadcast,
sendBroadcast, sendOrderedBroadcast,
sendOrderedBroadcast,
sendBroadcastAsUser,
sendBroadcastAsUser,
sendOrderedBroadcastAsUser,
sendStickyBroadcast,
sendStickyOrderedBroadcast,
removeStickyBroadcast,
sendStickyBroadcastAsUser,
sendStickyOrderedBroadcastAsUser,
removeStickyBroadcastAsUser,
registerReceiver, registerReceiver,
unregisterReceiver, startService,
stopService, bindService, unbindService,
startInstrumentation, getSystemService,
getSystemServiceName,
checkPermission,
checkCallingPermission

android.content.Context

getAssets, getResources,
getPackageManager,
getContentResolver, getMainLooper,
getApplicationContext,
registerComponentCallbacks,
unregisterComponentCallbacks, getText,
getString, getString, getColor,
getDrawable, getColorStateList,
setTheme, getTheme,
obtainStyledAttributes,
obtainStyledAttributes,
obtainStyledAttributes,
obtainStyledAttributes, getClassLoader,
getPackageName, getApplicationInfo,
getPackageResourcePath,
getPackageCodePath,
getSharedPreferences, openFileInput,
openFileOutput, deleteFile,
getFileStreamPath, getFilesDir,
getNoBackupFilesDir, getExternalFilesDir,
getExternalFilesDirs, getObbDir,
getObbDirs, getCacheDir,
getCodeCacheDir, getExternalCacheDir,
getExternalCacheDirs,
getExternalMediaDirs, fileList, getDir,
openOrCreateDatabase,
openOrCreateDatabase, deleteDatabase,
getDatabasePath, databaseList,
getWallpaper, peekWallpaper,
getWallpaperDesiredMinimumWidth,
getWallpaperDesiredMinimumHeight,
setWallpaper, setWallpaper,
clearWallpaper, startActivity,
startActivity, startActivities,
startActivities, startIntentSender,
startIntentSender, sendBroadcast,
sendBroadcast, sendOrderedBroadcast,
sendOrderedBroadcast,
sendBroadcastAsUser,
sendBroadcastAsUser,
sendOrderedBroadcastAsUser,
sendStickyBroadcast,
sendStickyOrderedBroadcast,
removeStickyBroadcast,
sendStickyBroadcastAsUser,
sendStickyOrderedBroadcastAsUser,
removeStickyBroadcastAsUser

java.lang.Object

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

Class USBTransport

Hierarchy

java.lang.Object

[com.smartdevicelink.transport.SdlTransport](#)

Package

com.smartdevicelink.transport

Overview

Class that implements USB transport. A note about USB Accessory protocol. If the device is already in the USB accessory mode, any side (computer or Android) can open connection even if the other side is not connected. Conversely, if one side simply disconnects, the other side will NOT be notified and unblocked from reading data until some data is sent again or the USB is physically disconnected.

Quick Jump

Fields

- `ACTION_USB_ACCESSORY_ATTACHED`

Constructors

- `USBTransport( USBTransportConfig usbTransportConfig, ITransportListener transportListener )`

Methods

- `getState()`
- `sendBytesOverTransport(SdIPacket packet)`
- `registerReciever()`
- `openConnection()`
- `disconnect()`
- `stopReading()`
- `getTransportType()`
- `isAccessorySupported(UsbAccessory accessory)`
- `getBroadcastComment()`

Fields

ACTION_USB_ACCESSORY_ATTACHED

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

Constant Value

"com.smartdevice.link.USB_ACCESSORY_ATTACHED"

Constructors

USBTransport(
USBTransportConfig
usbTransportConfig,
ITransportListener
transportListener)

Constructs the USBTransport instance.

```
public USBTransport(USBTransportConfig usbTransportConfig,  
ITransportListener transportListener)
```

Parameters

usbTransportConfig

Config object for the USB transport

transportListener

Listener that gets notified on different transport events

Methods

getState()

Returns the current state of transport.

```
public com.smartdevicelink.transport.USBTransport.State getState()
```

sendBytesOverTransport(SdlPacket packet)

Sends the array of bytes over USB.

```
protected boolean sendBytesOverTransport(SdlPacket packet)
```


Parameters

packet

The packet that is to be written out on the USB transport

registerReciever()

```
public void registerReciever()
```

openConnection()

Opens a USB connection if not open yet.

```
public void openConnection() throws SdlException
```

Throws

- SdlException

disconnect()

Closes the USB connection if open.

```
public void disconnect()
```

stopReading()

Asks the reader thread to stop while it's possible. If it's blocked on read(), there is no way to stop it except for physical USB disconnect.

```
public void stopReading()
```

getTransportType()

Returns the type of the transport.

```
public TransportType getTransportType()
```

isAccessorySupported(UsbAccessory accessory)

Checks if the specified connected USB accessory is what we expect.

```
public static boolean isAccessorySupported(UsbAccessory accessory)
```

Parameters

accessory

Accessory to check

getBroadcastComment()

```
public java.lang.String getBroadcastComment()
```

Inherited Methods

From Class	Methods
com.smartdevicelink.transport.SdlTransport	getIsConnected, handleReceivedPacket, sendBytesOverTransport, sendBytes, handleTransportConnected, handleTransportDisconnected, handleTransportError, openConnection, disconnect, getTransportType, getBroadcastComment
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class USBTransportConfig

Hierarchy

java.lang.Object

BaseTransportConfig

Package

com.smartdevicelink.transport

Overview

NOTE: This should no longer be used. See the `MultiplexTransportConfig` and guides to understand how to implement USB multiplexing. This class and method of USB connection will be removed in the next major release. If a router service is available to handle multiplexing of the usb transport it will be used, and this app will connect to whatever router service hosts the USB connection.

Quick Jump

Constructors

- `USBTransportConfig( Context mainActivity )`
- `USBTransportConfig( Context mainActivity, UsbAccessory usbAccessory )`
- `USBTransportConfig( Context mainActivity, boolean shareConnection, boolean queryUsbAcc )`
- `USBTransportConfig( Context mainActivity, UsbAccessory usbAccessory, boolean shareConnection, boolean queryUsbAcc )`

Methods

- `getQueryUsbAcc()`
- `getUSBContext()`
- `getUsbAccessory()`
- `setUsbAccessory(UsbAccessory value)`
- `getTransportType()`

Constructors

USBTransportConfig(Context mainActivity)

NOTE: This should no longer be used. See the MultiplexTransportConfig and guides to understand how to implement USB multiplexing. This class and method of USB connection will be removed in the next major release. If a router service is available to handle multiplexing of the usb transport it will be used, and this app will connect to whatever router service hosts the USB connection.

```
public USBTransportConfig(Context mainActivity)
```

Parameters

mainActivity

context used to start USB transport

USBTransportConfig(Context mainActivity, UsbAccessory usbAccessory)

NOTE: This should no longer be used. See the MultiplexTransportConfig and guides to understand how to implement USB multiplexing. This class and

method of USB connection will be removed in the next major release. If a router service is available to handle multiplexing of the usb transport it will be used, and this app will connect to whatever router service hosts the USB connection.

```
public USBTransportConfig(Context mainActivity, UsbAccessory  
usbAccessory)
```

Parameters

mainActivity

context used to start USB transport

usbAccessory

the accessory that was given to this app

USBTransportConfig(Context mainActivity, boolean shareConnection, boolean queryUsbAcc)

NOTE: This should no longer be used. See the MultiplexTransportConfig and guides to understand how to implement USB multiplexing. This class and method of USB connection will be removed in the next major release. If a router

service is available to handle multiplexing of the usb transport it will be used, and this app will connect to whatever router service hosts the USB connection.

```
public USBTransportConfig(Context mainActivity,boolean  
shareConnection,boolean queryUsbAcc)
```

Parameters

mainActivity

context used to start USB transport

shareConnection

enable other sessions on this app to use this USB connection

queryUsbAcc

attempt to query the USB accessory if none is provided

USBTransportConfig(Context mainActivity, UsbAccessory usbAccessory, boolean shareConnection, boolean queryUsbAcc)

NOTE: This should no longer be used. See the MultiplexTransportConfig and guides to understand how to implement USB multiplexing. This class and method of USB connection will be removed in the next major release. If a router service is available to handle multiplexing of the usb transport it will be used, and this app will connect to whatever router service hosts the USB connection.

```
public USBTransportConfig(Context mainActivity,UsbAccessory  
usbAccessory,boolean shareConnection,boolean queryUsbAcc)
```

Parameters

mainActivity

context used to start USB transport

usbAccessory

the accessory that was given to this app

shareConnection

enable other sessions on this app to use this USB connection

queryUsbAcc

attempt to query the USB accessory if none is provided

Methods

getQueryUsbAcc()

```
public java.lang.Boolean getQueryUsbAcc()
```

getUSBContext()

```
public android.content.Context getUSBContext()
```

getUsbAccessory()

```
public android.hardware.usb.UsbAccessory getUsbAccessory()
```

setUsbAccessory(UsbAccessory value)

```
public void setUsbAccessory(UsbAccessory value)
```

Parameters

value

getTransportType()

```
public TransportType getTransportType()
```

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class UsbTransferProvider

Hierarchy

java.lang.Object

Package

com.smartdevicelink.transport

Quick Jump

Constructors

- [UsbTransferProvider\(Context context, ComponentName service, UsbAccessory usbAccessory, UsbTransferProvider.UsbTransferCallback callback \)](#)

Methods

- `setFlags(int flags)`
- `checkIsConnected()`
- `cancel()`

Constructors

`UsbTransferProvider(Context
context, ComponentName
service, UsbAccessory
usbAccessory,
UsbTransferProvider.UsbTransferCallback
callback)`

```
public UsbTransferProvider(Context context,ComponentName service,  
UsbAccessory usbAccessory,UsbTransferProvider.UsbTransferCallback  
callback)
```

Parameters

context

service
usbAccessory
callback

Methods

setFlags(int flags)

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

Parameters

flags

checkIsConnected()

```
public void checkIsConnected()
```

cancel()

```
public void cancel()
```

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Interface UsbTransferProvider.UsbTransferCa

Package

com.smartdevicelink.transport

Quick Jump

Methods

- `onUsbTransferUpdate(boolean success)`

Methods

onUsbTransferUpdate(boolean success)

```
public void onUsbTransferUpdate(boolean success)
```

Parameters

success

Class Version

Hierarchy

java.lang.Object

Package

com.smartdevicelink.proxy

Quick Jump

Fields

- **VERSION**

Constructors

- `Version()`

Fields

VERSION

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

Constructors

Version()

```
public Version()
```

Methods

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class VideoStreamManager

Hierarchy

java.lang.Object

BaseVideoStreamManager

Package

com.smartdevicelink.managers.video

Quick Jump

Constructors

- `VideoStreamManager( ISdl internalInterface )`

Methods

- `start(CompletionListener listener)`
- `startRemoteDisplayStream(Context context,java.lang.Class<? extends com.smartdevicelink.streaming.video.SdlRemoteDisplay> remoteDisplayClass,VideoStreamingParameters parameters,boolean encrypted)`
- `startStreaming(VideoStreamingParameters parameters,boolean encrypted)`
- `stopStreaming()`
- `resumeStreaming()`
- `dispose()`
- `isServiceActive()`
- `isStreaming()`
- `isPaused()`
- `currentVideoStreamState()`
- `convertTouchEvent(OnTouchEvent touchEvent)`
- `onTransportUpdate(java.util.List<TransportRecord> connectedTransports,boolean audioStreamTransportAvail,boolean videoStreamTransportAvail)`

Constructors

VideoStreamManager(ISdl internalInterface)

```
public VideoStreamManager(ISdl internalInterface)
```

Parameters

internalInterface

Methods

start(CompletionListener
listener)

```
public void start(CompletionListener listener)
```

Parameters

listener

```
startRemoteDisplayStream(  
Context context,  
java.lang.Class<? extends  
com.smartdevicelink.streaming.video.S  
remoteDisplayClass,  
VideoStreamingParameters  
parameters, boolean encrypted )
```

Starts streaming a remote display to the module if there is a connected session. This method of streaming requires the device to be on API level 19 or higher

```
public void startRemoteDisplayStream(Context context,java.lang.Class  
<? extends com.smartdevicelink.streaming.video.SdlRemoteDisplay>  
remoteDisplayClass,VideoStreamingParameters parameters,boolean  
encrypted)
```

Parameters

context

a context that can be used to create the remote display

remoteDisplayClass

class object of the remote display. This class will be used to create an instance of the remote display and will be projected to the module

parameters

streaming parameters to be used when streaming. If null is sent in, the default/optimized options will be used. If you are unsure about what parameters to be used it is best to just send null and let the system determine what works best for the currently connected module.

encrypted

a flag of if the stream should be encrypted. Only set if you have a supplied encryption library that the module can understand.

startStreaming(VideoStreamingParameters parameters, boolean encrypted)

Starts video service, sets up encoder, haptic manager, and remote display.
Begins streaming the remote display.

```
protected void startStreaming(VideoStreamingParameters parameters,  
boolean encrypted)
```

Parameters

parameters

Video streaming parameters including: codec which will be used for streaming (currently, only VideoStreamingCodec.H264 is accepted), height and width of the video in pixels.

encrypted

Specify true if packets on this service have to be encrypted

stopStreaming()

Stops streaming from the remote display. To restart, call

```
public void stopStreaming()
```

resumeStreaming()

Resumes streaming after calling

```
public void resumeStreaming()
```


dispose()

Stops streaming, ends video streaming service and removes service listeners.

```
public void dispose()
```

isServiceActive()

Check if a video service is currently active

```
public boolean isServiceActive()
```

isStreaming()

Check if video is currently streaming and visible

```
public boolean isStreaming()
```

isPaused()

Check if video streaming has been paused due to app moving to background or manually stopped

```
public boolean isPaused()
```

currentVideoStreamState()

Gets the current video streaming state as defined in @StreamingStateMachine

```
public int currentVideoStreamState()
```

convertTouchEvent(OnTouchEvent touchEvent)

```
protected android.view.MotionEvent convertTouchEvent(OnTouchEvent  
touchEvent)
```

Parameters

touchEvent

onTransportUpdate(
java.util.List<TransportRecord>
connectedTransports, boolean
audioStreamTransportAvail,
boolean
videoStreamTransportAvail)

```
protected void onTransportUpdate(java.util.List<TransportRecord>  
connectedTransports,boolean audioStreamTransportAvail,boolean  
videoStreamTransportAvail)
```

Parameters

connectedTransports

audioStreamTransportAvail

videoStreamTransportAvail

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class VirtualDisplayEncoder

Hierarchy

java.lang.Object

Package

com.smartdevicelink.encoder

Quick Jump

Constructors

- `VirtualDisplayEncoder()`

Methods

- `init(Context context, IVideoStreamListener outputListener, VideoStreamingParameters streamingParams)`
- `getStreamingParams()`
- `setStreamingParams(int displayDensity, ImageResolution resolution, int frameRate, int bitrate, int interval, VideoStreamingFormat format)`
- `setStreamingParams(VideoStreamingParameters streamingParams)`
- `start()`
- `shutDown()`
- `getVirtualDisplay()`

Constructors

VirtualDisplayEncoder()

```
public VirtualDisplayEncoder()
```

Methods

init(Context context, IVideoStreamListener outputListener, VideoStreamingParameters streamingParams)

Initialization method for VirtualDisplayEncoder object. MUST be called before start() or shutdown() Will overwrite previously set videoWeight and videoHeight

```
public void init(Context context, IVideoStreamListener outputListener,  
VideoStreamingParameters streamingParams) throws java.lang.  
Exception
```

Parameters

context

to create the virtual display

outputListener

the listener that the video frames will be sent through

streamingParams

parameters to create the virtual display and encoder

Throws

- `java.lang.Exception`

getStreamingParams()

```
public VideoStreamingParameters getStreamingParams()
```

setStreamingParams(int
displayDensity, ImageResolution
resolution, int frameRate, int
bitrate, int interval,
VideoStreamingFormat format)

```
public void setStreamingParams(int displayDensity,ImageResolution  
resolution,int frameRate,int bitrate,int interval,VideoStreamingFormat  
format)
```

Parameters

displayDensity
resolution
frameRate
bitrate
interval
format

```
setStreamingParams(  
VideoStreamingParameters  
streamingParams )
```



```
public void setStreamingParams(VideoStreamingParameters  
streamingParams)
```

Parameters

streamingParams

start()

NOTE: Must call init() without error before calling this method. Prepares the encoder and virtual display.

```
public void start() throws java.lang.Exception
```

Throws

- java.lang.Exception

shutDown()

```
public void shutdown()
```

getVirtualDisplay()

```
public android.view.Display getVirtualDisplay()
```

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class WiFiSocketFactory

Hierarchy

java.lang.Object

Package

com.smartdevicelink.transport.utl

Quick Jump

Constructors

- [WiFiSocketFactory\(\)](#)

Methods

- [createSocket\(Context context\)](#)

Constructors

WiFiSocketFactory()

```
public WiFiSocketFactory()
```

Methods

createSocket(Context context)

Try to create a TCP socket which is bound to Wi-Fi network (for Android 5+) On Android 5 and later, this method tries to create a Socket instance which is bound to a Wi-Fi network. If the phone is not connected to a Wi-Fi network, or the app lacks required permission (ACCESS_NETWORK_STATE), then this method simply creates a Socket instance with "new Socket();".

```
public static java.net.Socket createSocket(Context context)
```

Parameters

context

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize

Class WiProProtocol

Hierarchy

java.lang.Object

com.smartdevicelink.protocol.AbstractProtocol

Package

com.smartdevicelink.protocol

Quick Jump

Fields

- MAX_PROTOCOL_VERSION
- V1_V2_MTU_SIZE
- V3_V4_MTU_SIZE
- V1_HEADER_SIZE
- V2_HEADER_SIZE

Constructors

- WiProProtocol(IProtocolListener protocolListener)

Methods

- getMtu()
- getMtu(SessionType type)
- getVersion() **Deprecated**
- getProtocolVersion()
- getMajorVersionByte()

- `setVersion(byte version)`
- `StartProtocolSession(SessionType sessionType)`
- `EndProtocolSession(SessionType sessionType,byte sessionID,int hashId)`
- `SendMessage(ProtocolMessage protocolMsg)`
- `handlePacketReceived(SdlPacket packet)`
- `getFrameAssemblerForFrame(SdlPacket packet)`
- `StartProtocolService(SessionType sessionType,byte sessionID,boolean isEncrypted)`
- `SetHeartbeatSendInterval(int heartbeatSendInterval_ms)`
- `SetHeartbeatReceiveInterval(int heartbeatReceiveInterval_ms)`
- `SendHeartBeat(byte sessionID)`
- `SendHeartBeatACK(byte sessionID)`
- `EndProtocolService(SessionType serviceType,byte sessionID)`

Fields

MAX_PROTOCOL_VERSION

```
public static final Version MAX_PROTOCOL_VERSION
```

V1_V2_MTU_SIZE

```
public static final int V1_V2_MTU_SIZE
```

Constant Value

1500

V3_V4_MTU_SIZE

```
public static final int V3_V4_MTU_SIZE
```

Constant Value

131072

V1_HEADER_SIZE

```
public static final int V1_HEADER_SIZE
```

Constant Value

8

V2_HEADER_SIZE

```
public static final int V2_HEADER_SIZE
```

Constant Value

12

Constructors

WiProProtocol(IProtocolListener protocolListener)

```
public WiProProtocol(IProtocolListener protocolListener)
```

Parameters

protocolListener

Methods

getMtu()

Retrieves the max payload size for a packet to be sent to the module

```
public int getMtu()
```

getMtu(SessionType type)

```
public long getMtu(SessionType type)
```

Parameters

type

getVersion()

Deprecated

Use `getProtocolVersion()` or `getMajorVersionByte` instead.

Returns the Major version of the currently used protocol version

```
public byte getVersion()
```

getProtocolVersion()

```
public Version getProtocolVersion()
```

getMajorVersionByte()

```
public byte getMajorVersionByte()
```

setVersion(byte version)

This method will set the major protocol version that we should use. It will also set the default MTU based on version.

```
public void setVersion(byte version)
```

Parameters

version

StartProtocolSession(SessionType sessionType)

```
public void StartProtocolSession(SessionType sessionType)
```

Parameters

sessionType

EndProtocolSession(
SessionType sessionType, byte
sessionID, int hashId)

```
public void EndProtocolSession(SessionType sessionType,byte sessionID  
,int hashId)
```

Parameters

sessionType

sessionID

hashId

SendMessage(ProtocolMessage protocolMsg)

```
public void SendMessage(ProtocolMessage protocolMsg)
```

Parameters

protocolMsg

handlePacketReceived(SdIPacket packet)

```
public void handlePacketReceived(SdIPacket packet)
```

Parameters

packet

getFrameAssemblerForFrame(
SdlPacket packet)

```
protected com.smartdevicelink.protocol.WiProProtocol.  
MessageFrameAssembler getFrameAssemblerForFrame(SdlPacket  
packet)
```

Parameters

packet

StartProtocolService(
SessionType sessionType, byte
sessionID, boolean isEncrypted)

```
public void StartProtocolService(SessionType sessionType,byte  
sessionID,boolean isEncrypted)
```

Parameters

sessionType
sessionID
isEncrypted

SetHeartbeatSendInterval(int heartbeatSendInterval_ms)

```
public void SetHeartbeatSendInterval(int heartbeatSendInterval_ms)
```

Parameters

heartbeatSendInterval_ms

SetHeartbeatReceiveInterval(int heartbeatReceiveInterval_ms)

```
public void SetHeartbeatReceiveInterval(int  
heartbeatReceiveInterval_ms)
```

Parameters

heartbeatReceiveInterval_ms

SendHeartBeat(byte sessionId)

```
public void SendHeartBeat(byte sessionId)
```


Parameters

sessionID

SendHeartBeatACK(byte sessionID)

```
public void SendHeartBeatACK(byte sessionID)
```

Parameters

sessionID

EndProtocolService(SessionType serviceType, byte sessionID)

```
public void EndProtocolService(SessionType serviceType,byte sessionID)
```

Parameters

serviceType
sessionID

Inherited Methods

From Class	Methods
<code>com.smartdevicelink.protocol.AbstractProtocol</code>	<code>SendMessage</code> , <code>getMtu</code> , <code>getMtu</code> , <code>handlePacketReceived</code> , <code>StartProtocolSession</code> , <code>StartProtocolService</code> , <code>EndProtocolService</code> , <code>EndProtocolSession</code> , <code>SetHeartbeatSendInterval</code> , <code>SetHeartbeatReceiveInterval</code> , <code>SendHeartBeat</code> , <code>SendHeartBeatACK</code> , <code>handleProtocolFrameReceived</code> , <code>handlePacketToSend</code> , <code>handleProtocolMessageReceived</code> , <code>handleProtocolSessionEndedNACK</code> , <code>handleProtocolSessionEnded</code> , <code>handleProtocolSessionStarted</code> , <code>handleProtocolSessionNACKed</code> , <code>handleProtocolError</code> , <code>handleProtocolHeartbeat</code> , <code>handleProtocolHeartbeatACK</code> , <code>handleProtocolServiceDataACK</code> , <code>onResetIncomingHeartbeat</code>
<code>java.lang.Object</code>	<code>getClass</code> , <code>hashCode</code> , <code>equals</code> , <code>clone</code> , <code>toString</code> , <code>notify</code> , <code>notifyAll</code> , <code>wait</code> , <code>wait</code> , <code>wait</code> , <code>finalize</code>

Class WiProProtocol.MessageFrameAssem

Hierarchy

`java.lang.Object`

Package

com.smartdevicelink.protocol

Quick Jump

Fields

- [hasFirstFrame](#)
- [accumulator](#)
- [totalSize](#)
- [framesRemaining](#)

Constructors

- [MessageFrameAssembler\(\)](#)

Methods

- [handleFirstDataFrame\(SdlPacket packet\)](#)
- [handleRemainingFrame\(SdlPacket packet\)](#)
- [notifyIfFinished\(SdlPacket packet\)](#)
- [handleMultiFrameMessageFrame\(SdlPacket packet\)](#)

- `handleFrame(SdlPacket packet)`

Fields

hasFirstFrame

```
protected boolean hasFirstFrame
```

accumulator

```
protected java.io.ByteArrayOutputStream accumulator
```

totalSize

```
protected int totalSize
```

framesRemaining

```
protected int framesRemaining
```

Constructors

MessageFrameAssembler()

```
protected MessageFrameAssembler()
```

Methods

handleFirstDataFrame(SdlPacket packet)

```
protected void handleFirstDataFrame(SdlPacket packet)
```

Parameters

packet

handleRemainingFrame(
SdlPacket packet)

```
protected void handleRemainingFrame(SdlPacket packet)
```

Parameters

packet

notifyIfFinished(SdlPacket
packet)

```
protected void notifyIfFinished(SdIPacket packet)
```

Parameters

packet

handleMultiFrameMessageFrame (SdIPacket packet)

```
protected void handleMultiFrameMessageFrame(SdIPacket packet)
```

Parameters

packet

handleFrame(SdlPacket packet)

```
protected void handleFrame(SdlPacket packet)
```

Parameters

packet

Inherited Methods

From Class	Methods
java.lang.Object	getClass, hashCode, equals, clone, toString, notify, notifyAll, wait, wait, wait, finalize