



VNC® Mobile Server SDK for Android

Reference Manual

Version 4.2.3.6776

<http://automotive.realvnc.com/>

This SDK provides a Java API for adding VNC server capability to an Android application.

See: [Description](#)

Packages

Package	Description
com.realvnc.mirrorlink	Provides the classes describing various aspects of a MirrorLink connection.
com.realvnc.util	Provides some utility classes useful in the creation of a server.
com.realvnc.vncserver.android	Provides the main classes for use in managing a server instance.
com.realvnc.vncserver.core	Provides the core classes which are independent of Android.

This SDK provides a Java API for adding VNC server capability to an Android application.

The VNC Server SDK is provided as an Android "library project" in the SDK directory. Refer to the Android developer documentation for instructions on how to integrate a library project into your application.

Your AndroidManifest.xml file also needs to contain the following permission:

```
<uses-permission android:name="android.permission.WAKE_LOCK"/>
```

In order for [VncContextInformationManager](#) to function, your app must meet the following requirements:

Up to Android 4.4. Your AndroidManifest.xml file needs to contain the following permission:

```
<uses-permission android:name="android.permission.GET_TASKS"/>
```

Android 5.0 and above. You must either use a context information reporting capable remote control service, or your app must be system signed and your AndroidManifest.xml file needs to contain the following permission:

```
<uses-permission android:name="android.permission.REAL_GET_TASKS"/>
```

If you make use of the TCP based bearers, such as C, L, D or USBAND, then it is necessary for the AndroidManifest.xml file to contain the following permission:

```
<uses-permission android:name="android.permission.INTERNET"/>
```

The basic steps to implement a VNC server are as follows:

1. **Create an Android service.** A typical VNC server has to run "in the background" while the user is using other applications. On Android the correct mechanism to use for this is a Service object. See the Android developer documentation for details.

A Service is not strictly required; the VNC server can run in any Android context, such as an Activity. However this is unlikely to be useful in practice as it is possible for some of the methods in the server to block, and methods running on an application's main thread should not block.
2. **Implement a [VncServerListener](#) object.** The VNC server will call this object's methods to notify you of changes to its connection status. In particular it will call `connectedCb()` when a connection is established; you must respond by calling `accept(true)` to allow the connection to proceed.

`VncServerListener` is an abstract interface. You could implement it in your Service object.
3. **Create a [VncServer](#) object.** Pass in your Android service (as the context parameter) and your `VncServerListener` object.
4. **Control the VNC server** by calling the `VncServer` object's methods. For instance you can call the `listen()` method to make the server listen for incoming connections.

Please note that at the time of writing the core Android OS does not include facilities for screen grabbing and event injection, both of which are required to implement a functioning VNC server. These facilities are provided by RealVNC in the form of a remote control service. See the

document "Enabling Remote Control For Android Devices", provided with the VNC Mobile Solution for further details about supported devices and how to prepare a suitable remote control service.

VA H.264 Encoding

The VNC Server SDK can support the VA H.264 encoding, if an H.264 encoder plugin is provided. The basic steps to enable the H.264 support are as follows:

1. Implement a `VncH264Encoder` object.
2. Call `VncServer.setH264Encoder(VncH264Encoder h264Encoder, boolean debugModeEnabled)` to register the above implemented object with the SDK.

MirrorLink

This version of the Android server SDK is compatible with the MirrorLink standard. MirrorLink extends normal VNC. For information on how to create a MirrorLink-compliant server, see the methods on `VncServer` which are prefixed with 'ml' (for example `VncServer.mlSendDeviceStatus(VncDeviceStatus deviceStatus)`) and the `VncServerMirrorLinkListener` class.

Please note that MirrorLink functionality will not work unless the server SDK is provided with a license which specifically enables MirrorLink server functionality. This can be added using `VncServer.addLicense(String licenseText)`.

Further information about the MirrorLink standard can be obtained from the [Car Connectivity Consortium](#).

Legal Information

Copyright (C) 2002-2018 RealVNC Ltd. All Rights Reserved.

Details of and copyright notices for third-party software that is used by the VNC Android Server SDK can be found in the file `Acknowledgements.txt` in the SDK distribution.

RealVNC and VNC are trademarks of RealVNC Limited and are protected by trademark registrations and/or pending trademark applications in the European Union, United States of America and other jurisdictions. MirrorLink is a registered trademark of Car Connectivity Consortium LLC. Other trademarks are the property of their respective owners. Protected by UK patents 2481870, 2491657; US patents 8760366, 9137657; EU patent 2652951.

Hierarchy For All Packages

Package Hierarchies:

com.realvnc.mirrorlink, com.realvnc.util, com.realvnc.vncserver.android, com.realvnc.vncserver.core

Class Hierarchy

- java.lang.Object
 - com.realvnc.vncserver.android.**CustomRemoteControlServiceRequests**
 - com.realvnc.mirrorlink.**DisplayConfiguration**
 - com.realvnc.mirrorlink.**VNCClientDisplayConfiguration**
 - com.realvnc.mirrorlink.**VNCServerDisplayConfiguration**
 - com.realvnc.mirrorlink.**EventConfiguration**
 - com.realvnc.mirrorlink.**VNCServerEventConfiguration**
 - com.realvnc.mirrorlink.**VNCClientEventConfiguration**
 - com.realvnc.util.**IniFile**
 - com.realvnc.vncserver.android.**RemoteControlServiceCodes**
 - java.lang.Throwable (implements java.io.Serializable)
 - java.lang.Exception
 - com.realvnc.util.**IniFile.BadFormatException**
 - com.realvnc.vncserver.core.**VncException**
 - com.realvnc.vncserver.core.**VncLicenseNotValidException**
 - com.realvnc.mirrorlink.**VNCAudioBlockingNotification**
 - com.realvnc.mirrorlink.**VNCAudioInfo**
 - com.realvnc.vncserver.core.**VncAuthType**
 - com.realvnc.vncserver.core.**VncCommandStringBase**
 - com.realvnc.vncserver.android.**VncCommandString**
 - com.realvnc.mirrorlink.**VNCContextInformation**
 - com.realvnc.vncserver.android.**VncContextInformationManager**
 - com.realvnc.mirrorlink.**VNCDeviceStatus**
 - com.realvnc.vncserver.android.**VncDisplayInformationManager**
 - com.realvnc.vncserver.core.**VncEncryptionType**
 - com.realvnc.mirrorlink.**VNCFramebufferBlockingNotification**
 - com.realvnc.vncserver.android.**VncH264Encoder**
 - com.realvnc.util.**VncLog**
 - com.realvnc.mirrorlink.**VNCMirrorLinkKeys**
 - com.realvnc.vncserver.android.**VncOrientationManager**
 - com.realvnc.vncserver.core.**VncPixelFormat**
 - com.realvnc.vncserver.android.**VncServer**
 - com.realvnc.vncserver.android.**VncServerCallbackHandler** (implements com.realvnc.vncserver.android.VncServerOrientationListener)
 - com.realvnc.vncserver.android.**MirrorLinkCallbackHandler** (implements com.realvnc.vncserver.android.VncServerMirrorLinkListener)
 - com.realvnc.vncserver.core.**VncServerCoreErrors**
 - com.realvnc.vncserver.core.**VncServerState**
 - com.realvnc.vncserver.android.**VncSizeInt**
 - com.realvnc.mirrorlink.**VNCViewerEventConfiguration**

Interface Hierarchy

- com.realvnc.vncserver.core.**VncBearer**
- com.realvnc.vncserver.core.**VncBearerCallbacks**
- com.realvnc.vncserver.core.**VncBearerInfo**
- com.realvnc.vncserver.core.**VncConnection**
- com.realvnc.vncserver.android.**VncContextInformationManager.AccessibilityServiceProvider**
- com.realvnc.vncserver.android.**VncContextInformationManager.CapturedContextInformation**
- com.realvnc.vncserver.android.**VncContextInformationManager.Listener**
- com.realvnc.vncserver.android.**VncDisplayInformationManager.Listener**
- com.realvnc.vncserver.android.**VncExtension**
- com.realvnc.vncserver.android.**VncExtensionListener**

- com.realvnc.vncserver.android.**VncH264Encoder.BufferOwner**
- com.realvnc.vncserver.android.**VncH264Encoder.ScreenGrabHelper**
- com.realvnc.vncserver.android.**VncMirrorLinkKeyEventListener**
- com.realvnc.vncserver.android.**VncRemoteControlInfo**
- com.realvnc.vncserver.android.**VncRemoteFeatureCheckListener**
- com.realvnc.vncserver.android.**VncServerListener**
 - com.realvnc.vncserver.android.**VncServerMirrorLinkListener**
 - com.realvnc.vncserver.android.**VncServerOrientationListener**

Enum Hierarchy

- java.lang.Object
 - java.lang.Enum<E> (implements java.lang.Comparable<T>, java.io.Serializable)
 - com.realvnc.vncserver.android.**VncContextInformationManager.ListenerPriority**
 - com.realvnc.vncserver.android.**VncH264Encoder.FrameType**

Hierarchy For Package com.realvnc.mirrorlink

Package Hierarchies:

[All Packages](#)

Class Hierarchy

- java.lang.Object
 - com.realvnc.mirrorlink.**DisplayConfiguration**
 - com.realvnc.mirrorlink.**VNCClientDisplayConfiguration**
 - com.realvnc.mirrorlink.**VNCServerDisplayConfiguration**
 - com.realvnc.mirrorlink.**EventConfiguration**
 - com.realvnc.mirrorlink.**VNCServerEventConfiguration**
 - com.realvnc.mirrorlink.**VNCClientEventConfiguration**
 - com.realvnc.mirrorlink.**VNCAudioBlockingNotification**
 - com.realvnc.mirrorlink.**VNCAudioInfo**
 - com.realvnc.mirrorlink.**VNCContextInformation**
 - com.realvnc.mirrorlink.**VNCDeviceStatus**
 - com.realvnc.mirrorlink.**VNCFramebufferBlockingNotification**
 - com.realvnc.mirrorlink.**VNCMirrorLinkKeys**
 - com.realvnc.mirrorlink.**VNCViewerEventConfiguration**

Hierarchy For Package com.realvnc.util

Package Hierarchies:

[All Packages](#)

Class Hierarchy

- java.lang.Object
 - com.realvnc.util.**IniFile**
 - java.lang.Throwable (implements java.io.Serializable)
 - java.lang.Exception
 - com.realvnc.util.**IniFile.BadFormatException**
 - com.realvnc.util.**VncLog**

Hierarchy For Package com.realvnc.vncserver.android

Package Hierarchies:

[All Packages](#)

Class Hierarchy

- java.lang.Object
 - com.realvnc.vncserver.android.**CustomRemoteControlServiceRequests**
 - com.realvnc.vncserver.android.**RemoteControlServiceCodes**
 - com.realvnc.vncserver.core.**VncCommandStringBase**
 - com.realvnc.vncserver.android.**VncCommandString**
 - com.realvnc.vncserver.android.**VncContextInformationManager**
 - com.realvnc.vncserver.android.**VncDisplayInformationManager**
 - com.realvnc.vncserver.android.**VncH264Encoder**
 - com.realvnc.vncserver.android.**VncOrientationManager**
 - com.realvnc.vncserver.android.**VncServer**
 - com.realvnc.vncserver.android.**VncServerCallbackHandler** (implements com.realvnc.vncserver.android.VncServerOrientationListener)
 - com.realvnc.vncserver.android.**MirrorLinkCallbackHandler** (implements com.realvnc.vncserver.android.VncServerMirrorLinkListener)
 - com.realvnc.vncserver.android.**VncSizeInt**

Interface Hierarchy

- com.realvnc.vncserver.android.**VncContextInformationManager.AccessibilityServiceProvider**
- com.realvnc.vncserver.android.**VncContextInformationManager.CapturedContextInformation**
- com.realvnc.vncserver.android.**VncContextInformationManager.Listener**
- com.realvnc.vncserver.android.**VncDisplayInformationManager.Listener**
- com.realvnc.vncserver.android.**VncExtension**
- com.realvnc.vncserver.android.**VncExtensionListener**
- com.realvnc.vncserver.android.**VncH264Encoder.BufferOwner**
- com.realvnc.vncserver.android.**VncH264Encoder.ScreenGrabHelper**
- com.realvnc.vncserver.android.**VncMirrorLinkKeyEventListener**
- com.realvnc.vncserver.android.**VncRemoteControlInfo**
- com.realvnc.vncserver.android.**VncRemoteFeatureCheckListener**
- com.realvnc.vncserver.android.**VncServerListener**
 - com.realvnc.vncserver.android.**VncServerMirrorLinkListener**
 - com.realvnc.vncserver.android.**VncServerOrientationListener**

Enum Hierarchy

- java.lang.Object
 - java.lang.Enum<E> (implements java.lang.Comparable<T>, java.io.Serializable)
 - com.realvnc.vncserver.android.**VncContextInformationManager.ListenerPriority**
 - com.realvnc.vncserver.android.**VncH264Encoder.FrameType**

Hierarchy For Package com.realvnc.vncserver.core

Package Hierarchies:

[All Packages](#)

Class Hierarchy

- java.lang.Object
 - java.lang.Throwable (implements java.io.Serializable)
 - java.lang.Exception
 - com.realvnc.vncserver.core.VncException
 - com.realvnc.vncserver.core.VncLicenseNotValidException
 - com.realvnc.vncserver.core.VncAuthType
 - com.realvnc.vncserver.core.VncCommandStringBase
 - com.realvnc.vncserver.core.VncEncryptionType
 - com.realvnc.vncserver.core.VncPixelFormat
 - com.realvnc.vncserver.core.VncServerCoreErrors
 - com.realvnc.vncserver.core.VncServerState

Interface Hierarchy

- com.realvnc.vncserver.core.VncBearer
- com.realvnc.vncserver.core.VncBearerCallbacks
- com.realvnc.vncserver.core.VncBearerInfo
- com.realvnc.vncserver.core.VncConnection

Package com.realvnc.mirrorlink

Provides the classes describing various aspects of a MirrorLink connection.

See: [Description](#)

Class Summary	
Class	Description
DisplayConfiguration	Class defining constants for use in <code>VNCServerDisplayConfiguration</code> and <code>VNCClientDisplayConfiguration</code> classes.
EventConfiguration	Class defining constants for use in <code>VNCServerEventConfiguration</code> and <code>VNCClientEventConfiguration</code> classes.
VNCAudioBlockingNotification	Class holding an AudioBlockingNotification MirrorLink extension message to be sent to the server.
VNCAudioInfo	Class containing constants to be used when defining audio information for an application.
VNCClientDisplayConfiguration	Class holding a ClientDisplayConfiguration MirrorLink extension message to be sent to the server.
VNCClientEventConfiguration	Class holding a ClientEventConfiguration MirrorLink extension message to be sent to the server.
VNCContextInformation	Class holding a decoded ContextInformation rectangle that has been received from the server.
VNCDeviceStatus	Class holding a decoded DeviceStatus MirrorLink extension message that has been received from, or will be sent to, the server.
VNCFramebufferBlockingNotification	Class holding a FramebufferBlockingNotification MirrorLink extension message to be sent to the server.
VNCMirrorLinkKeys	VNCMirrorLinkKeys
VNCServerDisplayConfiguration	Class holding a decoded ServerDisplayConfiguration MirrorLink extension message that has been received from the server.
VNCServerEventConfiguration	Class holding a decoded ServerEventConfiguration MirrorLink extension message that has been received from the server.
VNCViewerEventConfiguration	Deprecated

Package com.realvnc.mirrorlink Description

Provides the classes describing various aspects of a MirrorLink connection.

Package com.realvnc.util

Provides some utility classes useful in the creation of a server.

See: [Description](#)

Class Summary	
Class	Description
IniFile	
VncLog	

Exception Summary	
Exception	Description

Package com.realvnc.util Description

Provides some utility classes useful in the creation of a server.

Package com.realvnc.vncserver.android

Provides the main classes for use in managing a server instance.

See: [Description](#)

Interface Summary

Interface	Description
VncContextInformationManager.AccessibilityServiceProvider	This interface represents a class that facilitates the usage of an accessibility service for context information gathering.
VncContextInformationManager.CapturedContextInformation	Interface describing the context information for an area of the device screen.
VncContextInformationManager.Listener	This interface allows objects to be notified of changes to the context information for the visual elements of the screen.
VncDisplayInformationManager.Listener	Listener interface used by the VNC Server Display Information Manager to notify the application that some of the display information has changed.
VncExtension	This is an opaque object used as a unique handle for an externally registered protocol extension.
VncExtensionListener	This interface is used for receiving externally defined protocol extension messages.
VncH264Encoder.BufferOwner	
VncH264Encoder.ScreenGrabHelper	
VncMirrorLinkKeyEventListener	Listener interface allowing server SDK users to implement custom handling of MirrorLink key events.
VncRemoteControlInfo	Objects implementing this interface are used to provide detailed information about the forms of remote control, and can be obtained through a call to the VncServer.getRemoteControlInfo() method.
VncRemoteFeatureCheckListener	This interface is used for receiving externally defined remote feature checks.
VncServerListener	Listener interface used by the VNC server to notify the application that certain events have occurred.
VncServerMirrorLinkListener	Extension to the VncServerListener class to provide extra callbacks in relation to events using the MirrorLink protocol.
VncServerOrientationListener	A type of listener which can be informed of orientation changes detected by the Android VNC server SDK.

Class Summary

Class	Description
CustomRemoteControlServiceRequests	Custom requests supported by Remote Control Service implementations provided by RealVNC for Android platforms.
MirrorLinkCallbackHandler	Base class for callbacks received for a MirrorLink server.
RemoteControlServiceCodes	Return or error codes that may be reported by Remote Control Service implementations provided by RealVNC for Android platforms.
VncCommandString	Android implementation encapsulating a VNC command string.
VncContextInformationManager	This class provides management of the context information for the applications, view and windows present on the display.
VncDisplayInformationManager	This class provides management of the display information related to the VNC session.
VncH264Encoder	This class can be extended to implement an H.264 encoder.
VncOrientationManager	This class provides management of the orientation of the device display.

VncServer	This class provides the API for a VNC server.
VncServerCallbackHandler	Base class for callbacks received for a VNC server.
VncSizeInt	Represents a width and height.

Enum Summary

Enum	Description
VncContextInformationManager.ListenerPriority	Enum to indicate priority of listeners.
VncH264Encoder.FrameType	

Package com.realvnc.vncserver.android Description

Provides the main classes for use in managing a server instance.

Package com.realvnc.vncserver.core

Provides the core classes which are independent of Android.

See: [Description](#)

Interface Summary

Interface	Description
VncBearer	Objects implementing this interface are used by the SDK to provide data transport facilities between the server and viewer.
VncBearerCallbacks	Objects implementing this interface are used by the SDK to provide a way for the bearers to call SDK provided functionality.
VncBearerInfo	Objects implementing this interface are used to provided detailed information on a pluggable bearer, and can be obtained through a call to the getBearerInfo method of the VncServer object.
VncConnection	An object representing a connection across which the server will talk to a VNC viewer.

Class Summary

Class	Description
VncAuthType	Type of authentication to be used by the VNC server.
VncCommandStringBase	A abstract class for encapsulating a VNC command string.
VncEncryptionType	Type of authentication to be used by the VNC server.
VncPixelFormat	Defines the format of the pixels in a framebuffer.
VncServerCoreErrors	VNC specific error codes to be returned from the VNC server.
VncServerState	Constants representing the various states that the VNC server can be in.

Exception Summary

Exception	Description
VncException	An exception class to describe errors using standard VNC error codes.
VncLicenseNotValidException	An exception class to describe license not valid errors.

Package com.realvnc.vncserver.core Description

Provides the core classes which are independent of Android.

Deprecated API

Contents

Deprecated Classes

Deprecated Fields

Deprecated Methods

Deprecated Classes

Class and Description

`com.realvnc.mirrorlink.VNCViewerEventConfiguration`

Deprecated Fields

Field and Description

`com.realvnc.mirrorlink.VNCDeviceStatus.FEATURE_VOICE_INPUT_REROUTING_DISABLED`

Use `VNCDeviceStatus.FEATURE_MICROPHONE_INPUT_DISABLED` instead.

`com.realvnc.mirrorlink.VNCDeviceStatus.FEATURE_VOICE_INPUT_REROUTING_ENABLED`

Use `VNCDeviceStatus.FEATURE_MICROPHONE_INPUT_ENABLED` instead.

`com.realvnc.mirrorlink.VNCDeviceStatus.FEATURE_VOICE_INPUT_REROUTING_IGNORED`

Use `VNCDeviceStatus.FEATURE_MICROPHONE_INPUT_IGNORED` instead.

`com.realvnc.mirrorlink.VNCDeviceStatus.FEATURE_VOICE_INPUT_REROUTING_MASK`

Use `VNCDeviceStatus.FEATURE_MICROPHONE_INPUT_MASK` instead.

Deprecated Methods

Method and Description

`com.realvnc.mirrorlink.VNCContextInformation.getContentRulesFollowed()`

Since MirrorLink 1.3. Must be ignored by MirrorLink 1.3 clients.

`com.realvnc.vncserver.android.VncOrientationManager.lockOrientation(int)`

Use `VncOrientationManager.lockOrientationEx(int)` instead

`com.realvnc.vncserver.android.VncServer.mlFramebufferBlockingNotificationHandled()`

Use `VncServer.mlFramebufferBlockingNotificationHandled(VNCFramebufferBlockingNotification)` instead.

`com.realvnc.vncserver.android.VncOrientationManager.restoreOrientationLock(int)`

Use `VncOrientationManager.restoreOrientationLockEx(int)` instead. Requests that the screen orientation lock is restored to the requested value once the VNC connection is over.

This method can be called at any point of the VNC connection, but it is recommended to be used as soon as the connection has started, so that when the VNC connection terminates the screen orientation lock holds the same value it did before it started.

The orientation provided should be one of the constants defined in this class: `VncOrientationManager.ORIENTATION_DISABLE_LOCK`, `VncOrientationManager.ORIENTATION_LANDSCAPE_LOCK` or `VncOrientationManager.ORIENTATION_PORTRAIT_LOCK`.

This feature was introduced in version 3.4. Older versions of the RCS default to disabling screen orientation lock at the end of a session. For compatibility, if this method is not called, that behaviour is reproduced (i.e. the orientation lock will always be restored to `VncOrientationManager.ORIENTATION_DISABLE_LOCK`).

If restoring the orientation lock is supported for the current set-up, this method will return `true`. Otherwise it will return `false`.

This method is deprecated because it will fail if invoked too early,

Constant Field Values

Contents

com.realvnc.*

com.realvnc.*

com.realvnc.mirrorlink.DisplayConfiguration

Modifier and Type	Constant Field	Value
public static final int	FRAMEBUFFER_CONFIGURATION_DOWNSCALING	8
public static final int	FRAMEBUFFER_CONFIGURATION_REPLACE_EMPTY_UPDATES	16
public static final int	FRAMEBUFFER_CONFIGURATION_SERVERSIDE_ORIENTATION_SWITCH	1
public static final int	FRAMEBUFFER_CONFIGURATION_SERVERSIDE_ROTATION	2
public static final int	FRAMEBUFFER_CONFIGURATION_SUPPORTS_FRAMEBUFFER_ALTERNATIVE_TEXT	32
public static final int	FRAMEBUFFER_CONFIGURATION_UPSCALING	4
public static final int	PIXELFORMAT_SUPPORT_ANY_16	8388608
public static final int	PIXELFORMAT_SUPPORT_ANY_24	32768
public static final int	PIXELFORMAT_SUPPORT_ANY_32	128
public static final int	PIXELFORMAT_SUPPORT_ARGB888_32	1
public static final int	PIXELFORMAT_SUPPORT_GRAYSCALE_16	16777216
public static final int	PIXELFORMAT_SUPPORT_GRAYSCALE_8	33554432
public static final int	PIXELFORMAT_SUPPORT_NONE	0
public static final int	PIXELFORMAT_SUPPORT_RGB_343_16	524288
public static final int	PIXELFORMAT_SUPPORT_RGB444_16	262144
public static final int	PIXELFORMAT_SUPPORT_RGB555_16	131072
public static final int	PIXELFORMAT_SUPPORT_RGB565_16	65536
public static final int	PIXELFORMAT_SUPPORT_RGB888_32	256
public static final int	RESIZE_FACTOR_1_1	1
public static final int	RESIZE_FACTOR_1_10	128
public static final int	RESIZE_FACTOR_1_16	256
public static final int	RESIZE_FACTOR_1_2	2
public static final int	RESIZE_FACTOR_1_3	4
public static final int	RESIZE_FACTOR_1_32	512
public static final int	RESIZE_FACTOR_1_4	8
public static final int	RESIZE_FACTOR_1_5	16
public static final int	RESIZE_FACTOR_1_6	32
public static final int	RESIZE_FACTOR_1_8	64
public static final int	RESIZE_FACTOR_2_3	1024
public static final int	RESIZE_FACTOR_3_4	2048
public static final int	RESIZE_FACTOR_NONE	0

com.realvnc.mirrorlink.EventConfiguration

Modifier and Type	Constant Field	Value
public static final int	DEVICE_KEY_SUPPORT_ALL	65535
public static final int	DEVICE_KEY_SUPPORT_APPLICATION	32
public static final int	DEVICE_KEY_SUPPORT_BACKWARD	4096
public static final int	DEVICE_KEY_SUPPORT_CLEAR	1024
public static final int	DEVICE_KEY_SUPPORT_DELETE	128
public static final int	DEVICE_KEY_SUPPORT_FORWARD	2048
public static final int	DEVICE_KEY_SUPPORT_HOME	8192
public static final int	DEVICE_KEY_SUPPORT_MENU	32768
public static final int	DEVICE_KEY_SUPPORT_OK	64

public static final int	DEVICE_KEY_SUPPORT_PHONE_CALL	1
public static final int	DEVICE_KEY_SUPPORT_PHONE_END	2
public static final int	DEVICE_KEY_SUPPORT_SEARCH	16384
public static final int	DEVICE_KEY_SUPPORT_SOFT_LEFT	4
public static final int	DEVICE_KEY_SUPPORT_SOFT_MIDDLE	8
public static final int	DEVICE_KEY_SUPPORT_SOFT_RIGHT	16
public static final int	DEVICE_KEY_SUPPORT_ZOOM_IN	256
public static final int	DEVICE_KEY_SUPPORT_ZOOM_OUT	512
public static final int	KNOB_KEY_SUPPORT_PULL_Z_0	16
public static final int	KNOB_KEY_SUPPORT_PULL_Z_1	4096
public static final int	KNOB_KEY_SUPPORT_PULL_Z_2	1048576
public static final int	KNOB_KEY_SUPPORT_PULL_Z_3	268435456
public static final int	KNOB_KEY_SUPPORT_PUSH_Z_0	8
public static final int	KNOB_KEY_SUPPORT_PUSH_Z_1	2048
public static final int	KNOB_KEY_SUPPORT_PUSH_Z_2	524288
public static final int	KNOB_KEY_SUPPORT_PUSH_Z_3	134217728
public static final int	KNOB_KEY_SUPPORT_ROTATE_X_0	32
public static final int	KNOB_KEY_SUPPORT_ROTATE_X_1	8192
public static final int	KNOB_KEY_SUPPORT_ROTATE_X_2	2097152
public static final int	KNOB_KEY_SUPPORT_ROTATE_X_3	536870912
public static final int	KNOB_KEY_SUPPORT_ROTATE_Y_0	64
public static final int	KNOB_KEY_SUPPORT_ROTATE_Y_1	16384
public static final int	KNOB_KEY_SUPPORT_ROTATE_Y_2	4194304
public static final int	KNOB_KEY_SUPPORT_ROTATE_Y_3	1073741824
public static final int	KNOB_KEY_SUPPORT_ROTATE_Z_0	128
public static final int	KNOB_KEY_SUPPORT_ROTATE_Z_1	32768
public static final int	KNOB_KEY_SUPPORT_ROTATE_Z_2	8388608
public static final int	KNOB_KEY_SUPPORT_ROTATE_Z_3	-2147483648
public static final int	KNOB_KEY_SUPPORT_SHIFT_X_0	1
public static final int	KNOB_KEY_SUPPORT_SHIFT_X_1	256
public static final int	KNOB_KEY_SUPPORT_SHIFT_X_2	65536
public static final int	KNOB_KEY_SUPPORT_SHIFT_X_3	16777216
public static final int	KNOB_KEY_SUPPORT_SHIFT_XY_0	4
public static final int	KNOB_KEY_SUPPORT_SHIFT_XY_1	1024
public static final int	KNOB_KEY_SUPPORT_SHIFT_XY_2	262144
public static final int	KNOB_KEY_SUPPORT_SHIFT_XY_3	67108864
public static final int	KNOB_KEY_SUPPORT_SHIFT_Y_0	2
public static final int	KNOB_KEY_SUPPORT_SHIFT_Y_1	512
public static final int	KNOB_KEY_SUPPORT_SHIFT_Y_2	131072
public static final int	KNOB_KEY_SUPPORT_SHIFT_Y_3	33554432
public static final int	MISC_KEY_SUPPORT_EVENT_MAPPING	8
public static final int	MISC_KEY_SUPPORT_FUNCTION_KEY_0	256
public static final int	MISC_KEY_SUPPORT_FUNCTION_KEY_1	512
public static final int	MISC_KEY_SUPPORT_FUNCTION_KEY_2	1024
public static final int	MISC_KEY_SUPPORT_FUNCTION_KEY_3	2048
public static final int	MISC_KEY_SUPPORT_FUNCTION_KEY_4	4096
public static final int	MISC_KEY_SUPPORT_FUNCTION_KEY_5	8192
public static final int	MISC_KEY_SUPPORT_FUNCTION_KEY_6	16384
public static final int	MISC_KEY_SUPPORT_FUNCTION_KEY_7	32768
public static final int	MISC_KEY_SUPPORT_FUNCTION_KEY_MASK	65280
public static final int	MISC_KEY_SUPPORT_FUNCTION_KEY_SHIFT	8
public static final int	MISC_KEY_SUPPORT_ITU	1
public static final int	MISC_KEY_SUPPORT_KEY_EVENT_LISTING	4
public static final int	MISC_KEY_SUPPORT_KEY_MAPPING_MASK	8
public static final int	MISC_KEY_SUPPORT_KEY_MAPPING_SHIFT	3
public static final int	MISC_KEY_SUPPORT_VIRTUAL_KEYBOARD_TRIGGER	2
public static final int	MULTIMEDIA_KEY_SUPPORT_FORWARD	8
public static final int	MULTIMEDIA_KEY_SUPPORT_MUTE	128

public static final int	MULTIMEDIA_KEY_SUPPORT_NEXT	32
public static final int	MULTIMEDIA_KEY_SUPPORT_PAUSE	2
public static final int	MULTIMEDIA_KEY_SUPPORT_PHOTO	512
public static final int	MULTIMEDIA_KEY_SUPPORT_PLAY	1
public static final int	MULTIMEDIA_KEY_SUPPORT_PREVIOUS	64
public static final int	MULTIMEDIA_KEY_SUPPORT_REWIND	16
public static final int	MULTIMEDIA_KEY_SUPPORT_STOP	4
public static final int	MULTIMEDIA_KEY_SUPPORT_UNMUTE	256
public static final int	POINTER_SUPPORT_POINTER_BUTTON_1	256
public static final int	POINTER_SUPPORT_POINTER_BUTTON_2	512
public static final int	POINTER_SUPPORT_POINTER_BUTTON_3	1024
public static final int	POINTER_SUPPORT_POINTER_BUTTON_4	2048
public static final int	POINTER_SUPPORT_POINTER_BUTTON_5	4096
public static final int	POINTER_SUPPORT_POINTER_BUTTON_6	8192
public static final int	POINTER_SUPPORT_POINTER_BUTTON_7	16384
public static final int	POINTER_SUPPORT_POINTER_BUTTON_8	32768
public static final int	POINTER_SUPPORT_POINTER_BUTTON_MASK	65280
public static final int	POINTER_SUPPORT_POINTER_EVENTS	1
public static final int	POINTER_SUPPORT_TOUCH_COUNT_1	0
public static final int	POINTER_SUPPORT_TOUCH_COUNT_10	589824
public static final int	POINTER_SUPPORT_TOUCH_COUNT_2	65536
public static final int	POINTER_SUPPORT_TOUCH_COUNT_3	131072
public static final int	POINTER_SUPPORT_TOUCH_COUNT_4	196608
public static final int	POINTER_SUPPORT_TOUCH_COUNT_5	262144
public static final int	POINTER_SUPPORT_TOUCH_COUNT_6	327680
public static final int	POINTER_SUPPORT_TOUCH_COUNT_7	393216
public static final int	POINTER_SUPPORT_TOUCH_COUNT_8	458752
public static final int	POINTER_SUPPORT_TOUCH_COUNT_9	524288
public static final int	POINTER_SUPPORT_TOUCH_COUNT_MASK	16711680
public static final int	POINTER_SUPPORT_TOUCH_COUNT_MASK_SHIFT	16
public static final int	POINTER_SUPPORT_TOUCH_EVENT_PRESSURE_MASK	-16777216
public static final int	POINTER_SUPPORT_TOUCH_EVENT_PRESSURE_MASK_SHIFT	24
public static final int	POINTER_SUPPORT_TOUCH_EVENTS	2

com.realvnc.mirrorlink.VNCAudioBlockingNotification

Modifier and Type	Constant Field	Value
public static final int	REASON_APPLICATION_CATEGORY_NOT_ALLOWED	1
public static final int	REASON_APPLICATION_NOT_TRUSTED	2
public static final int	REASON_APPLICATION_UNIQUE_ID_NOT_ALLOWED	4
public static final int	REASON_GLOBALLY_MUTED	8
public static final int	REASON_STREAM_MUTED	16
public static final int	REASON_UNBLOCK	0

com.realvnc.mirrorlink.VNCAudioInfo

Modifier and Type	Constant Field	Value
public static final int	AUDIO_CONTENT_CATEGORY_MEDIA_AUDIO_IN	4
public static final int	AUDIO_CONTENT_CATEGORY_MEDIA_AUDIO_OUT	2
public static final int	AUDIO_CONTENT_CATEGORY_MISC	-2147483648
public static final int	AUDIO_CONTENT_CATEGORY_PHONE_AUDIO	1
public static final int	AUDIO_CONTENT_CATEGORY_UNKNOWN	0
public static final int	AUDIO_CONTENT_CATEGORY_VOICE_COMMAND_IN	16
public static final int	AUDIO_CONTENT_CATEGORY_VOICE_COMMAND_OUT	8

com.realvnc.mirrorlink.VNCContextInformation

Modifier and Type	Constant Field	Value
-------------------	----------------	-------

public static final int	APPLICATION_CATEGORY_BROWSER	393216
public static final int	APPLICATION_CATEGORY_BROWSER_APPLICATION_STORE	393217
public static final int	APPLICATION_CATEGORY_IMMERSIVE_HOME_SCREEN	65542
public static final int	APPLICATION_CATEGORY_INFORMATION	524288
public static final int	APPLICATION_CATEGORY_INFORMATION_CLOCK	524294
public static final int	APPLICATION_CATEGORY_INFORMATION_NEWS	524289
public static final int	APPLICATION_CATEGORY_INFORMATION_SPORTS	524293
public static final int	APPLICATION_CATEGORY_INFORMATION_STOCKS	524291
public static final int	APPLICATION_CATEGORY_INFORMATION_TRAVEL	524292
public static final int	APPLICATION_CATEGORY_INFORMATION_WEATHER	524290
public static final int	APPLICATION_CATEGORY_MASK	-65536
public static final int	APPLICATION_CATEGORY_MEDIA	196608
public static final int	APPLICATION_CATEGORY_MEDIA_GAMING	196611
public static final int	APPLICATION_CATEGORY_MEDIA_IMAGE	196612
public static final int	APPLICATION_CATEGORY_MEDIA_MUSIC	196609
public static final int	APPLICATION_CATEGORY_MEDIA_VIDEO	196610
public static final int	APPLICATION_CATEGORY_MESSAGING	262144
public static final int	APPLICATION_CATEGORY_MESSAGING_EMAIL	262147
public static final int	APPLICATION_CATEGORY_MESSAGING_MMS	262146
public static final int	APPLICATION_CATEGORY_MESSAGING_SMS	262145
public static final int	APPLICATION_CATEGORY_NAVIGATION	327680
public static final int	APPLICATION_CATEGORY_NO_UI	-268435456
public static final int	APPLICATION_CATEGORY_NO_UI_CLIENT	-268435454
public static final int	APPLICATION_CATEGORY_NO_UI_CONVERSATIONAL_AUDIO	-268435424
public static final int	APPLICATION_CATEGORY_NO_UI_SERVER	-268435455
public static final int	APPLICATION_CATEGORY_NO_UI_VOICE_COMMAND_ENGINE	-268435440
public static final int	APPLICATION_CATEGORY_PHONE	131072
public static final int	APPLICATION_CATEGORY_PHONE_CALL_LOG	131074
public static final int	APPLICATION_CATEGORY_PHONE_CONTACT_LIST	131073
public static final int	APPLICATION_CATEGORY_PHONE_IMMERSIVE_CALL	131075
public static final int	APPLICATION_CATEGORY_PIM	655360
public static final int	APPLICATION_CATEGORY_PIM_CALENDAR	655361
public static final int	APPLICATION_CATEGORY_PIM_NOTES	655362
public static final int	APPLICATION_CATEGORY_PRODUCTIVITY	458752
public static final int	APPLICATION_CATEGORY_PRODUCTIVITY_DOCUMENT_EDITOR	458754
public static final int	APPLICATION_CATEGORY_PRODUCTIVITY_DOCUMENT_VIEWER	458753
public static final int	APPLICATION_CATEGORY_SOCIAL_NETWORKING	589824
public static final int	APPLICATION_CATEGORY_SWITCH_TO_CLIENT_NATIVE_UI	-268369921
public static final int	APPLICATION_CATEGORY_SYSTEM	-65536
public static final int	APPLICATION_CATEGORY_SYSTEM_INPUT_BLUETOOTH_PIN	-65534
public static final int	APPLICATION_CATEGORY_SYSTEM_INPUT_OTHER_PASSWORD	-65521
public static final int	APPLICATION_CATEGORY_SYSTEM_INPUT_UNLOCK_PIN	-65535
public static final int	APPLICATION_CATEGORY_SYSTEM_VOICE_COMMAND_CONFIRMATION	-65520
public static final int	APPLICATION_CATEGORY_TESTING_AND_CERTIFICATION	-131072
public static final int	APPLICATION_CATEGORY_UI	65536
public static final int	APPLICATION_CATEGORY_UI_APPLICATION_LISTING	65540
public static final int	APPLICATION_CATEGORY_UI_HOME_SCREEN	65537
public static final int	APPLICATION_CATEGORY_UI_MENU	65538
public static final int	APPLICATION_CATEGORY_UI_NOTIFICATION	65539
public static final int	APPLICATION_CATEGORY_UI_SETTINGS	65541
public static final int	APPLICATION_CATEGORY_UNKNOWN	0
public static final int	TRUST_LEVEL_APPLICATION_CERTIFICATE	160
public static final int	TRUST_LEVEL_REGISTERED_APPLICATION	128
public static final int	TRUST_LEVEL_SELF_REGISTERED_APPLICATION	96
public static final int	TRUST_LEVEL_UNKNOWN	0
public static final int	TRUST_LEVEL_USER_CONFIGURATION	64
public static final int	VISUAL_CONTENT_CATEGORY_CAR_MODE	65536

public static final int	VISUAL_CONTENT_CATEGORY_GRAPHICS_3D	16
public static final int	VISUAL_CONTENT_CATEGORY_GRAPHICS_VECTOR	128
public static final int	VISUAL_CONTENT_CATEGORY_IMAGE	4
public static final int	VISUAL_CONTENT_CATEGORY_MISC	-2147483648
public static final int	VISUAL_CONTENT_CATEGORY_TEXT	1
public static final int	VISUAL_CONTENT_CATEGORY_UI	32
public static final int	VISUAL_CONTENT_CATEGORY_UNKNOWN	0
public static final int	VISUAL_CONTENT_CATEGORY_VIDEO	2

com.realvnc.mirrorlink.VNCDeviceStatus

Modifier and Type	Constant Field	Value
public static final int	FEATURE_DEVICE_LOCK_DISABLED	8
public static final int	FEATURE_DEVICE_LOCK_ENABLED	12
public static final int	FEATURE_DEVICE_LOCK_IGNORED	0
public static final int	FEATURE_DEVICE_LOCK_MASK	12
public static final int	FEATURE_DRIVER_DISTRACTION_AVOIDANCE_DISABLED	131072
public static final int	FEATURE_DRIVER_DISTRACTION_AVOIDANCE_ENABLED	196608
public static final int	FEATURE_DRIVER_DISTRACTION_AVOIDANCE_IGNORED	0
public static final int	FEATURE_DRIVER_DISTRACTION_AVOIDANCE_MASK	196608
public static final int	FEATURE_FRAMEBUFFER_ROTATION_0_DEGREES	67108864
public static final int	FEATURE_FRAMEBUFFER_ROTATION_180_DEGREES	100663296
public static final int	FEATURE_FRAMEBUFFER_ROTATION_270_DEGREES	117440512
public static final int	FEATURE_FRAMEBUFFER_ROTATION_90_DEGREES	83886080
public static final int	FEATURE_FRAMEBUFFER_ROTATION_IGNORED	0
public static final int	FEATURE_FRAMEBUFFER_ROTATION_MASK	117440512
public static final int	FEATURE_KEY_LOCK_DISABLED	2
public static final int	FEATURE_KEY_LOCK_ENABLED	3
public static final int	FEATURE_KEY_LOCK_IGNORED	0
public static final int	FEATURE_KEY_LOCK_MASK	3
public static final int	FEATURE_MICROPHONE_INPUT_DISABLED	2048
public static final int	FEATURE_MICROPHONE_INPUT_ENABLED	3072
public static final int	FEATURE_MICROPHONE_INPUT_IGNORED	0
public static final int	FEATURE_MICROPHONE_INPUT_MASK	3072
public static final int	FEATURE_NIGHT_MODE_DISABLED	128
public static final int	FEATURE_NIGHT_MODE_ENABLED	192
public static final int	FEATURE_NIGHT_MODE_IGNORED	0
public static final int	FEATURE_NIGHT_MODE_MASK	192
public static final int	FEATURE_ORIENTATION_IGNORED	0
public static final int	FEATURE_ORIENTATION_LANDSCAPE	268435456
public static final int	FEATURE_ORIENTATION_MASK	402653184
public static final int	FEATURE_ORIENTATION_PORTRAIT	402653184
public static final int	FEATURE_SCREENSAVER_DISABLED	32
public static final int	FEATURE_SCREENSAVER_ENABLED	48
public static final int	FEATURE_SCREENSAVER_IGNORED	0
public static final int	FEATURE_SCREENSAVER_MASK	48
public static final int	FEATURE_VOICE_INPUT_DISABLED	512
public static final int	FEATURE_VOICE_INPUT_ENABLED	768
public static final int	FEATURE_VOICE_INPUT_IGNORED	0
public static final int	FEATURE_VOICE_INPUT_MASK	768
public static final int	FEATURE_VOICE_INPUT_REROUTING_DISABLED	2048
public static final int	FEATURE_VOICE_INPUT_REROUTING_ENABLED	3072
public static final int	FEATURE_VOICE_INPUT_REROUTING_IGNORED	0
public static final int	FEATURE_VOICE_INPUT_REROUTING_MASK	3072

com.realvnc.mirrorlink.VNCFramebufferBlockingNotification

Modifier and Type	Constant Field	Value
-------------------	----------------	-------

public static final int	REASON_APPLICATION_CATEGORY_NOT_ALLOWED	2
public static final int	REASON_APPLICATION_NOT_TRUSTED	8
public static final int	REASON_APPLICATION_UNIQUE_ID_NOT_ALLOWED	32
public static final int	REASON_CONTENT_CATEGORY_NOT_ALLOWED	1
public static final int	REASON_CONTENT_NOT_TRUSTED	4
public static final int	REASON_CONTENT_RULES_NOT_FOLLOWED	16
public static final int	REASON_UI_LAYOUT_NOT_SUPPORTED	1024
public static final int	REASON_UI_NOT_IN_FOCUS	256
public static final int	REASON_UI_NOT_VISIBLE	512

com.realvnc.mirrorlink.VNCMirrorLinkKeys

Modifier and Type	Constant Field	Value
public static final int	XK_DEVICE_APPLICATION	805306885
public static final int	XK_DEVICE_BACKWARD	805306892
public static final int	XK_DEVICE_CLEAR	805306890
public static final int	XK_DEVICE_DELETE	805306887
public static final int	XK_DEVICE_FORWARD	805306891
public static final int	XK_DEVICE_HOME	805306893
public static final int	XK_DEVICE_MENU	805306895
public static final int	XK_DEVICE_OK	805306886
public static final int	XK_DEVICE_PHONE_CALL	805306880
public static final int	XK_DEVICE_PHONE_END	805306881
public static final int	XK_DEVICE_SEARCH	805306894
public static final int	XK_DEVICE_SOFT_LEFT	805306882
public static final int	XK_DEVICE_SOFT_MIDDLE	805306883
public static final int	XK_DEVICE_SOFT_RIGHT	805306884
public static final int	XK_DEVICE_ZOOM_IN	805306888
public static final int	XK_DEVICE_ZOOM_OUT	805306889
public static final int	XK_FUNCTION_KEY_0	805307136
public static final int	XK_FUNCTION_KEY_1	805307137
public static final int	XK_FUNCTION_KEY_10	805307146
public static final int	XK_FUNCTION_KEY_11	805307147
public static final int	XK_FUNCTION_KEY_12	805307148
public static final int	XK_FUNCTION_KEY_2	805307138
public static final int	XK_FUNCTION_KEY_3	805307139
public static final int	XK_FUNCTION_KEY_4	805307140
public static final int	XK_FUNCTION_KEY_5	805307141
public static final int	XK_FUNCTION_KEY_6	805307142
public static final int	XK_FUNCTION_KEY_7	805307143
public static final int	XK_FUNCTION_KEY_8	805307144
public static final int	XK_FUNCTION_KEY_9	805307145
public static final int	XK_ITU_KEY_0	805306624
public static final int	XK_ITU_KEY_1	805306625
public static final int	XK_ITU_KEY_2	805306626
public static final int	XK_ITU_KEY_3	805306627
public static final int	XK_ITU_KEY_4	805306628
public static final int	XK_ITU_KEY_5	805306629
public static final int	XK_ITU_KEY_6	805306630
public static final int	XK_ITU_KEY_7	805306631
public static final int	XK_ITU_KEY_8	805306632
public static final int	XK_ITU_KEY_9	805306633
public static final int	XK_ITU_KEY_ASTERIX	805306634
public static final int	XK_ITU_KEY_POUND	805306635
public static final int	XK_KNOB_2D_ROTATE_x_0	805306378
public static final int	XK_KNOB_2D_ROTATE_x_0	805306379
public static final int	XK_KNOB_2D_ROTATE_x_1	805306394
public static final int	XK_KNOB_2D_ROTATE_x_1	805306395

public static final int	KK_KNOB_2D_ROTATE_x_2	805306410
public static final int	KK_KNOB_2D_ROTATE_X_2	805306411
public static final int	KK_KNOB_2D_ROTATE_x_3	805306426
public static final int	KK_KNOB_2D_ROTATE_X_3	805306427
public static final int	KK_KNOB_2D_ROTATE_y_0	805306380
public static final int	KK_KNOB_2D_ROTATE_Y_0	805306381
public static final int	KK_KNOB_2D_ROTATE_y_1	805306396
public static final int	KK_KNOB_2D_ROTATE_Y_1	805306397
public static final int	KK_KNOB_2D_ROTATE_y_2	805306412
public static final int	KK_KNOB_2D_ROTATE_Y_2	805306413
public static final int	KK_KNOB_2D_ROTATE_y_3	805306428
public static final int	KK_KNOB_2D_ROTATE_Y_3	805306429
public static final int	KK_KNOB_2D_ROTATE_z_0	805306382
public static final int	KK_KNOB_2D_ROTATE_Z_0	805306383
public static final int	KK_KNOB_2D_ROTATE_z_1	805306398
public static final int	KK_KNOB_2D_ROTATE_Z_1	805306399
public static final int	KK_KNOB_2D_ROTATE_z_2	805306414
public static final int	KK_KNOB_2D_ROTATE_Z_2	805306415
public static final int	KK_KNOB_2D_ROTATE_z_3	805306430
public static final int	KK_KNOB_2D_ROTATE_Z_3	805306431
public static final int	KK_KNOB_2D_SHIFT_DOWN_0	805306373
public static final int	KK_KNOB_2D_SHIFT_DOWN_1	805306389
public static final int	KK_KNOB_2D_SHIFT_DOWN_2	805306405
public static final int	KK_KNOB_2D_SHIFT_DOWN_3	805306421
public static final int	KK_KNOB_2D_SHIFT_DOWN_LEFT_0	805306375
public static final int	KK_KNOB_2D_SHIFT_DOWN_LEFT_1	805306391
public static final int	KK_KNOB_2D_SHIFT_DOWN_LEFT_2	805306407
public static final int	KK_KNOB_2D_SHIFT_DOWN_LEFT_3	805306423
public static final int	KK_KNOB_2D_SHIFT_DOWN_RIGHT_0	805306374
public static final int	KK_KNOB_2D_SHIFT_DOWN_RIGHT_1	805306390
public static final int	KK_KNOB_2D_SHIFT_DOWN_RIGHT_2	805306406
public static final int	KK_KNOB_2D_SHIFT_DOWN_RIGHT_3	805306422
public static final int	KK_KNOB_2D_SHIFT_LEFT_0	805306369
public static final int	KK_KNOB_2D_SHIFT_LEFT_1	805306385
public static final int	KK_KNOB_2D_SHIFT_LEFT_2	805306401
public static final int	KK_KNOB_2D_SHIFT_LEFT_3	805306417
public static final int	KK_KNOB_2D_SHIFT_PULL_0	805306377
public static final int	KK_KNOB_2D_SHIFT_PULL_1	805306393
public static final int	KK_KNOB_2D_SHIFT_PULL_2	805306409
public static final int	KK_KNOB_2D_SHIFT_PULL_3	805306425
public static final int	KK_KNOB_2D_SHIFT_PUSH_0	805306376
public static final int	KK_KNOB_2D_SHIFT_PUSH_1	805306392
public static final int	KK_KNOB_2D_SHIFT_PUSH_2	805306408
public static final int	KK_KNOB_2D_SHIFT_PUSH_3	805306424
public static final int	KK_KNOB_2D_SHIFT_RIGHT_0	805306368
public static final int	KK_KNOB_2D_SHIFT_RIGHT_1	805306384
public static final int	KK_KNOB_2D_SHIFT_RIGHT_2	805306400
public static final int	KK_KNOB_2D_SHIFT_RIGHT_3	805306416
public static final int	KK_KNOB_2D_SHIFT_UP_0	805306370
public static final int	KK_KNOB_2D_SHIFT_UP_1	805306386
public static final int	KK_KNOB_2D_SHIFT_UP_2	805306402
public static final int	KK_KNOB_2D_SHIFT_UP_3	805306418
public static final int	KK_KNOB_2D_SHIFT_UP_LEFT_0	805306372
public static final int	KK_KNOB_2D_SHIFT_UP_LEFT_1	805306388
public static final int	KK_KNOB_2D_SHIFT_UP_LEFT_2	805306404
public static final int	KK_KNOB_2D_SHIFT_UP_LEFT_3	805306420
public static final int	KK_KNOB_2D_SHIFT_UP_RIGHT_0	805306371

public static final int	XK_KNOB_2D_SHIFT_UP_RIGHT_1	805306387
public static final int	XK_KNOB_2D_SHIFT_UP_RIGHT_2	805306403
public static final int	XK_KNOB_2D_SHIFT_UP_RIGHT_3	805306419
public static final int	XK_MULTIMEDIA_FORWARD	805307395
public static final int	XK_MULTIMEDIA_MUTE	805307399
public static final int	XK_MULTIMEDIA_NEXT	805307397
public static final int	XK_MULTIMEDIA_PAUSE	805307393
public static final int	XK_MULTIMEDIA_PHOTO	805307401
public static final int	XK_MULTIMEDIA_PLAY	805307392
public static final int	XK_MULTIMEDIA_PREVIOUS	805307398
public static final int	XK_MULTIMEDIA_REWIND	805307396
public static final int	XK_MULTIMEDIA_STOP	805307394
public static final int	XK_MULTIMEDIA_UNMUTE	805307400

com.realvnc.util.IniFile.BadFormatException

Modifier and Type	Constant Field	Value
public static final long	serialVersionUID	1L

com.realvnc.vncserver.android.CustomRemoteControlServiceRequests

Modifier and Type	Constant Field	Value
public static final java.lang.String	ENABLE_HEADS_UP_NOTIFICATIONS	"com.realvnc.enableHeadsUpNotifications"
public static final java.lang.String	ENABLE_REMOTE_CONTROL	"com.realvnc.enableRemoteControl"
public static final java.lang.String	SET_STATUS_BAR_INFO	"com.realvnc.setStatusBarInfo"
public static final java.lang.String	SET_SYSTEM_UI_VISIBILITY	"com.realvnc.setSystemUiVisibility"

com.realvnc.vncserver.android.RemoteControlServiceCodes

Modifier and Type	Constant Field	Value
public static final int	RC_CAPTURE_TEMPORARILY_UNAVAILABLE	-1
public static final int	RC_DEVICE_ADMIN_NOT_ENABLED	2
public static final int	RC_DISCONNECTED	4
public static final int	RC_INCREMENTAL_UPDATES_UNAVAILABLE	5
public static final int	RC_PERMISSION_DENIED	1
public static final int	RC_SERVICE_ILLEGAL_ARGUMENT	8
public static final int	RC_SERVICE_ITSELF_LACKING_PERMISSIONS	6
public static final int	RC_SERVICE_LACKING_OTHER_OS_FACILITIES	7
public static final int	RC_SERVICE_UNAVAILABLE	3
public static final int	RC_SUCCESS	0

com.realvnc.vncserver.android.VncContextInformationManager

Modifier and Type	Constant Field	Value
public static final int	CHANGE_FLAG_ESTIMATED	4
public static final int	CHANGE_FLAG_POLLED	1
public static final int	CHANGE_FLAG_SYNCHRONOUS	2
public static final java.lang.String	CLASS_BUTTON_BAR	"com.android.internal.buttonbar"
public static final java.lang.String	CLASS_KEYGUARD	"com.android.internal.KeyguardView"
public static final java.lang.String	CLASS_STATUS_BAR	"com.android.internal.statusbar"
public static final java.lang.String	CLASS_TOAST	"com.android.internal.toast"
public static final int	CONTEXT_FLAG_SYSTEM_UI	1
public static final java.lang.String	PACKAGE_SYSTEM	"android.uid.system"

com.realvnc.vncserver.android.VncH264Encoder

Modifier and Type	Constant Field	Value
public static final int	H264_LEVEL_3_1	31
public static final int	H264_LEVEL_4_1	41

public static final int	H264_PROFILE_BASELINE	1
-------------------------	-----------------------	---

com.realvnc.vncserver.android.VncMirrorLinkKeyListener

Modifier and Type	Constant Field	Value
public static final int	FLAG_CLIENT_REPEAT	2
public static final int	FLAG_KEY_DOWN	1
public static final int	FLAG_SERVER_REPEAT	4

com.realvnc.vncserver.android.VncOrientationManager

Modifier and Type	Constant Field	Value
public static final int	ORIENTATION_DISABLE_LOCK	-1
public static final int	ORIENTATION_LANDSCAPE_LOCK	0
public static final int	ORIENTATION_PORTRAIT_LOCK	1

com.realvnc.vncserver.android.VncServer

Modifier and Type	Constant Field	Value
public static final int	FEATURE_CLIPBOARD	1
public static final int	FEATURE_COMPARE_FB	7
public static final int	FEATURE_MIRRORLINK_FORBID_PORTRAIT_ORIENTATION	9
public static final int	FEATURE_RFB4	5
public static final int	FEATURE_SEND_CLIPBOARD_ON_CONNECTION	2
public static final int	FEATURE_START_IN_LANDSCAPE	8
public static final int	FEATURE_VIEW_ONLY	4
public static final int	FEATURE_WINCE_SET_DISPLAY_POLL_FREQUENCY	3

com.realvnc.vncserver.core.VncAuthType

Modifier and Type	Constant Field	Value
public static final int	VNC_AUTH_NONE	1
public static final int	VNC_AUTH_PASS	2
public static final int	VNC_AUTH_REV	0
public static final int	VNC_AUTH_USER_PASS	3

com.realvnc.vncserver.core.VncEncryptionType

Modifier and Type	Constant Field	Value
public static final int	VNC_ENCRYPTION_AES_128	1
public static final int	VNC_ENCRYPTION_NONE	0

com.realvnc.vncserver.core.VncException

Modifier and Type	Constant Field	Value
public static final long	serialVersionUID	1L

com.realvnc.vncserver.core.VncLicenseNotValidException

Modifier and Type	Constant Field	Value
public static final long	serialVersionUID	1L

com.realvnc.vncserver.core.VncServerCoreErrors

Modifier and Type	Constant Field	Value
public static final int	VNCSERVER_ERR_ALREADY_EXISTS	65
public static final int	VNCSERVER_ERR_BAD_CHALLENGE	82
public static final int	VNCSERVER_ERR_BAD_CRYPT	47
public static final int	VNCSERVER_ERR_BAD_MESSAGE	80

public static final int	VNCSERVER_ERR_BAD_PIXEL_FORMAT	49
public static final int	VNCSERVER_ERR_BAD_PORT	25
public static final int	VNCSERVER_ERR_BAD_SESSION_ID	81
public static final int	VNCSERVER_ERR_BEARER_NOT_FOUND	50
public static final int	VNCSERVER_ERR_CAPTURE_FRAME_BUFFER_NOT_IMPLEMENTED	120
public static final int	VNCSERVER_ERR_COMMAND_FETCH_FAILED	101
public static final int	VNCSERVER_ERR_COMMAND_SUPERSEDED	106
public static final int	VNCSERVER_ERR_CONNECTION_CLOSED	43
public static final int	VNCSERVER_ERR_CONNECTION_REFUSED	22
public static final int	VNCSERVER_ERR_CRITICAL_CAPABILITY_UNSUPPORTED	55
public static final int	VNCSERVER_ERR_DEPRECATED_FIELD_USED	68
public static final int	VNCSERVER_ERR_ENVIRONMENT	107
public static final int	VNCSERVER_ERR_FEATURE_NOT_LICENSED	54
public static final int	VNCSERVER_ERR_HOST_UNREACHABLE	21
public static final int	VNCSERVER_ERR_INSUFFICIENT_BUFFER_SPACE	52
public static final int	VNCSERVER_ERR_INTERNAL_ERROR	102
public static final int	VNCSERVER_ERR_INVALID_COMMAND_STRING	44
public static final int	VNCSERVER_ERR_INVALID_PARAMETER	60
public static final int	VNCSERVER_ERR_KEY_GENERATION	63
public static final int	VNCSERVER_ERR_KEY_TOO_BIG	46
public static final int	VNCSERVER_ERR_LICENSE_NOT_VALID	53
public static final int	VNCSERVER_ERR_LOGIN_REJECTED	41
public static final int	VNCSERVER_ERR_NAME_LOOKUP_FAILED	23
public static final int	VNCSERVER_ERR_NETWORK	20
public static final int	VNCSERVER_ERR_NETWORK_LOST	26
public static final int	VNCSERVER_ERR_NO_ENCODINGS	48
public static final int	VNCSERVER_ERR_NO_SUITABLE_RCS	122
public static final int	VNCSERVER_ERR_NONE	0
public static final int	VNCSERVER_ERR_NOT_LICENSED_FOR_VIEWER	42
public static final int	VNCSERVER_ERR_PEER_TIMEOUT	83
public static final int	VNCSERVER_ERR_PERMISSIONS	3
public static final int	VNCSERVER_ERR_PORT_IN_USE	24
public static final int	VNCSERVER_ERR_PROTOCOL_MISMATCH	40
public static final int	VNCSERVER_ERR_RCS_EXITED	125
public static final int	VNCSERVER_ERR_RCS_LACKS_PERMISSIONS	123
public static final int	VNCSERVER_ERR_RCS_LIBRARY_NOT_FOUND	121
public static final int	VNCSERVER_ERR_RCS_NOT_ENABLED	124
public static final int	VNCSERVER_ERR_RESET	67
public static final int	VNCSERVER_ERR_RESOURCES	1
public static final int	VNCSERVER_ERR_SIGNATURE_REJECTED	51
public static final int	VNCSERVER_ERR_STATE	2
public static final int	VNCSERVER_ERR_TIMED_OUT	27
public static final int	VNCSERVER_ERR_TOO_LOW_ANDROID_VERSION	130
public static final int	VNCSERVER_ERR_TOO_LOW_OPENGL_ES_VERSION	131
public static final int	VNCSERVER_ERR_TOO_MANY_EXTENSIONS	66
public static final int	VNCSERVER_ERR_TOO_MANY_EXTERNAL_ENCODERS	56
public static final int	VNCSERVER_ERR_UNABLE_TO_START_SERVICE	64
public static final int	VNCSERVER_ERR_UNDERLYING_LIBRARY_NOT_FOUND	31
public static final int	VNCSERVER_ERR_UNSUPPORTED_AUTH	45
public static final int	VNCSERVER_ERR_USB_NOT_CONNECTED	30
public static final int	VNCSERVER_ERR_USER_REFUSED_CONNECTION	100

com.realvnc.vncserver.core.VncServerState

Modifier and Type	Constant Field	Value
public static final int	VNC_STATE_ACCEPT_REMOTE_KEY	7
public static final int	VNC_STATE_ACCEPTING	6
public static final int	VNC_STATE_AUTH	8

public static final int	VNC_STATE_AWAITING_KEY	1
public static final int	VNC_STATE_CONNECTING	4
public static final int	VNC_STATE_CONNECTING_RELAY	5
public static final int	VNC_STATE_DISCONNECTED	0
public static final int	VNC_STATE_EXITING	100
public static final int	VNC_STATE_GENERATING_KEY	2
public static final int	VNC_STATE_HANDSHAKING	11
public static final int	VNC_STATE_LISTENING	3
public static final int	VNC_STATE_ML_AWAITING_CLIENT_DISPLAY_CONFIGURATION	51
public static final int	VNC_STATE_ML_AWAITING_CLIENT_EVENT_CONFIGURATION	53
public static final int	VNC_STATE_ML_AWAITING_SERVER_DISPLAY_CONFIGURATION	50
public static final int	VNC_STATE_ML_AWAITING_SERVER_EVENT_CONFIGURATION	52
public static final int	VNC_STATE_REVERSE_AUTH	9
public static final int	VNC_STATE_RUNNING	12
public static final int	VNC_STATE_SETUP	10

Serialized Form

Package com.realvnc.util

Class `com.realvnc.util.IniFile.BadFormatException` extends `java.lang.Exception` implements `Serializable`

serialVersionUID: 1L

Package com.realvnc.vncserver.core

Class `com.realvnc.vncserver.core.VncException` extends `java.lang.Exception` implements `Serializable`

serialVersionUID: 1L

Serialized Fields

errorCode

`int errorCode`

cause

`java.lang.Exception cause`

Class `com.realvnc.vncserver.core.VncLicenseNotValidException` extends `VncException` implements `Serializable`

serialVersionUID: 1L

Serialized Fields

serialNumber

`byte[] serialNumber`

No longer set, in the event of an invalid license we do not return the serial number. This field has been kept in order to avoid a build break.

com.realvnc.mirrorlink

Class DisplayConfiguration

java.lang.Object
com.realvnc.mirrorlink.DisplayConfiguration

Direct Known Subclasses:
VNCClientDisplayConfiguration, VNCServerDisplayConfiguration

```
public abstract class DisplayConfiguration
extends java.lang.Object
```

Class defining constants for use inVNCServerDisplayConfiguration and VNCClientDisplayConfiguration classes.

The FRAMEBUFFER_* constants for use with the framebufferConfiguration field in the ServerDisplayConfiguration and ClientDisplayConfiguration structures.

If any of these bits is set in the VNCServerDisplayConfiguration, then the server is notifying the viewer application that the server supports the feature.

If you set any of these bits in the VNCClientDisplayConfiguration, then the viewer is indicating to the server that it intends to make use of the feature.

The PIXELFORMAT_* constants for use with the pixelFormatSupport field in the VNCServerDisplayConfiguration structure.

In a MirrorLink session, care should be taken to only request a pixel format that the server has said that it supports. (Support for 32-bit ARGB888 and 16-bit RGB565 is mandatory.)

Note that the Viewer SDK does not support any grayscale pixel formats, and does not support 24-bit true color pixel formats. (In this context, 24-bit is the size of each pixel, as given by the bitsPerPixel field in the VNCPixelFormat structure, and not the color depth.)

Field Summary

Fields	
Modifier and Type	Field and Description
static int	FRAMEBUFFER_CONFIGURATION_DOWNSCALING The server is capable of downscaling its framebuffer if its own dimensions are greater than those of the client display, which are specified in the VNCClientDisplayConfiguration structure.
static int	FRAMEBUFFER_CONFIGURATION_REPLACE_EMPTY_UPDATES The server is capable of suppressing empty update rectangles, so that the viewer application does not have to deal with them.
static int	FRAMEBUFFER_CONFIGURATION_SERVERSIDE_ORIENTATION_SWITCH The server is capable of switching orientations, as instructed by an appropriate DeviceStatusRequest message.
static int	FRAMEBUFFER_CONFIGURATION_SERVERSIDE_ROTATION The server is capable of rotating its framebuffer, as instructed by an appropriate DeviceStatusRequest message.
static int	FRAMEBUFFER_CONFIGURATION_SUPPORTS_FRAMEBUFFER_ALTERNATIVE_TEXT The server supports FramebufferAlternativeText messages.
static int	FRAMEBUFFER_CONFIGURATION_UPSCALING The server is capable of upscaling its framebuffer if its own dimensions are less than those of the client display, which are specified in the VNCClientDisplayConfiguration structure.
static int	PIXELFORMAT_SUPPORT_ANY_16 Supports any other 16-bit true color pixel formats.
static int	PIXELFORMAT_SUPPORT_ANY_24 Supports any other 24-bit true color pixel formats.
static int	PIXELFORMAT_SUPPORT_ANY_32 Supports any other 32-bit true color pixel formats.

static int	PIXELFORMAT_SUPPORT_ARGB888_32	Supports the ARGB32 pixel format.
static int	PIXELFORMAT_SUPPORT_GRAYSCALE_16	Supports 16-bit grayscale.
static int	PIXELFORMAT_SUPPORT_GRAYSCALE_8	Supports 8-bit grayscale.
static int	PIXELFORMAT_SUPPORT_NONE	Supports no pixel formats.
static int	PIXELFORMAT_SUPPORT_RGB_343_16	Supports the RGB343 pixel formats.
static int	PIXELFORMAT_SUPPORT_RGB444_16	Supports the RGB444 pixel formats.
static int	PIXELFORMAT_SUPPORT_RGB555_16	Supports the RGB555 pixel formats.
static int	PIXELFORMAT_SUPPORT_RGB565_16	Supports the RGB565 pixel formats.
static int	PIXELFORMAT_SUPPORT_RGB888_32	Supports the RGB888 pixel format.
static int	RESIZE_FACTOR_1_1	Resize factor of 1/1
static int	RESIZE_FACTOR_1_10	Resize factor of 1/10
static int	RESIZE_FACTOR_1_16	Resize factor of 1/16
static int	RESIZE_FACTOR_1_2	Resize factor of 1/2
static int	RESIZE_FACTOR_1_3	Resize factor of 1/3
static int	RESIZE_FACTOR_1_32	Resize factor of 1/32
static int	RESIZE_FACTOR_1_4	Resize factor of 1/4
static int	RESIZE_FACTOR_1_5	Resize factor of 1/5
static int	RESIZE_FACTOR_1_6	Resize factor of 1/6
static int	RESIZE_FACTOR_1_8	Resize factor of 1/8
static int	RESIZE_FACTOR_2_3	Resize factor of 2_3
static int	RESIZE_FACTOR_3_4	Resize factor of 3_4
static int	RESIZE_FACTOR_NONE	No supported resize factors.

Constructor Summary

Constructors
Constructor and Description
DisplayConfiguration()

Method Summary

Methods inherited from class java.lang.Object

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

Field Detail

FRAMEBUFFER_CONFIGURATION_SERVERSIDE_ORIENTATION_SWITCH

```
public static final int FRAMEBUFFER_CONFIGURATION_SERVERSIDE_ORIENTATION_SWITCH
```

The server is capable of switching orientations, as instructed by an appropriate DeviceStatusRequest message.

This capability has been deprecated in MirrorLink 1.3. Your application should not attempt to use this capability if it supports MirrorLink 1.3 or above.

See Also:

[Constant Field Values](#)

FRAMEBUFFER_CONFIGURATION_SERVERSIDE_ROTATION

```
public static final int FRAMEBUFFER_CONFIGURATION_SERVERSIDE_ROTATION
```

The server is capable of rotating its framebuffer, as instructed by an appropriate DeviceStatusRequest message.

This capability has been deprecated in MirrorLink 1.3. Your application should not attempt to use this capability if it supports MirrorLink 1.3 or above.

See Also:

[Constant Field Values](#)

FRAMEBUFFER_CONFIGURATION_UPSCALING

```
public static final int FRAMEBUFFER_CONFIGURATION_UPSCALING
```

The server is capable of upscaling its framebuffer if its own dimensions are less than those of the client display, which are specified in the `VNCCClientDisplayConfiguration` structure.

This capability has been deprecated in MirrorLink 1.3 for servers. Server applications should not advertise this capability if they support MirrorLink 1.3 or above. Viewer applications which support MirrorLink 1.3 or above may still advertise this capability.

See Also:

[Constant Field Values](#)

FRAMEBUFFER_CONFIGURATION_DOWNSCALING

```
public static final int FRAMEBUFFER_CONFIGURATION_DOWNSCALING
```

The server is capable of downscaling its framebuffer if its own dimensions are greater than those of the client display, which are specified in the `VNCCClientDisplayConfiguration` structure.

This capability has been deprecated in MirrorLink 1.3 for viewers. Viewer applications should not advertise this capability if they support MirrorLink 1.3 or above. Server applications which support MirrorLink 1.3 or above may still advertise this capability.

See Also:

[Constant Field Values](#)

FRAMEBUFFER_CONFIGURATION_REPLACE_EMPTY_UPDATES

```
public static final int FRAMEBUFFER_CONFIGURATION_REPLACE_EMPTY_UPDATES
```

The server is capable of suppressing empty update rectangles, so that the viewer application does not have to deal with them.

This capability has been removed in MirrorLink 1.1. Your application should not attempt to use this flag if it supports MirrorLink 1.1 or above.

See Also:

[Constant Field Values](#)

FRAMEBUFFER_CONFIGURATION_SUPPORTS_FRAMEBUFFER_ALTERNATIVE_TEXT

```
public static final int FRAMEBUFFER_CONFIGURATION_SUPPORTS_FRAMEBUFFER_ALTERNATIVE_TEXT
```

The server supports FramebufferAlternativeText messages.

VNC SDKs do not support FramebufferAlternativeText messages. This feature has also been deprecated in MirrorLink 1.3. Your application should therefore not attempt to use this flag.

See Also:

[Constant Field Values](#)

PIXELFORMAT_SUPPORT_NONE

```
public static final int PIXELFORMAT_SUPPORT_NONE
```

Supports no pixel formats. For use only in the VNCClientDisplayConfiguration message, where it indicates that the transform encoding implementation cannot transcode between pixel formats.

Viewer applications should advertise support for either [PIXELFORMAT_SUPPORT_ARGB888_32](#), [PIXELFORMAT_SUPPORT_RGB565_16](#), or both, if they support MirrorLink 1.3 or above. Server applications should always support both of these pixel formats.

See Also:

[Constant Field Values](#)

PIXELFORMAT_SUPPORT_ARGB888_32

```
public static final int PIXELFORMAT_SUPPORT_ARGB888_32
```

Supports the ARGB32 pixel format.

Viewer applications should advertise support for either this pixel format, [PIXELFORMAT_SUPPORT_RGB565_16](#), or both, if they support MirrorLink 1.3 or above. Server applications should always support this pixel format.

See Also:

[Constant Field Values](#)

PIXELFORMAT_SUPPORT_ANY_32

```
public static final int PIXELFORMAT_SUPPORT_ANY_32
```

Supports any other 32-bit true color pixel formats. For use only in the VNCServerDisplayConfiguration message.

These pixel formats have been deprecated in MirrorLink 1.3. Your application should not attempt to use these pixel formats if it supports MirrorLink 1.3 or above.

See Also:[Constant Field Values](#)**PIXELFORMAT_SUPPORT_RGB888_32**

```
public static final int PIXELFORMAT_SUPPORT_RGB888_32
```

Supports the RGB888 pixel format.

This pixel format has been deprecated in MirrorLink 1.3. Your application should not attempt to use this pixel format if it supports MirrorLink 1.3 or above.

See Also:[Constant Field Values](#)**PIXELFORMAT_SUPPORT_ANY_24**

```
public static final int PIXELFORMAT_SUPPORT_ANY_24
```

Supports any other 24-bit true color pixel formats. For use only in the VNCServerDisplayConfiguration message

These pixel formats have been deprecated in MirrorLink 1.3. Your application should not attempt to use these pixel formats if it supports MirrorLink 1.3 or above.

See Also:[Constant Field Values](#)**PIXELFORMAT_SUPPORT_RGB565_16**

```
public static final int PIXELFORMAT_SUPPORT_RGB565_16
```

Supports the RGB565 pixel formats.

Viewer applications should advertise support for either this pixel format, `PIXELFORMAT_SUPPORT_ARGB888_32`, or both, if they support MirrorLink 1.3 or above. Server applications should always support this pixel format.

See Also:[Constant Field Values](#)**PIXELFORMAT_SUPPORT_RGB555_16**

```
public static final int PIXELFORMAT_SUPPORT_RGB555_16
```

Supports the RGB555 pixel formats.

These pixel formats have been deprecated in MirrorLink 1.3. Your application should not attempt to use these pixel formats if it supports MirrorLink 1.3 or above.

See Also:[Constant Field Values](#)**PIXELFORMAT_SUPPORT_RGB444_16**

```
public static final int PIXELFORMAT_SUPPORT_RGB444_16
```

Supports the RGB444 pixel formats.

These pixel formats have been deprecated in MirrorLink 1.3. Your application should not attempt to use these pixel formats if it supports

MirrorLink 1.3 or above.

See Also:

[Constant Field Values](#)

PIXELFORMAT_SUPPORT_RGB_343_16

```
public static final int PIXELFORMAT_SUPPORT_RGB_343_16
```

Supports the RGB343 pixel formats.

These pixel formats have been deprecated in MirrorLink 1.3. Your application should not attempt to use these pixel formats if it supports MirrorLink 1.3 or above.

See Also:

[Constant Field Values](#)

PIXELFORMAT_SUPPORT_ANY_16

```
public static final int PIXELFORMAT_SUPPORT_ANY_16
```

Supports any other 16-bit true color pixel formats.

These pixel formats have been deprecated in MirrorLink 1.3. Your application should not attempt to use these pixel formats if it supports MirrorLink 1.3 or above.

See Also:

[Constant Field Values](#)

PIXELFORMAT_SUPPORT_GRAYSCALE_16

```
public static final int PIXELFORMAT_SUPPORT_GRAYSCALE_16
```

Supports 16-bit grayscale.

This pixel format has been deprecated in MirrorLink 1.3. Your application should not attempt to use this pixel format if it supports MirrorLink 1.3 or above.

See Also:

[Constant Field Values](#)

PIXELFORMAT_SUPPORT_GRAYSCALE_8

```
public static final int PIXELFORMAT_SUPPORT_GRAYSCALE_8
```

Supports 8-bit grayscale.

This pixel format has been deprecated in MirrorLink 1.3. Your application should not attempt to use this pixel format if it supports MirrorLink 1.3 or above.

See Also:

[Constant Field Values](#)

RESIZE_FACTOR_NONE

```
public static final int RESIZE_FACTOR_NONE
```

No supported resize factors.

In MirrorLink 1.3 sessions and above, only this value should be used. This value should not be used in MirrorLink 1.2 sessions and below.

See Also:[Constant Field Values](#)**RESIZE_FACTOR_1_1**

```
public static final int RESIZE_FACTOR_1_1
```

Resize factor of 1/1

See Also:[Constant Field Values](#)**RESIZE_FACTOR_1_2**

```
public static final int RESIZE_FACTOR_1_2
```

Resize factor of 1/2

See Also:[Constant Field Values](#)**RESIZE_FACTOR_1_3**

```
public static final int RESIZE_FACTOR_1_3
```

Resize factor of 1/3

See Also:[Constant Field Values](#)**RESIZE_FACTOR_1_4**

```
public static final int RESIZE_FACTOR_1_4
```

Resize factor of 1/4

See Also:[Constant Field Values](#)**RESIZE_FACTOR_1_5**

```
public static final int RESIZE_FACTOR_1_5
```

Resize factor of 1/5

See Also:[Constant Field Values](#)**RESIZE_FACTOR_1_6**

```
public static final int RESIZE_FACTOR_1_6
```

Resize factor of 1/6

See Also:

[Constant Field Values](#)

RESIZE_FACTOR_1_8

```
public static final int RESIZE_FACTOR_1_8
```

Resize factor of 1/8

See Also:

[Constant Field Values](#)

RESIZE_FACTOR_1_10

```
public static final int RESIZE_FACTOR_1_10
```

Resize factor of 1/10

See Also:

[Constant Field Values](#)

RESIZE_FACTOR_1_16

```
public static final int RESIZE_FACTOR_1_16
```

Resize factor of 1/16

See Also:

[Constant Field Values](#)

RESIZE_FACTOR_1_32

```
public static final int RESIZE_FACTOR_1_32
```

Resize factor of 1/32

See Also:

[Constant Field Values](#)

RESIZE_FACTOR_2_3

```
public static final int RESIZE_FACTOR_2_3
```

Resize factor of 2_3

See Also:

[Constant Field Values](#)

RESIZE_FACTOR_3_4

```
public static final int RESIZE_FACTOR_3_4
```

Resize factor of 3_4

See Also:

[Constant Field Values](#)

Constructor Detail

DisplayConfiguration

```
public DisplayConfiguration()
```

com.realvnc.mirrorlink

Class EventConfiguration

java.lang.Object
com.realvnc.mirrorlink.EventConfiguration

Direct Known Subclasses:

VNCServerEventConfiguration

```
public abstract class EventConfiguration
extends java.lang.Object
```

Class defining constants for use inVNCServerEventConfiguration and VNCClientEventConfiguration classes.

For further information about each of these values consult the appropriate section of the MirrorLink specifications.

The KNOB_KEY_* constants are for use in the bitmasks used in VNCServerEventConfiguration.getKnobKeySupport() and VNCClientEventConfiguration.setKnobKeySupport(int).

The DEVICE_KEY_* constants are for use in the bitmasks used in VNCServerEventConfiguration.getDeviceKeySupport() and VNCClientEventConfiguration.setDeviceKeySupport(int).

The MULTIMEDIA_KEY_* constants are for use in the bitmasks used in VNCServerEventConfiguration.getMultimediaKeySupport() and VNCClientEventConfiguration.setMultimediaKeySupport(int).

The MISC_KEY_* constants are for use in the bitmasks used in VNCServerEventConfiguration.getMiscKeySupport() and VNCClientEventConfiguration.setMiscKeySupport(int). Some miscellaneous key support features have been deprecated in MirrorLink 1.3. Viewer and server applications should not attempt to use these features if they support MirrorLink 1.3 or above. See the individual MISC_KEY_* constants for further information.

The POINTER_* constants are for use in the bitmasks used in VNCServerEventConfiguration.getPointerSupport() and VNCClientEventConfiguration.setPointerSupport(int).

Field Summary

Fields	
Modifier and Type	Field and Description
static int	DEVICE_KEY_SUPPORT_ALL
static int	DEVICE_KEY_SUPPORT_APPLICATION
static int	DEVICE_KEY_SUPPORT_BACKWARD
static int	DEVICE_KEY_SUPPORT_CLEAR
static int	DEVICE_KEY_SUPPORT_DELETE
static int	DEVICE_KEY_SUPPORT_FORWARD
static int	DEVICE_KEY_SUPPORT_HOME
static int	DEVICE_KEY_SUPPORT_MENU
static int	DEVICE_KEY_SUPPORT_OK
static int	DEVICE_KEY_SUPPORT_PHONE_CALL
static int	DEVICE_KEY_SUPPORT_PHONE_END
static int	DEVICE_KEY_SUPPORT_SEARCH
static int	DEVICE_KEY_SUPPORT_SOFT_LEFT
static int	DEVICE_KEY_SUPPORT_SOFT_MIDDLE
static int	DEVICE_KEY_SUPPORT_SOFT_RIGHT
static int	DEVICE_KEY_SUPPORT_ZOOM_IN
static int	DEVICE_KEY_SUPPORT_ZOOM_OUT
static int	KNOB_KEY_SUPPORT_PULL_Z_0

static int	KNOB_KEY_SUPPORT_PULL_Z_1
static int	KNOB_KEY_SUPPORT_PULL_Z_2
static int	KNOB_KEY_SUPPORT_PULL_Z_3
static int	KNOB_KEY_SUPPORT_PUSH_Z_0
static int	KNOB_KEY_SUPPORT_PUSH_Z_1
static int	KNOB_KEY_SUPPORT_PUSH_Z_2
static int	KNOB_KEY_SUPPORT_PUSH_Z_3
static int	KNOB_KEY_SUPPORT_ROTATE_X_0
static int	KNOB_KEY_SUPPORT_ROTATE_X_1
static int	KNOB_KEY_SUPPORT_ROTATE_X_2
static int	KNOB_KEY_SUPPORT_ROTATE_X_3
static int	KNOB_KEY_SUPPORT_ROTATE_Y_0
static int	KNOB_KEY_SUPPORT_ROTATE_Y_1
static int	KNOB_KEY_SUPPORT_ROTATE_Y_2
static int	KNOB_KEY_SUPPORT_ROTATE_Y_3
static int	KNOB_KEY_SUPPORT_ROTATE_Z_0
static int	KNOB_KEY_SUPPORT_ROTATE_Z_1
static int	KNOB_KEY_SUPPORT_ROTATE_Z_2
static int	KNOB_KEY_SUPPORT_ROTATE_Z_3
static int	KNOB_KEY_SUPPORT_SHIFT_X_0
static int	KNOB_KEY_SUPPORT_SHIFT_X_1
static int	KNOB_KEY_SUPPORT_SHIFT_X_2
static int	KNOB_KEY_SUPPORT_SHIFT_X_3
static int	KNOB_KEY_SUPPORT_SHIFT_XY_0
static int	KNOB_KEY_SUPPORT_SHIFT_XY_1
static int	KNOB_KEY_SUPPORT_SHIFT_XY_2
static int	KNOB_KEY_SUPPORT_SHIFT_XY_3
static int	KNOB_KEY_SUPPORT_SHIFT_Y_0
static int	KNOB_KEY_SUPPORT_SHIFT_Y_1
static int	KNOB_KEY_SUPPORT_SHIFT_Y_2
static int	KNOB_KEY_SUPPORT_SHIFT_Y_3
static int	MISC_KEY_SUPPORT_EVENT_MAPPING Supports the Event Mapping feature.
static int	MISC_KEY_SUPPORT_FUNCTION_KEY_0 Supports Function Key 0.
static int	MISC_KEY_SUPPORT_FUNCTION_KEY_1 Supports Function Key 1.
static int	MISC_KEY_SUPPORT_FUNCTION_KEY_2 Supports Function Key 2.
static int	MISC_KEY_SUPPORT_FUNCTION_KEY_3 Supports Function Key 3.
static int	MISC_KEY_SUPPORT_FUNCTION_KEY_4 Supports Function Key 4.
static int	MISC_KEY_SUPPORT_FUNCTION_KEY_5 Supports Function Key 5.
static int	MISC_KEY_SUPPORT_FUNCTION_KEY_6 Supports Function Key 6.
static int	MISC_KEY_SUPPORT_FUNCTION_KEY_7 Supports Function Key 7.
static int	MISC_KEY_SUPPORT_FUNCTION_KEY_MASK

static int	MISC_KEY_SUPPORT_FUNCTION_KEY_SHIFT
static int	MISC_KEY_SUPPORT_ITU Supports ITU keypad events.
static int	MISC_KEY_SUPPORT_KEY_EVENT_LISTING Supports the Key Event Listing feature.
static int	MISC_KEY_SUPPORT_KEY_MAPPING_MASK
static int	MISC_KEY_SUPPORT_KEY_MAPPING_SHIFT
static int	MISC_KEY_SUPPORT_VIRTUAL_KEYBOARD_TRIGGER Supports the Virtual Keyboard Trigger feature.
static int	MULTIMEDIA_KEY_SUPPORT_FORWARD
static int	MULTIMEDIA_KEY_SUPPORT_MUTE
static int	MULTIMEDIA_KEY_SUPPORT_NEXT
static int	MULTIMEDIA_KEY_SUPPORT_PAUSE
static int	MULTIMEDIA_KEY_SUPPORT_PHOTO
static int	MULTIMEDIA_KEY_SUPPORT_PLAY
static int	MULTIMEDIA_KEY_SUPPORT_PREVIOUS
static int	MULTIMEDIA_KEY_SUPPORT_REWIND
static int	MULTIMEDIA_KEY_SUPPORT_STOP
static int	MULTIMEDIA_KEY_SUPPORT_UNMUTE
static int	POINTER_SUPPORT_POINTER_BUTTON_1
static int	POINTER_SUPPORT_POINTER_BUTTON_2
static int	POINTER_SUPPORT_POINTER_BUTTON_3
static int	POINTER_SUPPORT_POINTER_BUTTON_4
static int	POINTER_SUPPORT_POINTER_BUTTON_5
static int	POINTER_SUPPORT_POINTER_BUTTON_6
static int	POINTER_SUPPORT_POINTER_BUTTON_7
static int	POINTER_SUPPORT_POINTER_BUTTON_8
static int	POINTER_SUPPORT_POINTER_BUTTON_MASK
static int	POINTER_SUPPORT_POINTER_EVENTS
static int	POINTER_SUPPORT_TOUCH_COUNT_1
static int	POINTER_SUPPORT_TOUCH_COUNT_10
static int	POINTER_SUPPORT_TOUCH_COUNT_2
static int	POINTER_SUPPORT_TOUCH_COUNT_3
static int	POINTER_SUPPORT_TOUCH_COUNT_4
static int	POINTER_SUPPORT_TOUCH_COUNT_5
static int	POINTER_SUPPORT_TOUCH_COUNT_6
static int	POINTER_SUPPORT_TOUCH_COUNT_7
static int	POINTER_SUPPORT_TOUCH_COUNT_8
static int	POINTER_SUPPORT_TOUCH_COUNT_9
static int	POINTER_SUPPORT_TOUCH_COUNT_MASK
static int	POINTER_SUPPORT_TOUCH_COUNT_MASK_SHIFT
static int	POINTER_SUPPORT_TOUCH_EVENT_PRESSURE_MASK
static int	POINTER_SUPPORT_TOUCH_EVENT_PRESSURE_MASK_SHIFT
static int	POINTER_SUPPORT_TOUCH_EVENTS

Constructor Summary

Constructors

Constructor and Description
<code>EventConfiguration()</code>

Method Summary

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

Field Detail

Knob_Key_Support_Shift_X_0

```
public static final int Knob_Key_Support_Shift_X_0
```

See Also:
[Constant Field Values](#)

Knob_Key_Support_Shift_Y_0

```
public static final int Knob_Key_Support_Shift_Y_0
```

See Also:
[Constant Field Values](#)

Knob_Key_Support_Shift_XY_0

```
public static final int Knob_Key_Support_Shift_XY_0
```

See Also:
[Constant Field Values](#)

Knob_Key_Support_Push_Z_0

```
public static final int Knob_Key_Support_Push_Z_0
```

See Also:
[Constant Field Values](#)

Knob_Key_Support_Pull_Z_0

```
public static final int Knob_Key_Support_Pull_Z_0
```

See Also:
[Constant Field Values](#)

KNOB_KEY_SUPPORT_ROTATE_X_0

```
public static final int KNOB_KEY_SUPPORT_ROTATE_X_0
```

See Also:[Constant Field Values](#)**KNOB_KEY_SUPPORT_ROTATE_Y_0**

```
public static final int KNOB_KEY_SUPPORT_ROTATE_Y_0
```

See Also:[Constant Field Values](#)**KNOB_KEY_SUPPORT_ROTATE_Z_0**

```
public static final int KNOB_KEY_SUPPORT_ROTATE_Z_0
```

See Also:[Constant Field Values](#)**KNOB_KEY_SUPPORT_SHIFT_X_1**

```
public static final int KNOB_KEY_SUPPORT_SHIFT_X_1
```

See Also:[Constant Field Values](#)**KNOB_KEY_SUPPORT_SHIFT_Y_1**

```
public static final int KNOB_KEY_SUPPORT_SHIFT_Y_1
```

See Also:[Constant Field Values](#)**KNOB_KEY_SUPPORT_SHIFT_XY_1**

```
public static final int KNOB_KEY_SUPPORT_SHIFT_XY_1
```

See Also:[Constant Field Values](#)**KNOB_KEY_SUPPORT_PUSH_Z_1**

```
public static final int KNOB_KEY_SUPPORT_PUSH_Z_1
```

See Also:[Constant Field Values](#)**KNOB_KEY_SUPPORT_PULL_Z_1**

```
public static final int KNOB_KEY_SUPPORT_PULL_Z_1
```

See Also:[Constant Field Values](#)**KNOB_KEY_SUPPORT_ROTATE_X_1**

```
public static final int KNOB_KEY_SUPPORT_ROTATE_X_1
```

See Also:[Constant Field Values](#)**KNOB_KEY_SUPPORT_ROTATE_Y_1**

```
public static final int KNOB_KEY_SUPPORT_ROTATE_Y_1
```

See Also:[Constant Field Values](#)**KNOB_KEY_SUPPORT_ROTATE_Z_1**

```
public static final int KNOB_KEY_SUPPORT_ROTATE_Z_1
```

See Also:[Constant Field Values](#)**KNOB_KEY_SUPPORT_SHIFT_X_2**

```
public static final int KNOB_KEY_SUPPORT_SHIFT_X_2
```

See Also:[Constant Field Values](#)**KNOB_KEY_SUPPORT_SHIFT_Y_2**

```
public static final int KNOB_KEY_SUPPORT_SHIFT_Y_2
```

See Also:[Constant Field Values](#)**KNOB_KEY_SUPPORT_SHIFT_XY_2**

```
public static final int KNOB_KEY_SUPPORT_SHIFT_XY_2
```

See Also:[Constant Field Values](#)**KNOB_KEY_SUPPORT_PUSH_Z_2**

```
public static final int KNOB_KEY_SUPPORT_PUSH_Z_2
```

See Also:[Constant Field Values](#)**KNOB_KEY_SUPPORT_PULL_Z_2**

```
public static final int KNOB_KEY_SUPPORT_PULL_Z_2
```

See Also:[Constant Field Values](#)**KNOB_KEY_SUPPORT_ROTATE_X_2**

```
public static final int KNOB_KEY_SUPPORT_ROTATE_X_2
```

See Also:[Constant Field Values](#)**KNOB_KEY_SUPPORT_ROTATE_Y_2**

```
public static final int KNOB_KEY_SUPPORT_ROTATE_Y_2
```

See Also:[Constant Field Values](#)**KNOB_KEY_SUPPORT_ROTATE_Z_2**

```
public static final int KNOB_KEY_SUPPORT_ROTATE_Z_2
```

See Also:[Constant Field Values](#)**KNOB_KEY_SUPPORT_SHIFT_X_3**

```
public static final int KNOB_KEY_SUPPORT_SHIFT_X_3
```

See Also:[Constant Field Values](#)**KNOB_KEY_SUPPORT_SHIFT_Y_3**

```
public static final int KNOB_KEY_SUPPORT_SHIFT_Y_3
```

See Also:[Constant Field Values](#)**KNOB_KEY_SUPPORT_SHIFT_XY_3**

```
public static final int KNOB_KEY_SUPPORT_SHIFT_XY_3
```

See Also:

Constant Field Values

KNOB_KEY_SUPPORT_PUSH_Z_3

```
public static final int KNOB_KEY_SUPPORT_PUSH_Z_3
```

See Also:

[Constant Field Values](#)

KNOB_KEY_SUPPORT_PULL_Z_3

```
public static final int KNOB_KEY_SUPPORT_PULL_Z_3
```

See Also:

[Constant Field Values](#)

KNOB_KEY_SUPPORT_ROTATE_X_3

```
public static final int KNOB_KEY_SUPPORT_ROTATE_X_3
```

See Also:

[Constant Field Values](#)

KNOB_KEY_SUPPORT_ROTATE_Y_3

```
public static final int KNOB_KEY_SUPPORT_ROTATE_Y_3
```

See Also:

[Constant Field Values](#)

KNOB_KEY_SUPPORT_ROTATE_Z_3

```
public static final int KNOB_KEY_SUPPORT_ROTATE_Z_3
```

See Also:

[Constant Field Values](#)

DEVICE_KEY_SUPPORT_PHONE_CALL

```
public static final int DEVICE_KEY_SUPPORT_PHONE_CALL
```

See Also:

[Constant Field Values](#)

DEVICE_KEY_SUPPORT_PHONE_END

```
public static final int DEVICE_KEY_SUPPORT_PHONE_END
```

See Also:

[Constant Field Values](#)

DEVICE_KEY_SUPPORT_SOFT_LEFT

```
public static final int DEVICE_KEY_SUPPORT_SOFT_LEFT
```

See Also:[Constant Field Values](#)**DEVICE_KEY_SUPPORT_SOFT_MIDDLE**

```
public static final int DEVICE_KEY_SUPPORT_SOFT_MIDDLE
```

See Also:[Constant Field Values](#)**DEVICE_KEY_SUPPORT_SOFT_RIGHT**

```
public static final int DEVICE_KEY_SUPPORT_SOFT_RIGHT
```

See Also:[Constant Field Values](#)**DEVICE_KEY_SUPPORT_APPLICATION**

```
public static final int DEVICE_KEY_SUPPORT_APPLICATION
```

See Also:[Constant Field Values](#)**DEVICE_KEY_SUPPORT_OK**

```
public static final int DEVICE_KEY_SUPPORT_OK
```

See Also:[Constant Field Values](#)**DEVICE_KEY_SUPPORT_DELETE**

```
public static final int DEVICE_KEY_SUPPORT_DELETE
```

See Also:[Constant Field Values](#)**DEVICE_KEY_SUPPORT_ZOOM_IN**

```
public static final int DEVICE_KEY_SUPPORT_ZOOM_IN
```

See Also:[Constant Field Values](#)**DEVICE_KEY_SUPPORT_ZOOM_OUT**

```
public static final int DEVICE_KEY_SUPPORT_ZOOM_OUT
```

See Also:[Constant Field Values](#)**DEVICE_KEY_SUPPORT_CLEAR**

```
public static final int DEVICE_KEY_SUPPORT_CLEAR
```

See Also:[Constant Field Values](#)**DEVICE_KEY_SUPPORT_FORWARD**

```
public static final int DEVICE_KEY_SUPPORT_FORWARD
```

See Also:[Constant Field Values](#)**DEVICE_KEY_SUPPORT_BACKWARD**

```
public static final int DEVICE_KEY_SUPPORT_BACKWARD
```

See Also:[Constant Field Values](#)**DEVICE_KEY_SUPPORT_HOME**

```
public static final int DEVICE_KEY_SUPPORT_HOME
```

See Also:[Constant Field Values](#)**DEVICE_KEY_SUPPORT_SEARCH**

```
public static final int DEVICE_KEY_SUPPORT_SEARCH
```

See Also:[Constant Field Values](#)**DEVICE_KEY_SUPPORT_MENU**

```
public static final int DEVICE_KEY_SUPPORT_MENU
```

See Also:[Constant Field Values](#)**DEVICE_KEY_SUPPORT_ALL**

```
public static final int DEVICE_KEY_SUPPORT_ALL
```

See Also:[Constant Field Values](#)**MULTIMEDIA_KEY_SUPPORT_PLAY**

```
public static final int MULTIMEDIA_KEY_SUPPORT_PLAY
```

See Also:[Constant Field Values](#)**MULTIMEDIA_KEY_SUPPORT_PAUSE**

```
public static final int MULTIMEDIA_KEY_SUPPORT_PAUSE
```

See Also:[Constant Field Values](#)**MULTIMEDIA_KEY_SUPPORT_STOP**

```
public static final int MULTIMEDIA_KEY_SUPPORT_STOP
```

See Also:[Constant Field Values](#)**MULTIMEDIA_KEY_SUPPORT_FORWARD**

```
public static final int MULTIMEDIA_KEY_SUPPORT_FORWARD
```

See Also:[Constant Field Values](#)**MULTIMEDIA_KEY_SUPPORT_REWIND**

```
public static final int MULTIMEDIA_KEY_SUPPORT_REWIND
```

See Also:[Constant Field Values](#)**MULTIMEDIA_KEY_SUPPORT_NEXT**

```
public static final int MULTIMEDIA_KEY_SUPPORT_NEXT
```

See Also:[Constant Field Values](#)**MULTIMEDIA_KEY_SUPPORT_PREVIOUS**

```
public static final int MULTIMEDIA_KEY_SUPPORT_PREVIOUS
```

See Also:

[Constant Field Values](#)**MULTIMEDIA_KEY_SUPPORT_MUTE**

```
public static final int MULTIMEDIA_KEY_SUPPORT_MUTE
```

See Also:[Constant Field Values](#)**MULTIMEDIA_KEY_SUPPORT_UNMUTE**

```
public static final int MULTIMEDIA_KEY_SUPPORT_UNMUTE
```

See Also:[Constant Field Values](#)**MULTIMEDIA_KEY_SUPPORT_PHOTO**

```
public static final int MULTIMEDIA_KEY_SUPPORT_PHOTO
```

See Also:[Constant Field Values](#)**MISC_KEY_SUPPORT_ITU**

```
public static final int MISC_KEY_SUPPORT_ITU
```

Supports ITU keypad events.

These events have been deprecated in MirrorLink 1.3. Viewer and server applications should not attempt to support these events if they support MirrorLink 1.3 or above.

See Also:[Constant Field Values](#)**MISC_KEY_SUPPORT_VIRTUAL_KEYBOARD_TRIGGER**

```
public static final int MISC_KEY_SUPPORT_VIRTUAL_KEYBOARD_TRIGGER
```

Supports the Virtual Keyboard Trigger feature.

This feature has been deprecated in MirrorLink 1.3. Viewer and server applications should not attempt to use this feature if they support MirrorLink 1.3 or above.

See Also:[Constant Field Values](#)**MISC_KEY_SUPPORT_KEY_EVENT_LISTING**

```
public static final int MISC_KEY_SUPPORT_KEY_EVENT_LISTING
```

Supports the Key Event Listing feature.

This feature has been deprecated in MirrorLink 1.3. Viewer and server applications should not attempt to use this feature if they support MirrorLink 1.3 or above.

See Also:

[Constant Field Values](#)

MISC_KEY_SUPPORT_EVENT_MAPPING

`public static final int MISC_KEY_SUPPORT_EVENT_MAPPING`

Supports the Event Mapping feature.

This feature has been deprecated in MirrorLink 1.3 for clients. Viewer applications should not attempt to use this feature if they support MirrorLink 1.3 or above. Server applications must continue to enable this feature in the [VNCServerEventConfiguration](#).

See Also:

[Constant Field Values](#)

MISC_KEY_SUPPORT_FUNCTION_KEY_0

`public static final int MISC_KEY_SUPPORT_FUNCTION_KEY_0`

Supports Function Key 0.

See Also:

[Constant Field Values](#)

MISC_KEY_SUPPORT_FUNCTION_KEY_1

`public static final int MISC_KEY_SUPPORT_FUNCTION_KEY_1`

Supports Function Key 1.

See Also:

[Constant Field Values](#)

MISC_KEY_SUPPORT_FUNCTION_KEY_2

`public static final int MISC_KEY_SUPPORT_FUNCTION_KEY_2`

Supports Function Key 2.

See Also:

[Constant Field Values](#)

MISC_KEY_SUPPORT_FUNCTION_KEY_3

`public static final int MISC_KEY_SUPPORT_FUNCTION_KEY_3`

Supports Function Key 3.

See Also:

[Constant Field Values](#)

MISC_KEY_SUPPORT_FUNCTION_KEY_4

`public static final int MISC_KEY_SUPPORT_FUNCTION_KEY_4`

Supports Function Key 4.

See Also:

[Constant Field Values](#)

MISC_KEY_SUPPORT_FUNCTION_KEY_5

```
public static final int MISC_KEY_SUPPORT_FUNCTION_KEY_5
```

Supports Function Key 5.

See Also:

[Constant Field Values](#)

MISC_KEY_SUPPORT_FUNCTION_KEY_6

```
public static final int MISC_KEY_SUPPORT_FUNCTION_KEY_6
```

Supports Function Key 6.

See Also:

[Constant Field Values](#)

MISC_KEY_SUPPORT_FUNCTION_KEY_7

```
public static final int MISC_KEY_SUPPORT_FUNCTION_KEY_7
```

Supports Function Key 7.

See Also:

[Constant Field Values](#)

POINTER_SUPPORT_POINTER_EVENTS

```
public static final int POINTER_SUPPORT_POINTER_EVENTS
```

See Also:

[Constant Field Values](#)

POINTER_SUPPORT_TOUCH_EVENTS

```
public static final int POINTER_SUPPORT_TOUCH_EVENTS
```

See Also:

[Constant Field Values](#)

POINTER_SUPPORT_POINTER_BUTTON_1

```
public static final int POINTER_SUPPORT_POINTER_BUTTON_1
```

See Also:

[Constant Field Values](#)

POINTER_SUPPORT_POINTER_BUTTON_2

```
public static final int POINTER_SUPPORT_POINTER_BUTTON_2
```

See Also:[Constant Field Values](#)**POINTER_SUPPORT_POINTER_BUTTON_3**

```
public static final int POINTER_SUPPORT_POINTER_BUTTON_3
```

See Also:[Constant Field Values](#)**POINTER_SUPPORT_POINTER_BUTTON_4**

```
public static final int POINTER_SUPPORT_POINTER_BUTTON_4
```

See Also:[Constant Field Values](#)**POINTER_SUPPORT_POINTER_BUTTON_5**

```
public static final int POINTER_SUPPORT_POINTER_BUTTON_5
```

See Also:[Constant Field Values](#)**POINTER_SUPPORT_POINTER_BUTTON_6**

```
public static final int POINTER_SUPPORT_POINTER_BUTTON_6
```

See Also:[Constant Field Values](#)**POINTER_SUPPORT_POINTER_BUTTON_7**

```
public static final int POINTER_SUPPORT_POINTER_BUTTON_7
```

See Also:[Constant Field Values](#)**POINTER_SUPPORT_POINTER_BUTTON_8**

```
public static final int POINTER_SUPPORT_POINTER_BUTTON_8
```

See Also:[Constant Field Values](#)**POINTER_SUPPORT_POINTER_BUTTON_MASK**

```
public static final int POINTER_SUPPORT_POINTER_BUTTON_MASK
```

See Also:

[Constant Field Values](#)

POINTER_SUPPORT_TOUCH_COUNT_1

```
public static final int POINTER_SUPPORT_TOUCH_COUNT_1
```

See Also:

[Constant Field Values](#)

POINTER_SUPPORT_TOUCH_COUNT_2

```
public static final int POINTER_SUPPORT_TOUCH_COUNT_2
```

See Also:

[Constant Field Values](#)

POINTER_SUPPORT_TOUCH_COUNT_3

```
public static final int POINTER_SUPPORT_TOUCH_COUNT_3
```

See Also:

[Constant Field Values](#)

POINTER_SUPPORT_TOUCH_COUNT_4

```
public static final int POINTER_SUPPORT_TOUCH_COUNT_4
```

See Also:

[Constant Field Values](#)

POINTER_SUPPORT_TOUCH_COUNT_5

```
public static final int POINTER_SUPPORT_TOUCH_COUNT_5
```

See Also:

[Constant Field Values](#)

POINTER_SUPPORT_TOUCH_COUNT_6

```
public static final int POINTER_SUPPORT_TOUCH_COUNT_6
```

See Also:

[Constant Field Values](#)

POINTER_SUPPORT_TOUCH_COUNT_7

```
public static final int POINTER_SUPPORT_TOUCH_COUNT_7
```

See Also:[Constant Field Values](#)**POINTER_SUPPORT_TOUCH_COUNT_8**

```
public static final int POINTER_SUPPORT_TOUCH_COUNT_8
```

See Also:[Constant Field Values](#)**POINTER_SUPPORT_TOUCH_COUNT_9**

```
public static final int POINTER_SUPPORT_TOUCH_COUNT_9
```

See Also:[Constant Field Values](#)**POINTER_SUPPORT_TOUCH_COUNT_10**

```
public static final int POINTER_SUPPORT_TOUCH_COUNT_10
```

See Also:[Constant Field Values](#)**POINTER_SUPPORT_TOUCH_COUNT_MASK**

```
public static final int POINTER_SUPPORT_TOUCH_COUNT_MASK
```

See Also:[Constant Field Values](#)**POINTER_SUPPORT_TOUCH_COUNT_MASK_SHIFT**

```
public static final int POINTER_SUPPORT_TOUCH_COUNT_MASK_SHIFT
```

See Also:[Constant Field Values](#)**POINTER_SUPPORT_TOUCH_EVENT_PRESSURE_MASK**

```
public static final int POINTER_SUPPORT_TOUCH_EVENT_PRESSURE_MASK
```

See Also:[Constant Field Values](#)**POINTER_SUPPORT_TOUCH_EVENT_PRESSURE_MASK_SHIFT**

```
public static final int POINTER_SUPPORT_TOUCH_EVENT_PRESSURE_MASK_SHIFT
```

See Also:

Constant Field Values

MISC_KEY_SUPPORT_FUNCTION_KEY_MASK

```
public static final int MISC_KEY_SUPPORT_FUNCTION_KEY_MASK
```

See Also:

[Constant Field Values](#)

MISC_KEY_SUPPORT_FUNCTION_KEY_SHIFT

```
public static final int MISC_KEY_SUPPORT_FUNCTION_KEY_SHIFT
```

See Also:

[Constant Field Values](#)

MISC_KEY_SUPPORT_KEY_MAPPING_MASK

```
public static final int MISC_KEY_SUPPORT_KEY_MAPPING_MASK
```

See Also:

[Constant Field Values](#)

MISC_KEY_SUPPORT_KEY_MAPPING_SHIFT

```
public static final int MISC_KEY_SUPPORT_KEY_MAPPING_SHIFT
```

See Also:

[Constant Field Values](#)

Constructor Detail

EventConfiguration

```
public EventConfiguration()
```

com.realvnc.mirrorlink

Class VNCAudioBlockingNotification

java.lang.Object
com.realvnc.mirrorlink.VNCAudioBlockingNotification

```
public class VNCAudioBlockingNotification
extends java.lang.Object
```

Class holding an AudioBlockingNotification MirrorLink extension message to be sent to the server.

To send an AudioBlockingNotification, the viewer application should initialise the class and then call VNCViewer.sendAudioBlockingNotification.

Field Summary

Fields	
Modifier and Type	Field and Description
static int	REASON_APPLICATION_CATEGORY_NOT_ALLOWED The application's category has been disallowed (for example, by the driver distraction policy).
static int	REASON_APPLICATION_NOT_TRUSTED The server's trust in the application category that it reported is not sufficient to satisfy the viewer application.
static int	REASON_APPLICATION_UNIQUE_ID_NOT_ALLOWED The server application has been disallowed based on its unique ID.
static int	REASON_GLOBALLY_MUTED The user has muted all audio.
static int	REASON_STREAM_MUTED The user has muted a particular audio stream.
static int	REASON_UNBLOCK The application's blocked audio stream should be resumed.

Constructor Summary

Constructors	
Constructor and Description	
VNCAudioBlockingNotification (int applicationUniqueId, int reason)	Construct a VNCAudioBlockingNotification object.

Method Summary

Methods	
Modifier and Type	Method and Description
int	getApplicationUniqueId() Retrieves the unique ID of the blocked application.
int	getReason() Retrieves the reason for blocking the application.
java.lang.String	toString() Returns a string based representation of this object.

Methods inherited from class java.lang.Object

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

Field Detail**REASON_UNBLOCK**

```
public static final int REASON_UNBLOCK
```

The application's blocked audio stream should be resumed.

See Also:

[Constant Field Values](#)

REASON_APPLICATION_CATEGORY_NOT_ALLOWED

```
public static final int REASON_APPLICATION_CATEGORY_NOT_ALLOWED
```

The application's category has been disallowed (for example, by the driver distraction policy).

This reason has been deprecated in MirrorLink 1.3. Viewer applications should not specify this reason if they support MirrorLink 1.3 or above. Server applications supporting MirrorLink 1.3 or above should reinterpret this as [REASON_STREAM_MUTED](#).

See Also:

[Constant Field Values](#)

REASON_APPLICATION_NOT_TRUSTED

```
public static final int REASON_APPLICATION_NOT_TRUSTED
```

The server's trust in the application category that it reported is not sufficient to satisfy the viewer application.

This reason has been deprecated in MirrorLink 1.3. Viewer applications should not specify this reason if they support MirrorLink 1.3 or above. Server applications supporting MirrorLink 1.3 or above should reinterpret this as [REASON_STREAM_MUTED](#).

See Also:

[Constant Field Values](#)

REASON_APPLICATION_UNIQUE_ID_NOT_ALLOWED

```
public static final int REASON_APPLICATION_UNIQUE_ID_NOT_ALLOWED
```

The server application has been disallowed based on its unique ID.

This reason has been deprecated in MirrorLink 1.3. Viewer applications should not specify this reason if they support MirrorLink 1.3 or above. Server applications supporting MirrorLink 1.3 or above should reinterpret this as [REASON_STREAM_MUTED](#).

See Also:

[Constant Field Values](#)

REASON_GLOBALLY_MUTED

```
public static final int REASON_GLOBALLY_MUTED
```


The user has muted all audio.

See Also:

[Constant Field Values](#)

REASON_STREAM_MUTED

```
public static final int REASON_STREAM_MUTED
```

The user has muted a particular audio stream.

See Also:

[Constant Field Values](#)

Constructor Detail**VNCAudioBlockingNotification**

```
public VNCAudioBlockingNotification(int applicationUniqueId,  
                                     int reason)
```

Construct a VNCAudioBlockingNotification object.

Some audio blocking reasons have been deprecated in MirrorLink 1.3. Viewer applications should not specify these reasons if they support MirrorLink 1.3 or above. See the REASON_* constants for further information.

Parameters:

`applicationUniqueId` - The unique ID of the application that has been blocked.

The value should be taken from the audio negotiation with the MirrorLink UPnP server.

`reason` - The reason for the viewer application's decision to block the audio, as a bitfield made up of REASON_* constants. Depending on the reason, the server may choose to suspend playback of the audio stream.

A value of `REASON_UNBLOCK` indicates that the server should resume playback of the audio stream.

Method Detail**getApplicationUniqueId**

```
public int getApplicationUniqueId()
```

Retrieves the unique ID of the blocked application.

Returns:

The unique ID of the application that has been blocked.

getReason

```
public int getReason()
```

Retrieves the reason for blocking the application.

Returns:

The reason for the viewer application's decision to block the audio, as a bitfield made up of REASON_* constants.

toString

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

Returns a string based representation of this object. This includes the application unique ID and the reason.

Overrides:

toString in class java.lang.Object

Returns:

A string representation of this object.

com.realvnc.mirrorlink

Class VNCAudioInfo

java.lang.Object
com.realvnc.mirrorlink.VNCAudioInfo

```
public class VNCAudioInfo
extends java.lang.Object
```

Class containing constants to be used when defining audio information for an application.

Field Summary

Fields	
Modifier and Type	Field and Description
static int	AUDIO_CONTENT_CATEGORY_MEDIA_AUDIO_IN Audio content category representing media input.
static int	AUDIO_CONTENT_CATEGORY_MEDIA_AUDIO_OUT Audio content category representing media output.
static int	AUDIO_CONTENT_CATEGORY_MISC Audio content category representing miscellaneous audio.
static int	AUDIO_CONTENT_CATEGORY_PHONE_AUDIO Audio content category representing phone audio.
static int	AUDIO_CONTENT_CATEGORY_UNKNOWN Audio content category representing unknown content.
static int	AUDIO_CONTENT_CATEGORY_VOICE_COMMAND_IN Audio content category representing voice command input.
static int	AUDIO_CONTENT_CATEGORY_VOICE_COMMAND_OUT Audio content category representing voice command output.

Constructor Summary

Constructors	
Constructor and Description	
VNCAudioInfo()	

Method Summary

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

Field Detail

AUDIO_CONTENT_CATEGORY_UNKNOWN	
public static final int AUDIO_CONTENT_CATEGORY_UNKNOWN	

Audio content category representing unknown content.

See Also:

[Constant Field Values](#)

AUDIO_CONTENT_CATEGORY_PHONE_AUDIO

```
public static final int AUDIO_CONTENT_CATEGORY_PHONE_AUDIO
```

Audio content category representing phone audio.

See Also:

[Constant Field Values](#)

AUDIO_CONTENT_CATEGORY_MEDIA_AUDIO_OUT

```
public static final int AUDIO_CONTENT_CATEGORY_MEDIA_AUDIO_OUT
```

Audio content category representing media output.

See Also:

[Constant Field Values](#)

AUDIO_CONTENT_CATEGORY_MEDIA_AUDIO_IN

```
public static final int AUDIO_CONTENT_CATEGORY_MEDIA_AUDIO_IN
```

Audio content category representing media input.

See Also:

[Constant Field Values](#)

AUDIO_CONTENT_CATEGORY_VOICE_COMMAND_OUT

```
public static final int AUDIO_CONTENT_CATEGORY_VOICE_COMMAND_OUT
```

Audio content category representing voice command output.

See Also:

[Constant Field Values](#)

AUDIO_CONTENT_CATEGORY_VOICE_COMMAND_IN

```
public static final int AUDIO_CONTENT_CATEGORY_VOICE_COMMAND_IN
```

Audio content category representing voice command input.

See Also:

[Constant Field Values](#)

AUDIO_CONTENT_CATEGORY_MISC

```
public static final int AUDIO_CONTENT_CATEGORY_MISC
```

Audio content category representing miscellaneous audio.

See Also:

[Constant Field Values](#)

Constructor Detail

VNCAudioInfo

```
public VNCAudioInfo()
```

com.realvnc.mirrorlink

Class VNCClientDisplayConfiguration

java.lang.Object

com.realvnc.mirrorlink.DisplayConfiguration
com.realvnc.mirrorlink.VNCClientDisplayConfiguration

```
public class VNCClientDisplayConfiguration
extends DisplayConfiguration
```

Class holding a ClientDisplayConfiguration MirrorLink extension message to be sent to the server.

The SDK pre-fills the VNCClientDisplayConfiguration object with sensible default values before sending it to the server. A viewer application is free to override these values, but it is the viewer application's responsibility to ensure that the overridden values are sane.

Field Summary

Fields inherited from class com.realvnc.mirrorlink.DisplayConfiguration

FRAMEBUFFER_CONFIGURATION_DOWNSCALING, FRAMEBUFFER_CONFIGURATION_REPLACE_EMPTY_UPDATES, FRAMEBUFFER_CONFIGURATION_SERVERSIDE_ORIENTATION_SWITCH, FRAMEBUFFER_CONFIGURATION_SERVERSIDE_ROTATION, FRAMEBUFFER_CONFIGURATION_SUPPORTS_FRAMEBUFFER_ALTERNATIVE_TEXT, FRAMEBUFFER_CONFIGURATION_UPSCALING, PIXELFORMAT_SUPPORT_ANY_16, PIXELFORMAT_SUPPORT_ANY_24, PIXELFORMAT_SUPPORT_ANY_32, PIXELFORMAT_SUPPORT_ARGB888_32, PIXELFORMAT_SUPPORT_GRAYSCALE_16, PIXELFORMAT_SUPPORT_GRAYSCALE_8, PIXELFORMAT_SUPPORT_NONE, PIXELFORMAT_SUPPORT_RGB_343_16, PIXELFORMAT_SUPPORT_RGB444_16, PIXELFORMAT_SUPPORT_RGB555_16, PIXELFORMAT_SUPPORT_RGB565_16, PIXELFORMAT_SUPPORT_RGB888_32, RESIZE_FACTOR_1_1, RESIZE_FACTOR_1_10, RESIZE_FACTOR_1_16, RESIZE_FACTOR_1_2, RESIZE_FACTOR_1_3, RESIZE_FACTOR_1_32, RESIZE_FACTOR_1_4, RESIZE_FACTOR_1_5, RESIZE_FACTOR_1_6, RESIZE_FACTOR_1_8, RESIZE_FACTOR_2_3, RESIZE_FACTOR_3_4, RESIZE_FACTOR_NONE

Constructor Summary

Constructors

Constructor and Description
<div>VNCClientDisplayConfiguration(int clientMajorVersion, int clientMinorVersion, int framebufferConfiguration, int clientDisplayWidthPixels, int clientDisplayHeightPixels, int clientDisplayWidthMillimeters, int clientDisplayHeightMillimeters, int clientDistanceFromUserMillimeters, int pixelFormatSupport, int resizeFactors)</div> <div>Constructs a new client display configuration object.</div>

Method Summary

Methods

Modifier and Type	Method and Description
int	<div>getClientDisplayHeightMillimeters()</div> <div>Retrieves the display height size.</div>
int	<div>getClientDisplayHeightPixels()</div> <div>Retrieves the display pixel height.</div>
int	<div>getClientDisplayWidthMillimeters()</div> <div>Retrieves the display width size.</div>
int	<div>getClientDisplayWidthPixels()</div> <div>Retrieves the display pixel width.</div>

int	<code>getClientDistanceFromUserMillimeters()</code> Retrieves the expected distance from user.
int	<code>getClientMajorVersion()</code> Retrieves the major client version.
int	<code>getClientMinorVersion()</code> Retrieves the minor client version.
int	<code>getFramebufferConfiguration()</code> Retrieves the framebuffer configuration.
int	<code>getPixelFormatSupport()</code> Returns the pixel formats supported by the client for use with the Transform encoding.
int	<code>getResizeFactors()</code> Returns the resize factors supported by the client for use with the Transform encoding.
void	<code>setClientDisplayHeightMillimeters(int clientDisplayHeightMillimeters)</code> Sets the display height size.
void	<code>setClientDisplayHeightPixels(int clientDisplayHeightPixels)</code> Sets the display height.
void	<code>setClientDisplayWidthMillimeters(int clientDisplayWidthMillimeters)</code> Sets the display width size.
void	<code>setClientDisplayWidthPixels(int clientDisplayWidthPixels)</code> Sets the display pixel width.
void	<code>setClientDistanceFromUserMillimeters(int clientDistanceFromUserMillimeters)</code> Sets the expected distance from user.
void	<code>setClientMajorVersion(int clientMajorVersion)</code> Sets the major client version.
void	<code>setClientMinorVersion(int clientMinorVersion)</code> Sets the minor client version.
void	<code>setFramebufferConfiguration(int framebufferConfiguration)</code> Sets the framebuffer configuration.
void	<code>setPixelFormatSupport(int pixelFormatSupport)</code> Sets the pixel formats supported by the client.
void	<code>setResizeFactors(int resizeFactors)</code> Sets the resize factors supported by the client for use with the Transform encoding.
java.lang.String	<code>toString()</code> Returns a string based representation of this object.

Methods inherited from class java.lang.Object

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

Constructor Detail

VNCCClientDisplayConfiguration

```
public VNCCClientDisplayConfiguration(int clientMajorVersion,
                                     int clientMinorVersion,
                                     int framebufferConfiguration,
                                     int clientDisplayWidthPixels,
                                     int clientDisplayHeightPixels,
                                     int clientDisplayWidthMillimeters,
                                     int clientDisplayHeightMillimeters,
                                     int clientDistanceFromUserMillimeters,
                                     int pixelFormatSupport,
                                     int resizeFactors)
```

Constructs a new client display configuration object.

Note that many of these fields have been deprecated in MirrorLink 1.3. If a client display configuration message has its MirrorLink

version set to 1.3 or above, only the following values are allowed by the specification:

- **framebufferConfiguration**: 0x0000 or `DisplayConfiguration.FRAMEBUFFER_CONFIGURATION_UPSCALING`
- **pixelFormatSupport**: `DisplayConfiguration.PIXELFORMAT_SUPPORT_RGB565_16`, `DisplayConfiguration.PIXELFORMAT_SUPPORT_ARGB888_32`, or both
- **resizeFactors**: `DisplayConfiguration.RESIZE_FACTOR_NONE`

Parameters:

`clientMajorVersion` - The major client version.

`clientMinorVersion` - The minor client version.

`framebufferConfiguration` - A bitfield of framebuffer configuration.

`clientDisplayWidthPixels` - The display pixel height.

`clientDisplayHeightPixels` - The display pixel width.

`clientDisplayWidthMillimeters` - The display width size.

`clientDisplayHeightMillimeters` - The display height size.

`clientDistanceFromUserMillimeters` - The expected distance from user.

`pixelFormatSupport` - Pixel formats supported by client.

`resizeFactors` - Resize factors supported by client.

Method Detail

getClientMajorVersion

```
public int getClientMajorVersion()
```

Retrieves the major client version.

Returns:

The major version number of the MirrorLink specification followed by the client.

setClientMajorVersion

```
public void setClientMajorVersion(int clientMajorVersion)
```

Sets the major client version.

The SDK pre-fills this to the major version advertised by the server. If the version advertised by the server is newer than the version supported by the SDK, this will be limited to the maximum SDK-supported version.

Parameters:

`clientMajorVersion` - The major version number of the MirrorLink Mode followed by the client.

getClientMinorVersion

```
public int getClientMinorVersion()
```

Retrieves the minor client version.

Returns:

The minor version number of the MirrorLink specification followed by the client.

setClientMinorVersion

```
public void setClientMinorVersion(int clientMinorVersion)
```

Sets the minor client version.

The SDK pre-fills this to the minor version advertised by the server. If the version advertised by the server is newer than the version supported by the SDK, this will be limited to the maximum SDK-supported version.

Parameters:

`clientMinorVersion` - The minor version number of the MirrorLink specification followed by the client.

getFramebufferConfiguration

```
public int getFramebufferConfiguration()
```

Retrieves the framebuffer configuration.

Returns:

A bitfield made up of `FRAMEBUFFER_CONFIGURATION_*` values describing which of the server's advertised framebuffer capabilities it intends to use.

setFramebufferConfiguration

```
public void setFramebufferConfiguration(int framebufferConfiguration)
```

Sets the framebuffer configuration.

The SDK pre-fills this to `FRAMEBUFFER_CONFIGURATION_UPSCALING | FRAMEBUFFER_CONFIGURATION_DOWNSCALING` with the addition of `FRAMEBUFFER_CONFIGURATION_SERVERSIDE_ORIENTATION_SWITCH` if the server has advertised that it supports it.

Some capabilities have been deprecated in MirrorLink 1.3. Your application should not attempt to use these capabilities if it supports MirrorLink 1.3 or above. See [DisplayConfiguration](#) for further information.

Parameters:

`framebufferConfiguration` - A bitfield made up of `FRAMEBUFFER_CONFIGURATION_*` values describing which of the server's advertised framebuffer capabilities it intends to use.

getClientDisplayWidthPixels

```
public int getClientDisplayWidthPixels()
```

Retrieves the display pixel width.

Returns:

The width in pixels of the area in which the viewer application will display the framebuffer.

setClientDisplayWidthPixels

```
public void setClientDisplayWidthPixels(int clientDisplayWidthPixels)
```

Sets the display pixel width.

The SDK pre-fills this to the width of the server display in pixels. However, you should always override this with the actual width of the viewer-side display, if at all possible. Note that MirrorLink 1.3 requires a minimum of 800 for this value.

Parameters:

`clientDisplayWidthPixels` - The width in pixels of the area in which the viewer application will display the framebuffer.

getClientDisplayHeightPixels

```
public int getClientDisplayHeightPixels()
```

Retrieves the display pixel height.

Returns:

The height in pixels of the area in which the viewer application will display the framebuffer.

setClientDisplayHeightPixels

```
public void setClientDisplayHeightPixels(int clientDisplayHeightPixels)
```

Sets the display height.

The SDK pre-fills this to the height of the server display in pixels. However, you should always override this with the actual height of the viewer-side display, if at all possible. Note that MirrorLink 1.3 requires a minimum of 480 for this value.

Parameters:

`clientDisplayHeightPixels` - The height in pixels of the area in which the viewer application will display the framebuffer.

getClientDisplayWidthMillimeters

```
public int getClientDisplayWidthMillimeters()
```

Retrieves the display width size.

Returns:

The width in millimeters of the area in which the viewer application will display the framebuffer.

setClientDisplayWidthMillimeters

```
public void setClientDisplayWidthMillimeters(int clientDisplayWidthMillimeters)
```

Sets the display width size.

The SDK pre-fills this to 0.

Parameters:

`clientDisplayWidthMillimeters` - The width in millimeters of the area in which the viewer application will display the framebuffer.

getClientDisplayHeightMillimeters

```
public int getClientDisplayHeightMillimeters()
```

Retrieves the display height size.

Returns:

The height in millimeters of the area in which the viewer application will display the framebuffer.

setClientDisplayHeightMillimeters

```
public void setClientDisplayHeightMillimeters(int clientDisplayHeightMillimeters)
```

Sets the display height size.

The SDK pre-fills this to 0.

Parameters:

`clientDisplayHeightMillimeters` - The height in millimeters of the area in which the viewer application will display the framebuffer.

getClientDistanceFromUserMillimeters

```
public int getClientDistanceFromUserMillimeters()
```

Retrieves the expected distance from user.

Returns:

The expected distance between the viewer display and the user in millimeters.

setClientDistanceFromUserMillimeters

```
public void setClientDistanceFromUserMillimeters(int clientDistanceFromUserMillimeters)
```

Sets the expected distance from user.

The SDK pre-fills this to 0. In-car applications may be able to provide the server with a good estimate for this value.

Parameters:

`clientDistanceFromUserMillimeters` - The expected distance between the viewer display and the user in millimeters.

getPixelFormatSupport

```
public int getPixelFormatSupport()
```

Returns the pixel formats supported by the client for use with the Transform encoding.

Returns:

a bitfield made up of `PIXEL_FORMAT_SUPPORT_*` values indicating the pixel formats supported by the client.

setPixelFormatSupport

```
public void setPixelFormatSupport(int pixelFormatSupport)
```

Sets the pixel formats supported by the client.

Some pixel formats have been deprecated in MirrorLink 1.3. Your application should not attempt to use these pixel formats if it supports MirrorLink 1.3 or above. See [DisplayConfiguration](#) for further information.

Parameters:

`pixelFormatSupport` - a bitfield made up of `PIXELFORMAT_SUPPORT_*` values indicating the pixel formats supported by the client.

getResizeFactors

```
public int getResizeFactors()
```

Returns the resize factors supported by the client for use with the Transform encoding.

Returns:

a bitfield made up of `RESIZE_FACTOR_*` values indicating the resize factors supported by the client.

setResizeFactors

```
public void setResizeFactors(int resizeFactors)
```

Sets the resize factors supported by the client for use with the Transform encoding.

The Transform encoding has been deprecated in MirrorLink 1.3. Your application should not support any resize factors in MirrorLink 1.3 sessions or above. See [DisplayConfiguration](#) for further information.

Parameters:

`resizeFactors` - a bitfield made up of `RESIZE_FACTOR_*` values indicating the resize factors supported by the client.

toString

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

Returns a string based representation of this object.

Overrides:

`toString` in class `java.lang.Object`

Returns:

A string representation of this object.

com.realvnc.mirrorlink

Class VNCClientEventConfiguration

java.lang.Object

com.realvnc.mirrorlink.EventConfiguration
com.realvnc.mirrorlink.VNCServerEventConfiguration
com.realvnc.mirrorlink.VNCClientEventConfiguration

```
public class VNCClientEventConfiguration
extends VNCServerEventConfiguration
```

Class holding a ClientEventConfiguration MirrorLink extension message to be sent to the server.

This class is identical to VNCServerEventConfiguration, except that the fields indicate which of the server's advertised features the client wishes to use.

The SDK pre-fills the VNCClientEventConfiguration object with sensible default values before sending it to the server. A viewer application is free to override these values, but it is the viewer application's responsibility to ensure that the overridden values are sane.

Viewer applications that support MirrorLink 1.1 or above, and do not support pointer events, are required to support certain knob key events. These viewer applications should set the following knob key support bits in the event configuration:

- EventConfiguration.KNOB_KEY_SUPPORT_SHIFT_X_0
- EventConfiguration.KNOB_KEY_SUPPORT_SHIFT_Y_0
- EventConfiguration.KNOB_KEY_SUPPORT_PUSH_Z_0
- EventConfiguration.KNOB_KEY_SUPPORT_ROTATE_Z_0

From MirrorLink 1.3 onwards, viewer applications not supporting pointer events must additionally support the back device key, and so should also set the EventConfiguration.DEVICE_KEY_SUPPORT_BACKWARD device key support bit in the event configuration.

Some miscellaneous key support features have been deprecated in MirrorLink 1.3. Viewer applications should not attempt to use these features if they support MirrorLink 1.3 or above. See EventConfiguration for further information.

Field Summary

Fields inherited from class com.realvnc.mirrorlink.VNCServerEventConfiguration

deviceKeySupport, keyboardCountry, keyboardLanguage, knobKeySupport, miscKeySupport, multimediaKeySupport, pointerSupport, uiCountry, uiLanguage

Fields inherited from class com.realvnc.mirrorlink.EventConfiguration

DEVICE_KEY_SUPPORT_ALL, DEVICE_KEY_SUPPORT_APPLICATION, DEVICE_KEY_SUPPORT_BACKWARD, DEVICE_KEY_SUPPORT_CLEAR, DEVICE_KEY_SUPPORT_DELETE, DEVICE_KEY_SUPPORT_FORWARD, DEVICE_KEY_SUPPORT_HOME, DEVICE_KEY_SUPPORT_MENU, DEVICE_KEY_SUPPORT_OK, DEVICE_KEY_SUPPORT_PHONE_CALL, DEVICE_KEY_SUPPORT_PHONE_END, DEVICE_KEY_SUPPORT_SEARCH, DEVICE_KEY_SUPPORT_SOFT_LEFT, DEVICE_KEY_SUPPORT_SOFT_MIDDLE, DEVICE_KEY_SUPPORT_SOFT_RIGHT, DEVICE_KEY_SUPPORT_ZOOM_IN, DEVICE_KEY_SUPPORT_ZOOM_OUT, KNOB_KEY_SUPPORT_PULL_Z_0, KNOB_KEY_SUPPORT_PULL_Z_1, KNOB_KEY_SUPPORT_PULL_Z_2, KNOB_KEY_SUPPORT_PULL_Z_3, KNOB_KEY_SUPPORT_PUSH_Z_0, KNOB_KEY_SUPPORT_PUSH_Z_1, KNOB_KEY_SUPPORT_PUSH_Z_2, KNOB_KEY_SUPPORT_PUSH_Z_3, KNOB_KEY_SUPPORT_ROTATE_X_0, KNOB_KEY_SUPPORT_ROTATE_X_1, KNOB_KEY_SUPPORT_ROTATE_X_2, KNOB_KEY_SUPPORT_ROTATE_X_3, KNOB_KEY_SUPPORT_ROTATE_Y_0, KNOB_KEY_SUPPORT_ROTATE_Y_1, KNOB_KEY_SUPPORT_ROTATE_Y_2, KNOB_KEY_SUPPORT_ROTATE_Y_3, KNOB_KEY_SUPPORT_ROTATE_Z_0, KNOB_KEY_SUPPORT_ROTATE_Z_1, KNOB_KEY_SUPPORT_ROTATE_Z_2, KNOB_KEY_SUPPORT_ROTATE_Z_3, KNOB_KEY_SUPPORT_SHIFT_X_0, KNOB_KEY_SUPPORT_SHIFT_X_1, KNOB_KEY_SUPPORT_SHIFT_X_2, KNOB_KEY_SUPPORT_SHIFT_X_3, KNOB_KEY_SUPPORT_SHIFT_XY_0, KNOB_KEY_SUPPORT_SHIFT_XY_1, KNOB_KEY_SUPPORT_SHIFT_XY_2, KNOB_KEY_SUPPORT_SHIFT_XY_3, KNOB_KEY_SUPPORT_SHIFT_Y_0, KNOB_KEY_SUPPORT_SHIFT_Y_1, KNOB_KEY_SUPPORT_SHIFT_Y_2, KNOB_KEY_SUPPORT_SHIFT_Y_3, MISC_KEY_SUPPORT_EVENT_MAPPING, MISC_KEY_SUPPORT_FUNCTION_KEY_0, MISC_KEY_SUPPORT_FUNCTION_KEY_1, MISC_KEY_SUPPORT_FUNCTION_KEY_2, MISC_KEY_SUPPORT_FUNCTION_KEY_3, MISC_KEY_SUPPORT_FUNCTION_KEY_4, MISC_KEY_SUPPORT_FUNCTION_KEY_5, MISC_KEY_SUPPORT_FUNCTION_KEY_6, MISC_KEY_SUPPORT_FUNCTION_KEY_7, MISC_KEY_SUPPORT_FUNCTION_KEY_MASK, MISC_KEY_SUPPORT_FUNCTION_KEY_SHIFT, MISC_KEY_SUPPORT_ITU, MISC_KEY_SUPPORT_KEY_EVENT_LISTING, MISC_KEY_SUPPORT_KEY_MAPPING_MASK, MISC_KEY_SUPPORT_KEY_MAPPING_SHIFT, MISC_KEY_SUPPORT_VIRTUAL_KEYBOARD_TRIGGER, MULTIMEDIA_KEY_SUPPORT_FORWARD,

```
MULTIMEDIA_KEY_SUPPORT_MUTE, MULTIMEDIA_KEY_SUPPORT_NEXT, MULTIMEDIA_KEY_SUPPORT_PAUSE,
MULTIMEDIA_KEY_SUPPORT_PHOTO, MULTIMEDIA_KEY_SUPPORT_PLAY, MULTIMEDIA_KEY_SUPPORT_PREVIOUS,
MULTIMEDIA_KEY_SUPPORT_REWIND, MULTIMEDIA_KEY_SUPPORT_STOP, MULTIMEDIA_KEY_SUPPORT_UNMUTE,
POINTER_SUPPORT_POINTER_BUTTON_1, POINTER_SUPPORT_POINTER_BUTTON_2, POINTER_SUPPORT_POINTER_BUTTON_3,
POINTER_SUPPORT_POINTER_BUTTON_4, POINTER_SUPPORT_POINTER_BUTTON_5, POINTER_SUPPORT_POINTER_BUTTON_6,
POINTER_SUPPORT_POINTER_BUTTON_7, POINTER_SUPPORT_POINTER_BUTTON_8,
POINTER_SUPPORT_POINTER_BUTTON_MASK, POINTER_SUPPORT_POINTER_EVENTS, POINTER_SUPPORT_TOUCH_COUNT_1,
POINTER_SUPPORT_TOUCH_COUNT_10, POINTER_SUPPORT_TOUCH_COUNT_2, POINTER_SUPPORT_TOUCH_COUNT_3,
POINTER_SUPPORT_TOUCH_COUNT_4, POINTER_SUPPORT_TOUCH_COUNT_5, POINTER_SUPPORT_TOUCH_COUNT_6,
POINTER_SUPPORT_TOUCH_COUNT_7, POINTER_SUPPORT_TOUCH_COUNT_8, POINTER_SUPPORT_TOUCH_COUNT_9,
POINTER_SUPPORT_TOUCH_COUNT_MASK, POINTER_SUPPORT_TOUCH_COUNT_MASK_SHIFT,
POINTER_SUPPORT_TOUCH_EVENT_PRESSURE_MASK, POINTER_SUPPORT_TOUCH_EVENT_PRESSURE_MASK_SHIFT,
POINTER_SUPPORT_TOUCH_EVENTS
```

Constructor Summary

Constructors

Constructor and Description

VNCClientEventConfiguration(java.lang.String keyboardLanguage, java.lang.String keyboardCountry, java.lang.String uiLanguage, java.lang.String uiCountry, int knobKeySupport, int deviceKeySupport, int multimediaKeySupport, int miscKeySupport, int pointerSupport)

Creates a new client event configuration object.

Method Summary

Methods

Modifier and Type	Method and Description
void	setDeviceKeySupport (int deviceKeySupport) Sets the device key support to the provided bitmask.
void	setKeyboardCountry (java.lang.String keyboardCountry) Sets the keyboard layout country code.
void	setKeyboardLanguage (java.lang.String keyboardLanguage) Sets the keyboard layout language code.
void	setKnobKeySupport (int knobKeySupport) Sets the knob key support to the provided bitmask.
void	setMiscKeySupport (int miscKeySupport) Sets the miscellaneous key support to the provided bitmask.
void	setMultimediaKeySupport (int multimediaKeySupport) Sets the multimedia key support to the provided bitmask.
void	setPointerSupport (int pointerSupport) Sets the pointer / touchscreen support to the provided bitmap.
void	setUiCountry (java.lang.String uiCountry) Sets the user interface country code.
void	setUiLanguage (java.lang.String uiLanguage) Sets the user interface language code.

Methods inherited from class com.realvnc.mirrorlink.VNCServerEventConfiguration

getDeviceKeySupport, getKeyboardCountry, getKeyboardLanguage, getKnobKeySupport, getMiscKeySupport, getMultimediaKeySupport, getNumFunctionKeysSupported, getPointerSupport, getUiCountry, getUiLanguage, isEventMappingSupported, isITUKeySupported, isKeyEventListingSupported, isVirtualKeyboardTriggerSupported, toString

Methods inherited from class java.lang.Object

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

Constructor Detail

VNCClientEventConfiguration

```
public VNCClientEventConfiguration(java.lang.String keyboardLanguage,  
                                   java.lang.String keyboardCountry,  
                                   java.lang.String uiLanguage,  
                                   java.lang.String uiCountry,  
                                   int knobKeySupport,  
                                   int deviceKeySupport,  
                                   int multimediaKeySupport,  
                                   int miscKeySupport,  
                                   int pointerSupport)
```

Creates a new client event configuration object.

Parameters:

`keyboardLanguage` - The keyboard layout language code.

`keyboardCountry` - The keyboard layout country code.

`uiLanguage` - The user interface language code.

`uiCountry` - The user interface country code.

`knobKeySupport` - The knob key support, as a bitmask.

`deviceKeySupport` - The device key support, as a bitmask.

`multimediaKeySupport` - The multimedia key support, as a bitmask.

`miscKeySupport` - The miscellaneous key support, as a bitmask.

`pointerSupport` - The pointer / touchscreen support, as a bitmask.

Method Detail

setKeyboardLanguage

```
public void setKeyboardLanguage(java.lang.String keyboardLanguage)
```

Sets the keyboard layout language code. The SDK pre-fills this to the value of `keyboardLanguage` in the `VNCServerEventConfiguration`.

Parameters:

`keyboardLanguage` - The ISO 639-1 language code for the server's keyboard layout.

setKeyboardCountry

```
public void setKeyboardCountry(java.lang.String keyboardCountry)
```

Sets the keyboard layout country code. The SDK pre-fills this to the value of `keyboardCountry` in the `VNCServerEventConfiguration`.

Parameters:

`keyboardCountry` - The ISO 3166-1 country code for the server's keyboard layout.

setUiLanguage

```
public void setUiLanguage(java.lang.String uiLanguage)
```

Sets the user interface language code. The SDK pre-fills this to the value of uiLanguage in the VNCTServerEventConfiguration.

Parameters:

uiLanguage - The ISO 639-1 language code for the server's user interface language.

setUiCountry

```
public void setUiCountry(java.lang.String uiCountry)
```

Sets the user interface country code. The SDK pre-fills this to the value of uiCountry in the VNCTServerEventConfiguration.

Parameters:

uiCountry - The ISO 3166-1 country code for the server's user interface language.

setKnobKeySupport

```
public void setKnobKeySupport(int knobKeySupport)
```

Sets the knob key support to the provided bitmask. The SDK pre-fills this to KNOB_KEY_SUPPORT_SHIFT_X_0 | KNOB_KEY_SUPPORT_SHIFT_Y_0, masked by the knobKeySupport value in the VNCTServerEventConfiguration. Support for these features of a single 2-D knob is equivalent to support for cursor keys.

Parameters:

knobKeySupport - The server's support for knob key input.

setDeviceKeySupport

```
public void setDeviceKeySupport(int deviceKeySupport)
```

Sets the device key support to the provided bitmask. The SDK pre-fills this to DEVICE_KEY_SUPPORT_ALL, masked by the deviceKeySupport value in the VNCTServerEventConfiguration.

Parameters:

deviceKeySupport - The server's support for MirrorLink device key input.

setMultimediaKeySupport

```
public void setMultimediaKeySupport(int multimediaKeySupport)
```

Sets the multimedia key support to the provided bitmask. The SDK pre-fills this to 0.

Parameters:

multimediaKeySupport - The server's support for multimedia key input.

setMiscKeySupport

```
public void setMiscKeySupport(int miscKeySupport)
```

Sets the miscellaneous key support to the provided bitmask.

In MirrorLink 1.2 sessions and below, the SDK pre-fills this to MISC_KEY_SUPPORT_ITU, masked by the miscKeySupport value in the VNCTServerEventConfiguration.

Some features have been deprecated in MirrorLink 1.3. Viewer applications should not attempt to use these features if they support MirrorLink 1.3 or above. See [EventConfiguration](#) for further information.

Parameters:

`miscKeySupport` - The server's support for miscellaneous MirrorLink key input.

setPointerSupport

```
public void setPointerSupport(int pointerSupport)
```

Sets the pointer / touchscreen support to the provided bitmap. The SDK pre-fills this to `POINTER_SUPPORT_POINTER_EVENTS | POINTER_SUPPORT_POINTER_BUTTON_1 | POINTER_SUPPORT_POINTER_BUTTON_2 | POINTER_SUPPORT_POINTER_BUTTON_3 | POINTER_SUPPORT_POINTER_BUTTON_4 | POINTER_SUPPORT_POINTER_BUTTON_5 | POINTER_SUPPORT_POINTER_BUTTON_6 | POINTER_SUPPORT_POINTER_BUTTON_7 | POINTER_SUPPORT_POINTER_BUTTON_8`, masked by the `pointerSupport` value in the `VNCServerEventConfiguration`.

Parameters:

`pointerSupport` - The server's support for pointer / touchscreen input.

com.realvnc.mirrorlink

Class VNCContextInformation

java.lang.Object
com.realvnc.mirrorlink.VNCContextInformation

```
public class VNCContextInformation
extends java.lang.Object
```

Class holding a decoded ContextInformation rectangle that has been received from the server.

Field Summary

Fields	
Modifier and Type	Field and Description
static int	APPLICATION_CATEGORY_BROWSER General browser category.
static int	APPLICATION_CATEGORY_BROWSER_APPLICATION_STORE Category representing an application store.
static int	APPLICATION_CATEGORY_IMMERSIVE_HOME_SCREEN Category representing an immersive home screen.
static int	APPLICATION_CATEGORY_INFORMATION General information category.
static int	APPLICATION_CATEGORY_INFORMATION_CLOCK Category representing a clock application.
static int	APPLICATION_CATEGORY_INFORMATION_NEWS Category representing a news information application.
static int	APPLICATION_CATEGORY_INFORMATION_SPORTS Category representing a sports information application.
static int	APPLICATION_CATEGORY_INFORMATION_STOCKS Category representing a stocks information application.
static int	APPLICATION_CATEGORY_INFORMATION_TRAVEL Category representing a travel information application.
static int	APPLICATION_CATEGORY_INFORMATION_WEATHER Category representing a weather information application.
static int	APPLICATION_CATEGORY_MASK Used to mask out the category from the subcategory for testing.
static int	APPLICATION_CATEGORY_MEDIA General media applications category.
static int	APPLICATION_CATEGORY_MEDIA_GAMING Category representing a game or gaming related application.
static int	APPLICATION_CATEGORY_MEDIA_IMAGE Category representing an image application.
static int	APPLICATION_CATEGORY_MEDIA_MUSIC Category representing a music application.
static int	APPLICATION_CATEGORY_MEDIA_VIDEO Category representing a video.
static int	APPLICATION_CATEGORY_MESSAGING General messaging applications category.
static int	APPLICATION_CATEGORY_MESSAGING_EMAIL Category representing an email.
static int	APPLICATION_CATEGORY_MESSAGING_MMS Category representing an MMS.

static int	APPLICATION_CATEGORY_MESSAGING_SMS Category representing an SMS.
static int	APPLICATION_CATEGORY_NAVIGATION General navigation category.
static int	APPLICATION_CATEGORY_NO_UI General UI-less applications category.
static int	APPLICATION_CATEGORY_NO_UI_CLIENT Category representing a client.
static int	APPLICATION_CATEGORY_NO_UI_CONVERSATIONAL_AUDIO Category representing conversational audio.
static int	APPLICATION_CATEGORY_NO_UI_SERVER Category representing a server.
static int	APPLICATION_CATEGORY_NO_UI_VOICE_COMMAND_ENGINE Category representing a voice command engine.
static int	APPLICATION_CATEGORY_PHONE General phone call application category.
static int	APPLICATION_CATEGORY_PHONE_CALL_LOG Category representing a call log.
static int	APPLICATION_CATEGORY_PHONE_CONTACT_LIST Category representing a contact list.
static int	APPLICATION_CATEGORY_PHONE_IMMERSIVE_CALL Category representing an immersive phone call.
static int	APPLICATION_CATEGORY_PIM General Personal Information Management category.
static int	APPLICATION_CATEGORY_PIM_CALENDAR Category representing a calendar application.
static int	APPLICATION_CATEGORY_PIM_NOTES Category representing a notes application.
static int	APPLICATION_CATEGORY_PRODUCTIVITY General productivity category.
static int	APPLICATION_CATEGORY_PRODUCTIVITY_DOCUMENT_EDITOR Category representing a document editor.
static int	APPLICATION_CATEGORY_PRODUCTIVITY_DOCUMENT_VIEWER Category representing a document viewer.
static int	APPLICATION_CATEGORY_SOCIAL_NETWORKING General social networking category.
static int	APPLICATION_CATEGORY_SWITCH_TO_CLIENT_NATIVE_UI Category used within the context information to tell the client is should switch to the native UI, or disconnect the VNC session.
static int	APPLICATION_CATEGORY_SYSTEM General system category.
static int	APPLICATION_CATEGORY_SYSTEM_INPUT_BLUETOOTH_PIN Category representing a Bluetooth PIN code input.
static int	APPLICATION_CATEGORY_SYSTEM_INPUT_OTHER_PASSWORD Category representing a password input.
static int	APPLICATION_CATEGORY_SYSTEM_INPUT_UNLOCK_PIN Category representing a PIN input for device unlock.
static int	APPLICATION_CATEGORY_SYSTEM_VOICE_COMMAND_CONFIRMATION Category representing a voice command confirmation.
static int	APPLICATION_CATEGORY_TESTING_AND_CERTIFICATION General testing and certification category.
static int	APPLICATION_CATEGORY_UI General UI framework category.
static int	APPLICATION_CATEGORY_UI_APPLICATION_LISTING Category representing an application listing.

static int	APPLICATION_CATEGORY_UI_HOME_SCREEN Category representing a home screen / start-up screen.
static int	APPLICATION_CATEGORY_UI_MENU Category representing a menu.
static int	APPLICATION_CATEGORY_UI_NOTIFICATION Category representing a notification.
static int	APPLICATION_CATEGORY_UI_SETTINGS Category representing a settings application.
static int	APPLICATION_CATEGORY_UNKNOWN The server has no information about the application category.
static int	TRUST_LEVEL_APPLICATION_CERTIFICATE The provided data is under sole control of the VNC and UPnP server.
static int	TRUST_LEVEL_REGISTERED_APPLICATION The provided data is under sole control of the VNC and UPnP server.
static int	TRUST_LEVEL_SELF_REGISTERED_APPLICATION The provided data is under the control of the application.
static int	TRUST_LEVEL_UNKNOWN The server has no trust in the reported information.
static int	TRUST_LEVEL_USER_CONFIGURATION The provided data is under the control of the user.
static int	VISUAL_CONTENT_CATEGORY_CAR_MODE Content category representing car mode.
static int	VISUAL_CONTENT_CATEGORY_GRAPHICS_3D Content category representing 3D graphics.
static int	VISUAL_CONTENT_CATEGORY_GRAPHICS_VECTOR Content category representing vector graphics.
static int	VISUAL_CONTENT_CATEGORY_IMAGE Content category representing images.
static int	VISUAL_CONTENT_CATEGORY_MISC Content category representing miscellaneous content.
static int	VISUAL_CONTENT_CATEGORY_TEXT Content category representing text.
static int	VISUAL_CONTENT_CATEGORY_UI Content category representing user interface (e.g.
static int	VISUAL_CONTENT_CATEGORY_UNKNOWN Content category representing unknown content.
static int	VISUAL_CONTENT_CATEGORY_VIDEO Content category representing video.

Constructor Summary

Constructors

Constructor and Description

[VNContextInformation](#)(int applicationUniqueId, int applicationCategoryTrustLevel, int contentCategoryTrustLevel, int applicationCategory, int contentCategory, int contentRulesFollowed)

Constructs a new context information object.

Method Summary

Methods

Modifier and Type	Method and Description
int	<code>getApplicationCategory()</code> The category and sub-category into which the application falls.
int	<code>getApplicationCategoryTrustLevel()</code> The server's level of trust that the information in the applicationCategory field is correct.
int	<code>getApplicationUniqueId()</code> Return the application unique ID.
int	<code>getContentCategory()</code> The category of the content that the application is presenting.
int	<code>getContentCategoryTrustLevel()</code> The server's level of trust that the information in the contentCategory field is correct.
int	<code>getContentRulesFollowed()</code> Deprecated. <i>Since MirrorLink 1.3. Must be ignored by MirrorLink 1.3 clients.</i>
java.lang.String	<code>toString()</code> Return a textual representation of this object.

Methods inherited from class java.lang.Object

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

Field Detail

TRUST_LEVEL_UNKNOWN

```
public static final int TRUST_LEVEL_UNKNOWN
```

The server has no trust in the reported information. This is a possible value returned by `getApplicationCategoryTrustLevel()` and `getContentCategoryTrustLevel()`.

See Also:

[Constant Field Values](#)

TRUST_LEVEL_USER_CONFIGURATION

```
public static final int TRUST_LEVEL_USER_CONFIGURATION
```

The provided data is under the control of the user. This is a possible value returned by `getApplicationCategoryTrustLevel()` and `getContentCategoryTrustLevel()`.

See Also:

[Constant Field Values](#)

TRUST_LEVEL_SELF_REGISTERED_APPLICATION

```
public static final int TRUST_LEVEL_SELF_REGISTERED_APPLICATION
```

The provided data is under the control of the application. This is a possible value returned by `getApplicationCategoryTrustLevel()` and `getContentCategoryTrustLevel()`.

See Also:

[Constant Field Values](#)

TRUST_LEVEL_REGISTERED_APPLICATION

```
public static final int TRUST_LEVEL_REGISTERED_APPLICATION
```

The provided data is under sole control of the VNC and UPnP server. The application is known to them and has been uniquely identified. This is a possible value returned by `getApplicationCategoryTrustLevel()` and `getContentCategoryTrustLevel()`.

See Also:

[Constant Field Values](#)

TRUST_LEVEL_APPLICATION_CERTIFICATE

```
public static final int TRUST_LEVEL_APPLICATION_CERTIFICATE
```

The provided data is under sole control of the VNC and UPnP server. The data is derived from a valid application certificate. This is a possible value returned by `getApplicationCategoryTrustLevel()` and `getContentCategoryTrustLevel()`.

See Also:

[Constant Field Values](#)

APPLICATION_CATEGORY_MASK

```
public static final int APPLICATION_CATEGORY_MASK
```

Used to mask out the category from the subcategory for testing. This can be used to split the value returned by `getApplicationCategory()`.

See Also:

[Constant Field Values](#)

APPLICATION_CATEGORY_UNKNOWN

```
public static final int APPLICATION_CATEGORY_UNKNOWN
```

The server has no information about the application category. This can be returned by `getApplicationCategory()`.

See Also:

[Constant Field Values](#)

APPLICATION_CATEGORY_UI

```
public static final int APPLICATION_CATEGORY_UI
```

General UI framework category. This can be returned by `getApplicationCategory()`.

See Also:

[Constant Field Values](#)

APPLICATION_CATEGORY_UI_HOME_SCREEN

```
public static final int APPLICATION_CATEGORY_UI_HOME_SCREEN
```

Category representing a home screen / start-up screen. This can be returned by `getApplicationCategory()`.

See Also:

[Constant Field Values](#)

APPLICATION_CATEGORY_UI_MENU

```
public static final int APPLICATION_CATEGORY_UI_MENU
```

Category representing a menu. This can be returned by `getApplicationCategory()`.

See Also:

[Constant Field Values](#)

APPLICATION_CATEGORY_UI_NOTIFICATION

```
public static final int APPLICATION_CATEGORY_UI_NOTIFICATION
```

Category representing a notification. This can be returned by `getApplicationCategory()`.

See Also:

[Constant Field Values](#)

APPLICATION_CATEGORY_UI_APPLICATION_LISTING

```
public static final int APPLICATION_CATEGORY_UI_APPLICATION_LISTING
```

Category representing an application listing. This can be returned by `getApplicationCategory()`.

See Also:

[Constant Field Values](#)

APPLICATION_CATEGORY_UI_SETTINGS

```
public static final int APPLICATION_CATEGORY_UI_SETTINGS
```

Category representing a settings application. This can be returned by `getApplicationCategory()`.

See Also:

[Constant Field Values](#)

APPLICATION_CATEGORY_IMMERSIVE_HOME_SCREEN

```
public static final int APPLICATION_CATEGORY_IMMERSIVE_HOME_SCREEN
```

Category representing an immersive home screen. This can be returned by `getApplicationCategory()`.

See Also:

[Constant Field Values](#)

APPLICATION_CATEGORY_PHONE

```
public static final int APPLICATION_CATEGORY_PHONE
```

General phone call application category. This can be returned by `getApplicationCategory()`.

See Also:

[Constant Field Values](#)

APPLICATION_CATEGORY_PHONE_CONTACT_LIST

```
public static final int APPLICATION_CATEGORY_PHONE_CONTACT_LIST
```

Category representing a contact list. This can be returned by `getApplicationCategory()`.

See Also:

[Constant Field Values](#)

APPLICATION_CATEGORY_PHONE_CALL_LOG

```
public static final int APPLICATION_CATEGORY_PHONE_CALL_LOG
```

Category representing a call log. This can be returned by `getApplicationCategory()`.

See Also:

[Constant Field Values](#)

APPLICATION_CATEGORY_PHONE_IMMERSIVE_CALL

```
public static final int APPLICATION_CATEGORY_PHONE_IMMERSIVE_CALL
```

Category representing an immersive phone call. This can be returned by `getApplicationCategory()`.

See Also:

[Constant Field Values](#)

APPLICATION_CATEGORY_MEDIA

```
public static final int APPLICATION_CATEGORY_MEDIA
```

General media applications category. This can be returned by `getApplicationCategory()`.

See Also:

[Constant Field Values](#)

APPLICATION_CATEGORY_MEDIA_MUSIC

```
public static final int APPLICATION_CATEGORY_MEDIA_MUSIC
```

Category representing a music application.

See Also:

[Constant Field Values](#)

APPLICATION_CATEGORY_MEDIA_VIDEO

```
public static final int APPLICATION_CATEGORY_MEDIA_VIDEO
```

Category representing a video. This can be returned by `getApplicationCategory()`.

See Also:

[Constant Field Values](#)

APPLICATION_CATEGORY_MEDIA_GAMING

```
public static final int APPLICATION_CATEGORY_MEDIA_GAMING
```

Category representing a game or gaming related application. This can be returned by `getApplicationCategory()`.

See Also:

[Constant Field Values](#)

APPLICATION_CATEGORY_MEDIA_IMAGE

```
public static final int APPLICATION_CATEGORY_MEDIA_IMAGE
```

Category representing an image application. This can be returned by `getApplicationCategory()`.

See Also:

[Constant Field Values](#)

APPLICATION_CATEGORY_MESSAGING

```
public static final int APPLICATION_CATEGORY_MESSAGING
```

General messaging applications category. This can be returned by `getApplicationCategory()`.

See Also:

[Constant Field Values](#)

APPLICATION_CATEGORY_MESSAGING_SMS

```
public static final int APPLICATION_CATEGORY_MESSAGING_SMS
```

Category representing an SMS. This can be returned by `getApplicationCategory()`.

See Also:

[Constant Field Values](#)

APPLICATION_CATEGORY_MESSAGING_MMS

```
public static final int APPLICATION_CATEGORY_MESSAGING_MMS
```

Category representing an MMS. This can be returned by `getApplicationCategory()`.

See Also:

[Constant Field Values](#)

APPLICATION_CATEGORY_MESSAGING_EMAIL

```
public static final int APPLICATION_CATEGORY_MESSAGING_EMAIL
```

Category representing an email. This can be returned by `getApplicationCategory()`.

See Also:

[Constant Field Values](#)

APPLICATION_CATEGORY_NAVIGATION

```
public static final int APPLICATION_CATEGORY_NAVIGATION
```

General navigation category. This can be returned by `getApplicationCategory()`.

See Also:

[Constant Field Values](#)

APPLICATION_CATEGORY_BROWSER

```
public static final int APPLICATION_CATEGORY_BROWSER
```

General browser category. This can be returned by `getApplicationCategory()`.

See Also:

[Constant Field Values](#)

APPLICATION_CATEGORY_BROWSER_APPLICATION_STORE

```
public static final int APPLICATION_CATEGORY_BROWSER_APPLICATION_STORE
```

Category representing an application store. This can be returned by `getApplicationCategory()`.

See Also:

[Constant Field Values](#)

APPLICATION_CATEGORY_PRODUCTIVITY

```
public static final int APPLICATION_CATEGORY_PRODUCTIVITY
```

General productivity category. This can be returned by `getApplicationCategory()`.

See Also:

[Constant Field Values](#)

APPLICATION_CATEGORY_PRODUCTIVITY_DOCUMENT_VIEWER

```
public static final int APPLICATION_CATEGORY_PRODUCTIVITY_DOCUMENT_VIEWER
```

Category representing a document viewer. This can be returned by `getApplicationCategory()`.

See Also:

[Constant Field Values](#)

APPLICATION_CATEGORY_PRODUCTIVITY_DOCUMENT_EDITOR

```
public static final int APPLICATION_CATEGORY_PRODUCTIVITY_DOCUMENT_EDITOR
```

Category representing a document editor. This can be returned by `getApplicationCategory()`.

See Also:

[Constant Field Values](#)

APPLICATION_CATEGORY_INFORMATION

```
public static final int APPLICATION_CATEGORY_INFORMATION
```

General information category. This can be returned by `getApplicationCategory()`.

See Also:

[Constant Field Values](#)

APPLICATION_CATEGORY_INFORMATION_NEWS

```
public static final int APPLICATION_CATEGORY_INFORMATION_NEWS
```

Category representing a news information application. This can be returned by `getApplicationCategory()`.

See Also:

[Constant Field Values](#)

APPLICATION_CATEGORY_INFORMATION_WEATHER

```
public static final int APPLICATION_CATEGORY_INFORMATION_WEATHER
```

Category representing a weather information application. This can be returned by `getApplicationCategory()`.

See Also:

[Constant Field Values](#)

APPLICATION_CATEGORY_INFORMATION_STOCKS

```
public static final int APPLICATION_CATEGORY_INFORMATION_STOCKS
```

Category representing a stocks information application. This can be returned by `getApplicationCategory()`.

See Also:

[Constant Field Values](#)

APPLICATION_CATEGORY_INFORMATION_TRAVEL

```
public static final int APPLICATION_CATEGORY_INFORMATION_TRAVEL
```

Category representing a travel information application. This can be returned by `getApplicationCategory()`.

See Also:

[Constant Field Values](#)

APPLICATION_CATEGORY_INFORMATION_SPORTS

```
public static final int APPLICATION_CATEGORY_INFORMATION_SPORTS
```

Category representing a sports information application. This can be returned by `getApplicationCategory()`.

See Also:

[Constant Field Values](#)

APPLICATION_CATEGORY_INFORMATION_CLOCK

```
public static final int APPLICATION_CATEGORY_INFORMATION_CLOCK
```

Category representing a clock application. This can be returned by `getApplicationCategory()`.

See Also:

[Constant Field Values](#)

APPLICATION_CATEGORY_SOCIAL_NETWORKING

```
public static final int APPLICATION_CATEGORY_SOCIAL_NETWORKING
```

General social networking category. This can be returned by `getApplicationCategory()`.

See Also:

[Constant Field Values](#)

APPLICATION_CATEGORY_PIM

```
public static final int APPLICATION_CATEGORY_PIM
```

General Personal Information Management category. This can be returned by `getApplicationCategory()`.

See Also:

[Constant Field Values](#)

APPLICATION_CATEGORY_PIM_CALENDAR

```
public static final int APPLICATION_CATEGORY_PIM_CALENDAR
```

Category representing a calendar application. This can be returned by `getApplicationCategory()`.

See Also:

[Constant Field Values](#)

APPLICATION_CATEGORY_PIM_NOTES

```
public static final int APPLICATION_CATEGORY_PIM_NOTES
```

Category representing a notes application. This can be returned by `getApplicationCategory()`.

See Also:

[Constant Field Values](#)

APPLICATION_CATEGORY_NO_UI

```
public static final int APPLICATION_CATEGORY_NO_UI
```

General UI-less applications category. This can be returned by `getApplicationCategory()`.

See Also:

[Constant Field Values](#)

APPLICATION_CATEGORY_NO_UI_SERVER

```
public static final int APPLICATION_CATEGORY_NO_UI_SERVER
```

Category representing a server. This can be returned by `getApplicationCategory()`.

See Also:

[Constant Field Values](#)

APPLICATION_CATEGORY_NO_UI_CLIENT

```
public static final int APPLICATION_CATEGORY_NO_UI_CLIENT
```

Category representing a client. This can be returned by `getApplicationCategory()`.

See Also:

[Constant Field Values](#)

APPLICATION_CATEGORY_NO_UI_VOICE_COMMAND_ENGINE

```
public static final int APPLICATION_CATEGORY_NO_UI_VOICE_COMMAND_ENGINE
```

Category representing a voice command engine. This can be returned by `getApplicationCategory()`.

See Also:

[Constant Field Values](#)

APPLICATION_CATEGORY_NO_UI_CONVERSATIONAL_AUDIO

```
public static final int APPLICATION_CATEGORY_NO_UI_CONVERSATIONAL_AUDIO
```

Category representing conversational audio. This is not returned by `getApplicationCategory()` since the MirrorLink specification mandates that this category is only to be used within RTP header extensions.

See Also:

[Constant Field Values](#)

APPLICATION_CATEGORY_SWITCH_TO_CLIENT_NATIVE_UI

```
public static final int APPLICATION_CATEGORY_SWITCH_TO_CLIENT_NATIVE_UI
```

Category used within the context information to tell the client is should switch to the native UI, or disconnect the VNC session. This can be returned by `getApplicationCategory()`.

See Also:

[Constant Field Values](#)

APPLICATION_CATEGORY_TESTING_AND_CERTIFICATION

```
public static final int APPLICATION_CATEGORY_TESTING_AND_CERTIFICATION
```

General testing and certification category. This can be returned by `getApplicationCategory()`.

See Also:

[Constant Field Values](#)

APPLICATION_CATEGORY_SYSTEM

```
public static final int APPLICATION_CATEGORY_SYSTEM
```

General system category. This can be returned by `getApplicationCategory()`.

See Also:

[Constant Field Values](#)

APPLICATION_CATEGORY_SYSTEM_INPUT_UNLOCK_PIN

```
public static final int APPLICATION_CATEGORY_SYSTEM_INPUT_UNLOCK_PIN
```

Category representing a PIN input for device unlock. This can be returned by `getApplicationCategory()`.

See Also:

[Constant Field Values](#)

APPLICATION_CATEGORY_SYSTEM_INPUT_BLUETOOTH_PIN

```
public static final int APPLICATION_CATEGORY_SYSTEM_INPUT_BLUETOOTH_PIN
```

Category representing a Bluetooth PIN code input. This can be returned by `getApplicationCategory()`.

See Also:

[Constant Field Values](#)

APPLICATION_CATEGORY_SYSTEM_INPUT_OTHER_PASSWORD

```
public static final int APPLICATION_CATEGORY_SYSTEM_INPUT_OTHER_PASSWORD
```

Category representing a password input. This can be returned by `getApplicationCategory()`.

See Also:

[Constant Field Values](#)

APPLICATION_CATEGORY_SYSTEM_VOICE_COMMAND_CONFIRMATION

```
public static final int APPLICATION_CATEGORY_SYSTEM_VOICE_COMMAND_CONFIRMATION
```

Category representing a voice command confirmation. This can be returned by `getApplicationCategory()`.

See Also:

[Constant Field Values](#)

VISUAL_CONTENT_CATEGORY_UNKNOWN

```
public static final int VISUAL_CONTENT_CATEGORY_UNKNOWN
```

Content category representing unknown content. This can be returned by `getContentTypeCategory()`.

See Also:

[Constant Field Values](#)

VISUAL_CONTENT_CATEGORY_TEXT

```
public static final int VISUAL_CONTENT_CATEGORY_TEXT
```

Content category representing text. This can be returned by `getContentCategory()`.

See Also:

[Constant Field Values](#)

VISUAL_CONTENT_CATEGORY_VIDEO

```
public static final int VISUAL_CONTENT_CATEGORY_VIDEO
```

Content category representing video. This can be returned by `getContentCategory()`.

See Also:

[Constant Field Values](#)

VISUAL_CONTENT_CATEGORY_IMAGE

```
public static final int VISUAL_CONTENT_CATEGORY_IMAGE
```

Content category representing images. This can be returned by `getContentCategory()`.

See Also:

[Constant Field Values](#)

VISUAL_CONTENT_CATEGORY_GRAPHICS_VECTOR

```
public static final int VISUAL_CONTENT_CATEGORY_GRAPHICS_VECTOR
```

Content category representing vector graphics. This can be returned by `getContentCategory()`.

See Also:

[Constant Field Values](#)

VISUAL_CONTENT_CATEGORY_GRAPHICS_3D

```
public static final int VISUAL_CONTENT_CATEGORY_GRAPHICS_3D
```

Content category representing 3D graphics. This can be returned by `getContentCategory()`.

See Also:

[Constant Field Values](#)

VISUAL_CONTENT_CATEGORY_UI

```
public static final int VISUAL_CONTENT_CATEGORY_UI
```

Content category representing user interface (e.g. application menu). This can be returned by `getContentCategory()`.

See Also:

[Constant Field Values](#)

VISUAL_CONTENT_CATEGORY_CAR_MODE

```
public static final int VISUAL_CONTENT_CATEGORY_CAR_MODE
```

Content category representing car mode. This can be returned by `getContentCategory()`.

See Also:

[Constant Field Values](#)

VISUAL_CONTENT_CATEGORY_MISC

```
public static final int VISUAL_CONTENT_CATEGORY_MISC
```

Content category representing miscellaneous content. This can be returned by `getContentCategory()`.

See Also:

[Constant Field Values](#)

Constructor Detail**VNCCContextInformation**

```
public VNCCContextInformation(int applicationUniqueId,  
                             int applicationCategoryTrustLevel,  
                             int contentCategoryTrustLevel,  
                             int applicationCategory,  
                             int contentCategory,  
                             int contentRulesFollowed)
```

Constructs a new context information object.

Parameters:

`applicationUniqueId` - The unique identifier for this application.

`applicationCategoryTrustLevel` - The level of trust for the application category.

`contentCategoryTrustLevel` - The level of trust for the visual content category.

`applicationCategory` - The application category.

`contentCategory` - The visual content category.

`contentRulesFollowed` - The content rules followed by this content. Deprecated as of MirrorLink 1.3; must be set to 0x00000000

Method Detail**getApplicationUniqueId**

```
public int getApplicationUniqueId()
```

Return the application unique ID.

Returns:

The unique ID of the application that has drawn to the relevant part of the framebuffer.

getApplicationCategoryTrustLevel

```
public int getApplicationCategoryTrustLevel()
```

The server's level of trust that the information in the applicationCategory field is correct.

Returns:

One of the TRUST_LEVEL_* constants.

getContentCategoryTrustLevel

```
public int getContentCategoryTrustLevel()
```

The server's level of trust that the information in the contentCategory field is correct.

Returns:

One of the TRUST_LEVEL_* constants.

getApplicationCategory

```
public int getApplicationCategory()
```

The category and sub-category into which the application falls.

Returns:

One of the APPLICATION_CATEGORY_* constants.

getContentCategory

```
public int getContentCategory()
```

The category of the content that the application is presenting.

Returns:

One of the VISUAL_CONTENT_CATEGORY_* constants.

getContentRulesFollowed

```
public int getContentRulesFollowed()
```

Deprecated. Since MirrorLink 1.3. Must be ignored by MirrorLink 1.3 clients.

Retrieve the content rules followed by the application.

Returns:

A bit-field of the content rules, negotiated in the MirrorLink UPnP stream, with which the application has complied.

toString

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

Return a textual representation of this object.

Overrides:

toString in class java.lang.Object

Returns:

A textual representation of this object.

com.realvnc.mirrorlink

Class VNCDeviceStatus

java.lang.Object

com.realvnc.mirrorlink.VNCDeviceStatus

```
public class VNCDeviceStatus
extends java.lang.Object
```

Class holding a decoded DeviceStatus MirrorLink extension message that has been received from, or will be sent to, the server.

Field Summary

Fields

Modifier and Type	Field and Description
static int	FEATURE_DEVICE_LOCK_DISABLED Device lock is or should be disabled.
static int	FEATURE_DEVICE_LOCK_ENABLED Device lock is or should be enabled.
static int	FEATURE_DEVICE_LOCK_IGNORED This message does not specify the status of the device lock.
static int	FEATURE_DEVICE_LOCK_MASK Used to mask out the device lock status for testing.
static int	FEATURE_DRIVER_DISTRACTION_AVOIDANCE_DISABLED Driver distraction avoidance is or should be disabled.
static int	FEATURE_DRIVER_DISTRACTION_AVOIDANCE_ENABLED Driver distraction avoidance is or should be enabled.
static int	FEATURE_DRIVER_DISTRACTION_AVOIDANCE_IGNORED This message does not specify the status of driver distraction avoidance.
static int	FEATURE_DRIVER_DISTRACTION_AVOIDANCE_MASK Used to mask out the driver distraction avoidance status for testing.
static int	FEATURE_FRAMEBUFFER_ROTATION_0_DEGREES The server framebuffer is not or should not be rotated.
static int	FEATURE_FRAMEBUFFER_ROTATION_180_DEGREES The server framebuffer is or should be rotated 180 degrees.
static int	FEATURE_FRAMEBUFFER_ROTATION_270_DEGREES The server framebuffer is or should be rotated 270 degrees clockwise (or, equivalently, 90 degrees counter-clockwise).
static int	FEATURE_FRAMEBUFFER_ROTATION_90_DEGREES The server framebuffer is or should be rotated 90 degrees clockwise.
static int	FEATURE_FRAMEBUFFER_ROTATION_IGNORED This message does not specify the framebuffer rotation.
static int	FEATURE_FRAMEBUFFER_ROTATION_MASK Used to mask out the framebuffer rotation for testing.
static int	FEATURE_KEY_LOCK_DISABLED Key lock is or should be disabled.
static int	FEATURE_KEY_LOCK_ENABLED Key lock is or should be enabled.
static int	FEATURE_KEY_LOCK_IGNORED This message does not specify the status of the key lock.
static int	FEATURE_KEY_LOCK_MASK Used to mask out the key lock status for testing.
static int	FEATURE_MICROPHONE_INPUT_DISABLED

	Microphone input is or should be disabled.
static int	FEATURE_MICROPHONE_INPUT_ENABLED Microphone input is or should be enabled.
static int	FEATURE_MICROPHONE_INPUT_IGNORED This message does not specify the status of microphone input.
static int	FEATURE_MICROPHONE_INPUT_MASK Used to mask out the microphone input status for testing.
static int	FEATURE_NIGHT_MODE_DISABLED Night mode is or should be disabled.
static int	FEATURE_NIGHT_MODE_ENABLED Night mode is or should be enabled.
static int	FEATURE_NIGHT_MODE_IGNORED This message does not specify the status of night mode.
static int	FEATURE_NIGHT_MODE_MASK Used to mask out the night mode status for testing.
static int	FEATURE_ORIENTATION_IGNORED This message does not specify the display orientation.
static int	FEATURE_ORIENTATION_LANDSCAPE The display orientation is or should be landscape.
static int	FEATURE_ORIENTATION_MASK Used to mask out the display orientation for testing.
static int	FEATURE_ORIENTATION_PORTRAIT The display orientation is or should be portrait.
static int	FEATURE_SCREENSAVER_DISABLED Screensaver is or should be disabled.
static int	FEATURE_SCREENSAVER_ENABLED Screensaver is or should be enabled.
static int	FEATURE_SCREENSAVER_IGNORED This message does not specify the status of the screensaver.
static int	FEATURE_SCREENSAVER_MASK Used to mask out the screensaver status for testing.
static int	FEATURE_VOICE_INPUT_DISABLED Voice input is or should be disabled.
static int	FEATURE_VOICE_INPUT_ENABLED Voice input is or should be enabled.
static int	FEATURE_VOICE_INPUT_IGNORED This message does not specify the status of voice input.
static int	FEATURE_VOICE_INPUT_MASK Used to mask out the voice input status for testing.
static int	FEATURE_VOICE_INPUT_REROUTING_DISABLED Deprecated. <i>Use FEATURE_MICROPHONE_INPUT_DISABLED instead.</i>
static int	FEATURE_VOICE_INPUT_REROUTING_ENABLED Deprecated. <i>Use FEATURE_MICROPHONE_INPUT_ENABLED instead.</i>
static int	FEATURE_VOICE_INPUT_REROUTING_IGNORED Deprecated. <i>Use FEATURE_MICROPHONE_INPUT_IGNORED instead.</i>
static int	FEATURE_VOICE_INPUT_REROUTING_MASK Deprecated. <i>Use FEATURE_MICROPHONE_INPUT_MASK instead.</i>

Constructor Summary

Constructors**Constructor and Description**

`VNCDeviceStatus(int features)`
Construct a DeviceStatus object.

Method Summary**Methods**

Modifier and Type	Method and Description
int	<code>getFeatures()</code> Retrieves a bitmask of device status features.
void	<code>setFeatures(int features)</code> Sets a bitmask of device status features.
java.lang.String	<code>toString()</code> Retrives a textual representation of this object.

Methods inherited from class java.lang.Object

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

Field Detail**FEATURE_KEY_LOCK_IGNORED**

```
public static final int FEATURE_KEY_LOCK_IGNORED
```

This message does not specify the status of the key lock.

See Also:

[Constant Field Values](#)

FEATURE_KEY_LOCK_DISABLED

```
public static final int FEATURE_KEY_LOCK_DISABLED
```

Key lock is or should be disabled.

This feature has been deprecated in MirrorLink 1.3. If your application supports MirrorLink 1.3 or above, then you should only use `FEATURE_KEY_LOCK_IGNORED`.

See Also:

[Constant Field Values](#)

FEATURE_KEY_LOCK_ENABLED

```
public static final int FEATURE_KEY_LOCK_ENABLED
```

Key lock is or should be enabled.

This feature has been deprecated in MirrorLink 1.3. If your application supports MirrorLink 1.3 or above, then you should only use `FEATURE_KEY_LOCK_IGNORED`.

See Also:

[Constant Field Values](#)**FEATURE_KEY_LOCK_MASK**

```
public static final int FEATURE_KEY_LOCK_MASK
```

Used to mask out the key lock status for testing.

See Also:[Constant Field Values](#)**FEATURE_DEVICE_LOCK_IGNORED**

```
public static final int FEATURE_DEVICE_LOCK_IGNORED
```

This message does not specify the status of the device lock.

See Also:[Constant Field Values](#)**FEATURE_DEVICE_LOCK_DISABLED**

```
public static final int FEATURE_DEVICE_LOCK_DISABLED
```

Device lock is or should be disabled. This feature has been deprecated in MirrorLink 1.3 for clients. Viewer applications should always use [FEATURE_DEVICE_LOCK_IGNORED](#) in in device status requests if they support MirrorLink 1.3 or above. Server applications which support MirrorLink 1.3 or above may still report the current device lock status.

See Also:[Constant Field Values](#)**FEATURE_DEVICE_LOCK_ENABLED**

```
public static final int FEATURE_DEVICE_LOCK_ENABLED
```

Device lock is or should be enabled. This feature has been deprecated in MirrorLink 1.3 for clients. Viewer applications should always use [FEATURE_DEVICE_LOCK_IGNORED](#) in in device status requests if they support MirrorLink 1.3 or above. Server applications which support MirrorLink 1.3 or above may still report the current device lock status.

See Also:[Constant Field Values](#)**FEATURE_DEVICE_LOCK_MASK**

```
public static final int FEATURE_DEVICE_LOCK_MASK
```

Used to mask out the device lock status for testing.

See Also:[Constant Field Values](#)**FEATURE_SCREENSAVER_IGNORED**

```
public static final int FEATURE_SCREENSAVER_IGNORED
```

This message does not specify the status of the screensaver.

See Also:

[Constant Field Values](#)

FEATURE_SCREENSAVER_DISABLED

```
public static final int FEATURE_SCREENSAVER_DISABLED
```

Screensaver is or should be disabled.

This feature has been deprecated in MirrorLink 1.3. If your application supports MirrorLink 1.3 or above, then you should only use [FEATURE_SCREENSAVER_IGNORED](#).

See Also:

[Constant Field Values](#)

FEATURE_SCREENSAVER_ENABLED

```
public static final int FEATURE_SCREENSAVER_ENABLED
```

Screensaver is or should be enabled.

This feature has been deprecated in MirrorLink 1.3. If your application supports MirrorLink 1.3 or above, then you should only use [FEATURE_SCREENSAVER_IGNORED](#).

See Also:

[Constant Field Values](#)

FEATURE_SCREENSAVER_MASK

```
public static final int FEATURE_SCREENSAVER_MASK
```

Used to mask out the screensaver status for testing.

See Also:

[Constant Field Values](#)

FEATURE_NIGHT_MODE_IGNORED

```
public static final int FEATURE_NIGHT_MODE_IGNORED
```

This message does not specify the status of night mode.

See Also:

[Constant Field Values](#)

FEATURE_NIGHT_MODE_DISABLED

```
public static final int FEATURE_NIGHT_MODE_DISABLED
```

Night mode is or should be disabled.

See Also:

[Constant Field Values](#)

FEATURE_NIGHT_MODE_ENABLED

```
public static final int FEATURE_NIGHT_MODE_ENABLED
```

Night mode is or should be enabled.

See Also:

[Constant Field Values](#)

FEATURE_NIGHT_MODE_MASK

```
public static final int FEATURE_NIGHT_MODE_MASK
```

Used to mask out the night mode status for testing.

See Also:

[Constant Field Values](#)

FEATURE_VOICE_INPUT_IGNORED

```
public static final int FEATURE_VOICE_INPUT_IGNORED
```

This message does not specify the status of voice input.

See Also:

[Constant Field Values](#)

FEATURE_VOICE_INPUT_DISABLED

```
public static final int FEATURE_VOICE_INPUT_DISABLED
```

Voice input is or should be disabled.

See Also:

[Constant Field Values](#)

FEATURE_VOICE_INPUT_ENABLED

```
public static final int FEATURE_VOICE_INPUT_ENABLED
```

Voice input is or should be enabled.

See Also:

[Constant Field Values](#)

FEATURE_VOICE_INPUT_MASK

```
public static final int FEATURE_VOICE_INPUT_MASK
```

Used to mask out the voice input status for testing.

See Also:

[Constant Field Values](#)

FEATURE_MICROPHONE_INPUT_IGNORED

```
public static final int FEATURE_MICROPHONE_INPUT_IGNORED
```

This message does not specify the status of microphone input.

See Also:

[Constant Field Values](#)

FEATURE_MICROPHONE_INPUT_DISABLED

```
public static final int FEATURE_MICROPHONE_INPUT_DISABLED
```

Microphone input is or should be disabled.

See Also:

[Constant Field Values](#)

FEATURE_MICROPHONE_INPUT_ENABLED

```
public static final int FEATURE_MICROPHONE_INPUT_ENABLED
```

Microphone input is or should be enabled.

See Also:

[Constant Field Values](#)

FEATURE_MICROPHONE_INPUT_MASK

```
public static final int FEATURE_MICROPHONE_INPUT_MASK
```

Used to mask out the microphone input status for testing.

See Also:

[Constant Field Values](#)

FEATURE_VOICE_INPUT_REROUTING_IGNORED

@Deprecated

```
public static final int FEATURE_VOICE_INPUT_REROUTING_IGNORED
```

Deprecated. Use `FEATURE_MICROPHONE_INPUT_IGNORED` instead.

See Also:

[Constant Field Values](#)

FEATURE_VOICE_INPUT_REROUTING_DISABLED

@Deprecated

```
public static final int FEATURE_VOICE_INPUT_REROUTING_DISABLED
```

Deprecated. Use `FEATURE_MICROPHONE_INPUT_DISABLED` instead.

See Also:

[Constant Field Values](#)

FEATURE_VOICE_INPUT_REROUTING_ENABLED

@Deprecated

```
public static final int FEATURE_VOICE_INPUT_REROUTING_ENABLED
```

Deprecated. Use [FEATURE_MICROPHONE_INPUT_ENABLED](#) instead.

See Also:

[Constant Field Values](#)

FEATURE_VOICE_INPUT_REROUTING_MASK

@Deprecated

```
public static final int FEATURE_VOICE_INPUT_REROUTING_MASK
```

Deprecated. Use [FEATURE_MICROPHONE_INPUT_MASK](#) instead.

See Also:

[Constant Field Values](#)

FEATURE_DRIVER_DISTRACTION_AVOIDANCE_IGNORED

```
public static final int FEATURE_DRIVER_DISTRACTION_AVOIDANCE_IGNORED
```

This message does not specify the status of driver distraction avoidance.

See Also:

[Constant Field Values](#)

FEATURE_DRIVER_DISTRACTION_AVOIDANCE_DISABLED

```
public static final int FEATURE_DRIVER_DISTRACTION_AVOIDANCE_DISABLED
```

Driver distraction avoidance is or should be disabled.

See Also:

[Constant Field Values](#)

FEATURE_DRIVER_DISTRACTION_AVOIDANCE_ENABLED

```
public static final int FEATURE_DRIVER_DISTRACTION_AVOIDANCE_ENABLED
```

Driver distraction avoidance is or should be enabled.

See Also:

[Constant Field Values](#)

FEATURE_DRIVER_DISTRACTION_AVOIDANCE_MASK

```
public static final int FEATURE_DRIVER_DISTRACTION_AVOIDANCE_MASK
```

Used to mask out the driver distraction avoidance status for testing.

See Also:

[Constant Field Values](#)

FEATURE_FRAMEBUFFER_ROTATION_IGNORED

```
public static final int FEATURE_FRAMEBUFFER_ROTATION_IGNORED
```

This message does not specify the framebuffer rotation.

See Also:

[Constant Field Values](#)

FEATURE_FRAMEBUFFER_ROTATION_0_DEGREES

```
public static final int FEATURE_FRAMEBUFFER_ROTATION_0_DEGREES
```

The server framebuffer is not or should not be rotated.

See Also:

[Constant Field Values](#)

FEATURE_FRAMEBUFFER_ROTATION_90_DEGREES

```
public static final int FEATURE_FRAMEBUFFER_ROTATION_90_DEGREES
```

The server framebuffer is or should be rotated 90 degrees clockwise.

This feature has been deprecated in MirrorLink 1.3. Server applications should only use [FEATURE_FRAMEBUFFER_ROTATION_0_DEGREES](#) if they they support MirrorLink 1.3 or above. Viewer applications which support MirrorLink 1.3 or above may use either [FEATURE_FRAMEBUFFER_ROTATION_0_DEGREES](#) or [FEATURE_FRAMEBUFFER_ROTATION_IGNORED](#).

See Also:

[Constant Field Values](#)

FEATURE_FRAMEBUFFER_ROTATION_180_DEGREES

```
public static final int FEATURE_FRAMEBUFFER_ROTATION_180_DEGREES
```

The server framebuffer is or should be rotated 180 degrees.

This feature has been deprecated in MirrorLink 1.3. Server applications should only use [FEATURE_FRAMEBUFFER_ROTATION_0_DEGREES](#) if they they support MirrorLink 1.3 or above. Viewer applications which support MirrorLink 1.3 or above may use either [FEATURE_FRAMEBUFFER_ROTATION_0_DEGREES](#) or [FEATURE_FRAMEBUFFER_ROTATION_IGNORED](#).

See Also:

[Constant Field Values](#)

FEATURE_FRAMEBUFFER_ROTATION_270_DEGREES

```
public static final int FEATURE_FRAMEBUFFER_ROTATION_270_DEGREES
```

The server framebuffer is or should be rotated 270 degrees clockwise (or, equivalently, 90 degrees counter-clockwise).

This feature has been deprecated in MirrorLink 1.3. Server applications should only use [FEATURE_FRAMEBUFFER_ROTATION_0_DEGREES](#) if they they support MirrorLink 1.3 or above. Viewer applications which support MirrorLink 1.3 or above may use either [FEATURE_FRAMEBUFFER_ROTATION_0_DEGREES](#) or [FEATURE_FRAMEBUFFER_ROTATION_IGNORED](#).

See Also:

[Constant Field Values](#)

FEATURE_FRAMEBUFFER_ROTATION_MASK

```
public static final int FEATURE_FRAMEBUFFER_ROTATION_MASK
```

Used to mask out the framebuffer rotation for testing.

See Also:

[Constant Field Values](#)

FEATURE_ORIENTATION_IGNORED

```
public static final int FEATURE_ORIENTATION_IGNORED
```

This message does not specify the display orientation.

This feature has been deprecated in MirrorLink 1.3. If your application supports MirrorLink 1.3 or above, then you should only use [FEATURE_ORIENTATION_LANDSCAPE](#).

See Also:

[Constant Field Values](#)

FEATURE_ORIENTATION_LANDSCAPE

```
public static final int FEATURE_ORIENTATION_LANDSCAPE
```

The display orientation is or should be landscape.

See Also:

[Constant Field Values](#)

FEATURE_ORIENTATION_PORTRAIT

```
public static final int FEATURE_ORIENTATION_PORTRAIT
```

The display orientation is or should be portrait.

This feature has been deprecated in MirrorLink 1.3. If your application supports MirrorLink 1.3 or above, then you should only use [FEATURE_ORIENTATION_LANDSCAPE](#).

See Also:

[Constant Field Values](#)

FEATURE_ORIENTATION_MASK

```
public static final int FEATURE_ORIENTATION_MASK
```

Used to mask out the display orientation for testing.

See Also:

[Constant Field Values](#)

Constructor Detail

VNCDeviceStatus

```
public VNCDeviceStatus(int features)
```

Construct a DeviceStatus object.

Some features have been deprecated in MirrorLink 1.3. Your application should not attempt to use deprecated feature status values if it supports MirrorLink 1.3 or above. See the FEATURE_* values for further information.

Parameters:

`features` - A bitfield made up of FEATURE_* values.

Method Detail**getFeatures**

```
public int getFeatures()
```

Retrieves a bitmask of device status features.

Some features have been deprecated in MirrorLink 1.3. Your application should not attempt to use deprecated feature status values if it supports MirrorLink 1.3 or above. See the FEATURE_* values for further information.

Returns:

the features specified by this message, as a bitfield made up of FEATURE_* values.

setFeatures

```
public void setFeatures(int features)
```

Sets a bitmask of device status features.

Some features have been deprecated in MirrorLink 1.3. Your application should not attempt to use deprecated feature status values if it supports MirrorLink 1.3 or above. See the FEATURE_* values for further information.

Parameters:

`features` - The features to specify in this message, as a bitfield made up of FEATURE_* values.

toString

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

Retrives a textual representation of this object.

Overrides:

`toString` in class `java.lang.Object`

Returns:

A textual representation of this object.

com.realvnc.mirrorlink

Class VNCFramebufferBlockingNotification

java.lang.Object
com.realvnc.mirrorlink.VNCFramebufferBlockingNotification

```
public class VNCFramebufferBlockingNotification
extends java.lang.Object
```

Class holding a FramebufferBlockingNotification MirrorLink extension message to be sent to the server.

Field Summary

Fields	
Modifier and Type	Field and Description
static int	REASON_APPLICATION_CATEGORY_NOT_ALLOWED The application category has been disallowed (for example, by the driver distraction policy).
static int	REASON_APPLICATION_NOT_TRUSTED The server's trust in the application category that it reported is not sufficient to satisfy the viewer application.
static int	REASON_APPLICATION_UNIQUE_ID_NOT_ALLOWED The server application has been disallowed based on its unique ID (for example, by the driver distraction policy).
static int	REASON_CONTENT_CATEGORY_NOT_ALLOWED The application's content category has been disallowed (for example, by the driver distraction policy).
static int	REASON_CONTENT_NOT_TRUSTED The server's trust in the content category that it reported is not sufficient to satisfy the viewer application.
static int	REASON_CONTENT_RULES_NOT_FOLLOWED The server application has not followed the content rules that were communicated to the server via UPnP.
static int	REASON_UI_LAYOUT_NOT_SUPPORTED UI layout not supported.
static int	REASON_UI_NOT_IN_FOCUS The viewer application is not in focus.
static int	REASON_UI_NOT_VISIBLE The UI of the viewer application is not visible to the user.

Constructor Summary

Constructors	
Constructor and Description	
VNCFramebufferBlockingNotification(int applicationUniqueId, int reason)	Construct a VNCFramebufferBlockingNotification object.
VNCFramebufferBlockingNotification(int applicationUniqueId, int reason, int uniqueId)	Construct a VNCFramebufferBlockingNotification object.

Method Summary

Methods

Modifier and Type	Method and Description
int	<code>getApplicationUniqueId()</code> Retrieves the unique ID of a blocked application.
int	<code>getReason()</code> Retrieves the reason for application blocking.
int	<code>getUniqueId()</code> Retrieves the unique ID of the notification.
java.lang.String	<code>toString()</code> Retrieves a textual representation of this object.

Methods inherited from class java.lang.Object

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

Field Detail**REASON_CONTENT_CATEGORY_NOT_ALLOWED**

```
public static final int REASON_CONTENT_CATEGORY_NOT_ALLOWED
```

The application's content category has been disallowed (for example, by the driver distraction policy).

This reason has been deprecated in MirrorLink 1.1 and is forbidden to be used in MirrorLink 1.3. Viewer applications should not specify this reason if they support MirrorLink 1.3 or above. Server applications supporting MirrorLink 1.3 or above should reinterpret this as [`REASON_APPLICATION_NOT_TRUSTED`](#).

See Also:

[Constant Field Values](#)

REASON_APPLICATION_CATEGORY_NOT_ALLOWED

```
public static final int REASON_APPLICATION_CATEGORY_NOT_ALLOWED
```

The application category has been disallowed (for example, by the driver distraction policy).

This reason has been deprecated and is forbidden to be used in MirrorLink 1.3. Viewer applications should not specify this reason if they support MirrorLink 1.3 or above. Server applications supporting MirrorLink 1.3 or above should reinterpret this as [`REASON_APPLICATION_NOT_TRUSTED`](#).

See Also:

[Constant Field Values](#)

REASON_CONTENT_NOT_TRUSTED

```
public static final int REASON_CONTENT_NOT_TRUSTED
```

The server's trust in the content category that it reported is not sufficient to satisfy the viewer application.

This reason has been deprecated in MirrorLink 1.1 and is forbidden to be used in MirrorLink 1.3. Viewer applications should not specify this reason if they support MirrorLink 1.3 or above. Server applications supporting MirrorLink 1.3 or above should reinterpret this as [`REASON_APPLICATION_NOT_TRUSTED`](#).

See Also:

[Constant Field Values](#)

REASON_APPLICATION_NOT_TRUSTED

```
public static final int REASON_APPLICATION_NOT_TRUSTED
```

The server's trust in the application category that it reported is not sufficient to satisfy the viewer application.

See Also:

[Constant Field Values](#)

REASON_CONTENT_RULES_NOT_FOLLOWED

```
public static final int REASON_CONTENT_RULES_NOT_FOLLOWED
```

The server application has not followed the content rules that were communicated to the server via UPnP.

This reason has been deprecated in MirrorLink 1.1 and is forbidden to be used in MirrorLink 1.3. Viewer applications should not specify this reason if they support MirrorLink 1.3 or above. Server applications supporting MirrorLink 1.3 or above should reinterpret this as [REASON_APPLICATION_NOT_TRUSTED](#).

See Also:

[Constant Field Values](#)

REASON_APPLICATION_UNIQUE_ID_NOT_ALLOWED

```
public static final int REASON_APPLICATION_UNIQUE_ID_NOT_ALLOWED
```

The server application has been disallowed based on its unique ID (for example, by the driver distraction policy).

This reason has been deprecated and is forbidden to be used in MirrorLink 1.3. Viewer applications should not specify this reason if they support MirrorLink 1.3 or above. Server applications supporting MirrorLink 1.3 or above should reinterpret this as [REASON_APPLICATION_NOT_TRUSTED](#).

See Also:

[Constant Field Values](#)

REASON_UI_NOT_IN_FOCUS

```
public static final int REASON_UI_NOT_IN_FOCUS
```

The viewer application is not in focus.

This reason has been deprecated in MirrorLink 1.1 and is forbidden to be used in MirrorLink 1.3. Viewer applications should not specify this reason if they support MirrorLink 1.3 or above.

See Also:

[Constant Field Values](#)

REASON_UI_NOT_VISIBLE

```
public static final int REASON_UI_NOT_VISIBLE
```

The UI of the viewer application is not visible to the user.

See Also:

[Constant Field Values](#)

REASON_UI_LAYOUT_NOT_SUPPORTED

```
public static final int REASON_UI_LAYOUT_NOT_SUPPORTED
```

UI layout not supported.

The VNC Client may send a FramebufferBlockingNotification message with this bit set. In that case, the VNC Server must change the layout back to the original orientation, if necessary terminating the current application, and send another DesktopSize message.

See Also:

[Constant Field Values](#)

Constructor Detail**VNCFramebufferBlockingNotification**

```
public VNCFramebufferBlockingNotification(int applicationUniqueId,  
                                         int reason)
```

Construct a VNCFramebufferBlockingNotification object.

This constructor is deprecated for the Server SDK. Use `VNCFramebufferBlockingNotification(int,int,int)` instead.

Some framebuffer blocking reasons have been deprecated in MirrorLink 1.3. Viewer applications should not specify these reasons if they support MirrorLink 1.3 or above. See the REASON_* constants for further information.

Parameters:

`applicationUniqueId` - The unique ID of the application that has been blocked.

The value should be taken from the most recent `VNCContextInformation` passed to the viewer application for the relevant part of the framebuffer.

`reason` - The reason for the viewer application's decision to block the relevant part of the framebuffer, as a bitfield made up of REASON_* constants. Depending on the reason, the server may choose to dismiss the application in question.

VNCFramebufferBlockingNotification

```
public VNCFramebufferBlockingNotification(int applicationUniqueId,  
                                         int reason,  
                                         int uniqueId)
```

Construct a VNCFramebufferBlockingNotification object.

Some framebuffer blocking reasons have been deprecated in MirrorLink 1.3. Viewer applications should not specify these reasons if they support MirrorLink 1.3 or above. See the REASON_* constants for further information.

Parameters:

`applicationUniqueId` - The unique ID of the application that has been blocked.

The value should be taken from the most recent `VNCContextInformation` passed to the viewer application for the relevant part of the framebuffer.

`reason` - The reason for the viewer application's decision to block the relevant part of the framebuffer, as a bitfield made up of REASON_* constants. Depending on the reason, the server may choose to dismiss the application in question.

`uniqueId` - The unique ID of the notification. This is only needed by the Server SDK, to keep track of which notifications have been handled. The IDs will be provided in the notifications sent by the SDK callbacks. Applications should not create their own IDs.

Method Detail

getApplicationUniqueId

```
public int getApplicationUniqueId()
```

Retrieves the unique ID of a blocked application.

Returns:

The unique ID of the application that has been blocked.

getReason

```
public int getReason()
```

Retrieves the reason for application blocking.

Returns:

The reason for the viewer application's decision to block the relevant part of the framebuffer, as a bitfield made up of REASON_* constants.

getUniqueId

```
public int getUniqueId()
```

Retrieves the unique ID of the notification.

Returns:

The unique ID of the notification.

toString

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

Retrieves a textual representation of this object.

Overrides:

toString in class java.lang.Object

Returns:

A textual representation of this object.

com.realvnc.mirrorlink

Class VNCMirrorLinkKeys

java.lang.Object
com.realvnc.mirrorlink.VNCMirrorLinkKeys

```
public abstract class VNCMirrorLinkKeys
extends java.lang.Object
```

VNCMirrorLinkKeys

This file defines additional X key symbols for use with MirrorLink servers.

Note that devices are only likely to respond to particular key events if their operating system understands the idea behind the key event in question. For example, sending XK_DEVICE_BACKWARD to a device whose operating system does not expect the device to have a physical 'back' key is unlikely to have an effect.

Refer to Appendix A, 'Event Mapping', in the MirrorLink specification, for further information.

MirrorLink servers support up to four 2D knobs as input devices. The key symbol constants for knob input are generated by the XK_KNOB_2D_n family of constants, where n should be in the range 0 to 3 inclusive.

In the case of knob rotation, there is one constant for clockwise rotation and one constant for counter-clockwise rotation about each axis. Clockwise rotation is denoted by the axis name in lower-case (e.g. XK_KNOB_2D_ROTATE_x) and counter-clockwise rotation is denoted by the axis name in upper-case (e.g. XK_KNOB_2D_ROTATE_X).

Field Summary

Fields	
Modifier and Type	Field and Description
static int	XK_DEVICE_APPLICATION
static int	XK_DEVICE_BACKWARD
static int	XK_DEVICE_CLEAR
static int	XK_DEVICE_DELETE
static int	XK_DEVICE_FORWARD
static int	XK_DEVICE_HOME
static int	XK_DEVICE_MENU
static int	XK_DEVICE_OK
static int	XK_DEVICE_PHONE_CALL
static int	XK_DEVICE_PHONE_END
static int	XK_DEVICE_SEARCH
static int	XK_DEVICE_SOFT_LEFT
static int	XK_DEVICE_SOFT_MIDDLE
static int	XK_DEVICE_SOFT_RIGHT
static int	XK_DEVICE_ZOOM_IN
static int	XK_DEVICE_ZOOM_OUT
static int	XK_FUNCTION_KEY_0
static int	XK_FUNCTION_KEY_1
static int	XK_FUNCTION_KEY_10
static int	XK_FUNCTION_KEY_11
static int	XK_FUNCTION_KEY_12
static int	XK_FUNCTION_KEY_2
static int	XK_FUNCTION_KEY_3

static int	XK_FUNCTION_KEY_4
static int	XK_FUNCTION_KEY_5
static int	XK_FUNCTION_KEY_6
static int	XK_FUNCTION_KEY_7
static int	XK_FUNCTION_KEY_8
static int	XK_FUNCTION_KEY_9
static int	XK_ITU_KEY_0
static int	XK_ITU_KEY_1
static int	XK_ITU_KEY_2
static int	XK_ITU_KEY_3
static int	XK_ITU_KEY_4
static int	XK_ITU_KEY_5
static int	XK_ITU_KEY_6
static int	XK_ITU_KEY_7
static int	XK_ITU_KEY_8
static int	XK_ITU_KEY_9
static int	XK_ITU_KEY_ASTERIX
static int	XK_ITU_KEY_POUND
static int	XK_KNOB_2D_ROTATE_x_0
static int	XK_KNOB_2D_ROTATE_X_0
static int	XK_KNOB_2D_ROTATE_x_1
static int	XK_KNOB_2D_ROTATE_X_1
static int	XK_KNOB_2D_ROTATE_x_2
static int	XK_KNOB_2D_ROTATE_X_2
static int	XK_KNOB_2D_ROTATE_x_3
static int	XK_KNOB_2D_ROTATE_X_3
static int	XK_KNOB_2D_ROTATE_y_0
static int	XK_KNOB_2D_ROTATE_Y_0
static int	XK_KNOB_2D_ROTATE_y_1
static int	XK_KNOB_2D_ROTATE_Y_1
static int	XK_KNOB_2D_ROTATE_y_2
static int	XK_KNOB_2D_ROTATE_Y_2
static int	XK_KNOB_2D_ROTATE_y_3
static int	XK_KNOB_2D_ROTATE_Y_3
static int	XK_KNOB_2D_ROTATE_z_0
static int	XK_KNOB_2D_ROTATE_Z_0
static int	XK_KNOB_2D_ROTATE_z_1
static int	XK_KNOB_2D_ROTATE_Z_1
static int	XK_KNOB_2D_ROTATE_z_2
static int	XK_KNOB_2D_ROTATE_Z_2
static int	XK_KNOB_2D_ROTATE_z_3
static int	XK_KNOB_2D_ROTATE_Z_3
static int	XK_KNOB_2D_SHIFT_DOWN_0
static int	XK_KNOB_2D_SHIFT_DOWN_1
static int	XK_KNOB_2D_SHIFT_DOWN_2
static int	XK_KNOB_2D_SHIFT_DOWN_3
static int	XK_KNOB_2D_SHIFT_DOWN_LEFT_0
static int	XK_KNOB_2D_SHIFT_DOWN_LEFT_1

static int	XK_KNOB_2D_SHIFT_DOWN_LEFT_2
static int	XK_KNOB_2D_SHIFT_DOWN_LEFT_3
static int	XK_KNOB_2D_SHIFT_DOWN_RIGHT_0
static int	XK_KNOB_2D_SHIFT_DOWN_RIGHT_1
static int	XK_KNOB_2D_SHIFT_DOWN_RIGHT_2
static int	XK_KNOB_2D_SHIFT_DOWN_RIGHT_3
static int	XK_KNOB_2D_SHIFT_LEFT_0
static int	XK_KNOB_2D_SHIFT_LEFT_1
static int	XK_KNOB_2D_SHIFT_LEFT_2
static int	XK_KNOB_2D_SHIFT_LEFT_3
static int	XK_KNOB_2D_SHIFT_PULL_0
static int	XK_KNOB_2D_SHIFT_PULL_1
static int	XK_KNOB_2D_SHIFT_PULL_2
static int	XK_KNOB_2D_SHIFT_PULL_3
static int	XK_KNOB_2D_SHIFT_PUSH_0
static int	XK_KNOB_2D_SHIFT_PUSH_1
static int	XK_KNOB_2D_SHIFT_PUSH_2
static int	XK_KNOB_2D_SHIFT_PUSH_3
static int	XK_KNOB_2D_SHIFT_RIGHT_0
static int	XK_KNOB_2D_SHIFT_RIGHT_1
static int	XK_KNOB_2D_SHIFT_RIGHT_2
static int	XK_KNOB_2D_SHIFT_RIGHT_3
static int	XK_KNOB_2D_SHIFT_UP_0
static int	XK_KNOB_2D_SHIFT_UP_1
static int	XK_KNOB_2D_SHIFT_UP_2
static int	XK_KNOB_2D_SHIFT_UP_3
static int	XK_KNOB_2D_SHIFT_UP_LEFT_0
static int	XK_KNOB_2D_SHIFT_UP_LEFT_1
static int	XK_KNOB_2D_SHIFT_UP_LEFT_2
static int	XK_KNOB_2D_SHIFT_UP_LEFT_3
static int	XK_KNOB_2D_SHIFT_UP_RIGHT_0
static int	XK_KNOB_2D_SHIFT_UP_RIGHT_1
static int	XK_KNOB_2D_SHIFT_UP_RIGHT_2
static int	XK_KNOB_2D_SHIFT_UP_RIGHT_3
static int	XK_MULTIMEDIA_FORWARD
static int	XK_MULTIMEDIA_MUTE
static int	XK_MULTIMEDIA_NEXT
static int	XK_MULTIMEDIA_PAUSE
static int	XK_MULTIMEDIA_PHOTO
static int	XK_MULTIMEDIA_PLAY
static int	XK_MULTIMEDIA_PREVIOUS
static int	XK_MULTIMEDIA_REWIND
static int	XK_MULTIMEDIA_STOP
static int	XK_MULTIMEDIA_UNMUTE

Constructor Summary

Constructors

Constructor and Description

VNCMirrorLinkKeys ()

Method Summary

Methods inherited from class java.lang.Object

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

Field Detail

XK_ITU_KEY_0

public static final int XK_ITU_KEY_0

See Also:

Constant Field Values

XK_ITU_KEY_1

public static final int XK_ITU_KEY_1

See Also:

Constant Field Values

XK_ITU_KEY_2

public static final int XK_ITU_KEY_2

See Also:

Constant Field Values

XK_ITU_KEY_3

public static final int XK_ITU_KEY_3

See Also:

Constant Field Values

XK_ITU_KEY_4

public static final int XK_ITU_KEY_4

See Also:

Constant Field Values

05/07/2018 20:15

XK_ITU_KEY_5

```
public static final int XK_ITU_KEY_5
```

See Also:[Constant Field Values](#)**XK_ITU_KEY_6**

```
public static final int XK_ITU_KEY_6
```

See Also:[Constant Field Values](#)**XK_ITU_KEY_7**

```
public static final int XK_ITU_KEY_7
```

See Also:[Constant Field Values](#)**XK_ITU_KEY_8**

```
public static final int XK_ITU_KEY_8
```

See Also:[Constant Field Values](#)**XK_ITU_KEY_9**

```
public static final int XK_ITU_KEY_9
```

See Also:[Constant Field Values](#)**XK_ITU_KEY_ASTERIX**

```
public static final int XK_ITU_KEY_ASTERIX
```

See Also:[Constant Field Values](#)**XK_ITU_KEY_POUND**

```
public static final int XK_ITU_KEY_POUND
```

See Also:[Constant Field Values](#)**XK_DEVICE_PHONE_CALL**

```
public static final int XK_DEVICE_PHONE_CALL
```

See Also:[Constant Field Values](#)**XK_DEVICE_PHONE_END**

```
public static final int XK_DEVICE_PHONE_END
```

See Also:[Constant Field Values](#)**XK_DEVICE_SOFT_LEFT**

```
public static final int XK_DEVICE_SOFT_LEFT
```

See Also:[Constant Field Values](#)**XK_DEVICE_SOFT_MIDDLE**

```
public static final int XK_DEVICE_SOFT_MIDDLE
```

See Also:[Constant Field Values](#)**XK_DEVICE_SOFT_RIGHT**

```
public static final int XK_DEVICE_SOFT_RIGHT
```

See Also:[Constant Field Values](#)**XK_DEVICE_APPLICATION**

```
public static final int XK_DEVICE_APPLICATION
```

See Also:[Constant Field Values](#)**XK_DEVICE_OK**

```
public static final int XK_DEVICE_OK
```

See Also:[Constant Field Values](#)**XK_DEVICE_DELETE**

```
public static final int XK_DEVICE_DELETE
```


See Also:[Constant Field Values](#)**XK_DEVICE_ZOOM_IN**

```
public static final int XK_DEVICE_ZOOM_IN
```

See Also:[Constant Field Values](#)**XK_DEVICE_ZOOM_OUT**

```
public static final int XK_DEVICE_ZOOM_OUT
```

See Also:[Constant Field Values](#)**XK_DEVICE_CLEAR**

```
public static final int XK_DEVICE_CLEAR
```

See Also:[Constant Field Values](#)**XK_DEVICE_FORWARD**

```
public static final int XK_DEVICE_FORWARD
```

See Also:[Constant Field Values](#)**XK_DEVICE_BACKWARD**

```
public static final int XK_DEVICE_BACKWARD
```

See Also:[Constant Field Values](#)**XK_DEVICE_HOME**

```
public static final int XK_DEVICE_HOME
```

See Also:[Constant Field Values](#)**XK_DEVICE_SEARCH**

```
public static final int XK_DEVICE_SEARCH
```

See Also:

[Constant Field Values](#)

XK_DEVICE_MENU

```
public static final int XK_DEVICE_MENU
```

See Also:

[Constant Field Values](#)

XK_MULTIMEDIA_PLAY

```
public static final int XK_MULTIMEDIA_PLAY
```

See Also:

[Constant Field Values](#)

XK_MULTIMEDIA_PAUSE

```
public static final int XK_MULTIMEDIA_PAUSE
```

See Also:

[Constant Field Values](#)

XK_MULTIMEDIA_STOP

```
public static final int XK_MULTIMEDIA_STOP
```

See Also:

[Constant Field Values](#)

XK_MULTIMEDIA_FORWARD

```
public static final int XK_MULTIMEDIA_FORWARD
```

See Also:

[Constant Field Values](#)

XK_MULTIMEDIA_REWIND

```
public static final int XK_MULTIMEDIA_REWIND
```

See Also:

[Constant Field Values](#)

XK_MULTIMEDIA_NEXT

```
public static final int XK_MULTIMEDIA_NEXT
```

See Also:

[Constant Field Values](#)

XK_MULTIMEDIA_PREVIOUS

```
public static final int XK_MULTIMEDIA_PREVIOUS
```

See Also:[Constant Field Values](#)**XK_MULTIMEDIA_MUTE**

```
public static final int XK_MULTIMEDIA_MUTE
```

See Also:[Constant Field Values](#)**XK_MULTIMEDIA_UNMUTE**

```
public static final int XK_MULTIMEDIA_UNMUTE
```

See Also:[Constant Field Values](#)**XK_MULTIMEDIA_PHOTO**

```
public static final int XK_MULTIMEDIA_PHOTO
```

See Also:[Constant Field Values](#)**XK_FUNCTION_KEY_0**

```
public static final int XK_FUNCTION_KEY_0
```

See Also:[Constant Field Values](#)**XK_FUNCTION_KEY_1**

```
public static final int XK_FUNCTION_KEY_1
```

See Also:[Constant Field Values](#)**XK_FUNCTION_KEY_2**

```
public static final int XK_FUNCTION_KEY_2
```

See Also:[Constant Field Values](#)**XK_FUNCTION_KEY_3**

```
public static final int XK_FUNCTION_KEY_3
```

See Also:[Constant Field Values](#)**XK_FUNCTION_KEY_4**

```
public static final int XK_FUNCTION_KEY_4
```

See Also:[Constant Field Values](#)**XK_FUNCTION_KEY_5**

```
public static final int XK_FUNCTION_KEY_5
```

See Also:[Constant Field Values](#)**XK_FUNCTION_KEY_6**

```
public static final int XK_FUNCTION_KEY_6
```

See Also:[Constant Field Values](#)**XK_FUNCTION_KEY_7**

```
public static final int XK_FUNCTION_KEY_7
```

See Also:[Constant Field Values](#)**XK_FUNCTION_KEY_8**

```
public static final int XK_FUNCTION_KEY_8
```

See Also:[Constant Field Values](#)**XK_FUNCTION_KEY_9**

```
public static final int XK_FUNCTION_KEY_9
```

See Also:[Constant Field Values](#)**XK_FUNCTION_KEY_10**

```
public static final int XK_FUNCTION_KEY_10
```

See Also:[Constant Field Values](#)**XK_FUNCTION_KEY_11**

```
public static final int XK_FUNCTION_KEY_11
```

See Also:[Constant Field Values](#)**XK_FUNCTION_KEY_12**

```
public static final int XK_FUNCTION_KEY_12
```

See Also:[Constant Field Values](#)**XK_KNOB_2D_SHIFT_RIGHT_0**

```
public static final int XK_KNOB_2D_SHIFT_RIGHT_0
```

See Also:[Constant Field Values](#)**XK_KNOB_2D_SHIFT_RIGHT_1**

```
public static final int XK_KNOB_2D_SHIFT_RIGHT_1
```

See Also:[Constant Field Values](#)**XK_KNOB_2D_SHIFT_RIGHT_2**

```
public static final int XK_KNOB_2D_SHIFT_RIGHT_2
```

See Also:[Constant Field Values](#)**XK_KNOB_2D_SHIFT_RIGHT_3**

```
public static final int XK_KNOB_2D_SHIFT_RIGHT_3
```

See Also:[Constant Field Values](#)**XK_KNOB_2D_SHIFT_LEFT_0**

```
public static final int XK_KNOB_2D_SHIFT_LEFT_0
```

See Also:

Constant Field Values

XK_KNOB_2D_SHIFT_LEFT_1

```
public static final int XK_KNOB_2D_SHIFT_LEFT_1
```

See Also:[Constant Field Values](#)**XK_KNOB_2D_SHIFT_LEFT_2**

```
public static final int XK_KNOB_2D_SHIFT_LEFT_2
```

See Also:[Constant Field Values](#)**XK_KNOB_2D_SHIFT_LEFT_3**

```
public static final int XK_KNOB_2D_SHIFT_LEFT_3
```

See Also:[Constant Field Values](#)**XK_KNOB_2D_SHIFT_UP_0**

```
public static final int XK_KNOB_2D_SHIFT_UP_0
```

See Also:[Constant Field Values](#)**XK_KNOB_2D_SHIFT_UP_1**

```
public static final int XK_KNOB_2D_SHIFT_UP_1
```

See Also:[Constant Field Values](#)**XK_KNOB_2D_SHIFT_UP_2**

```
public static final int XK_KNOB_2D_SHIFT_UP_2
```

See Also:[Constant Field Values](#)**XK_KNOB_2D_SHIFT_UP_3**

```
public static final int XK_KNOB_2D_SHIFT_UP_3
```

See Also:[Constant Field Values](#)

XK_KNOB_2D_SHIFT_UP_RIGHT_0

```
public static final int XK_KNOB_2D_SHIFT_UP_RIGHT_0
```

See Also:[Constant Field Values](#)**XK_KNOB_2D_SHIFT_UP_RIGHT_1**

```
public static final int XK_KNOB_2D_SHIFT_UP_RIGHT_1
```

See Also:[Constant Field Values](#)**XK_KNOB_2D_SHIFT_UP_RIGHT_2**

```
public static final int XK_KNOB_2D_SHIFT_UP_RIGHT_2
```

See Also:[Constant Field Values](#)**XK_KNOB_2D_SHIFT_UP_RIGHT_3**

```
public static final int XK_KNOB_2D_SHIFT_UP_RIGHT_3
```

See Also:[Constant Field Values](#)**XK_KNOB_2D_SHIFT_UP_LEFT_0**

```
public static final int XK_KNOB_2D_SHIFT_UP_LEFT_0
```

See Also:[Constant Field Values](#)**XK_KNOB_2D_SHIFT_UP_LEFT_1**

```
public static final int XK_KNOB_2D_SHIFT_UP_LEFT_1
```

See Also:[Constant Field Values](#)**XK_KNOB_2D_SHIFT_UP_LEFT_2**

```
public static final int XK_KNOB_2D_SHIFT_UP_LEFT_2
```

See Also:[Constant Field Values](#)**XK_KNOB_2D_SHIFT_UP_LEFT_3**

```
public static final int XK_KNOB_2D_SHIFT_UP_LEFT_3
```

See Also:[Constant Field Values](#)**XK_KNOB_2D_SHIFT_DOWN_0**

```
public static final int XK_KNOB_2D_SHIFT_DOWN_0
```

See Also:[Constant Field Values](#)**XK_KNOB_2D_SHIFT_DOWN_1**

```
public static final int XK_KNOB_2D_SHIFT_DOWN_1
```

See Also:[Constant Field Values](#)**XK_KNOB_2D_SHIFT_DOWN_2**

```
public static final int XK_KNOB_2D_SHIFT_DOWN_2
```

See Also:[Constant Field Values](#)**XK_KNOB_2D_SHIFT_DOWN_3**

```
public static final int XK_KNOB_2D_SHIFT_DOWN_3
```

See Also:[Constant Field Values](#)**XK_KNOB_2D_SHIFT_DOWN_RIGHT_0**

```
public static final int XK_KNOB_2D_SHIFT_DOWN_RIGHT_0
```

See Also:[Constant Field Values](#)**XK_KNOB_2D_SHIFT_DOWN_RIGHT_1**

```
public static final int XK_KNOB_2D_SHIFT_DOWN_RIGHT_1
```

See Also:[Constant Field Values](#)**XK_KNOB_2D_SHIFT_DOWN_RIGHT_2**

```
public static final int XK_KNOB_2D_SHIFT_DOWN_RIGHT_2
```


See Also:[Constant Field Values](#)**XK_KNOB_2D_SHIFT_DOWN_RIGHT_3**

```
public static final int XK_KNOB_2D_SHIFT_DOWN_RIGHT_3
```

See Also:[Constant Field Values](#)**XK_KNOB_2D_SHIFT_DOWN_LEFT_0**

```
public static final int XK_KNOB_2D_SHIFT_DOWN_LEFT_0
```

See Also:[Constant Field Values](#)**XK_KNOB_2D_SHIFT_DOWN_LEFT_1**

```
public static final int XK_KNOB_2D_SHIFT_DOWN_LEFT_1
```

See Also:[Constant Field Values](#)**XK_KNOB_2D_SHIFT_DOWN_LEFT_2**

```
public static final int XK_KNOB_2D_SHIFT_DOWN_LEFT_2
```

See Also:[Constant Field Values](#)**XK_KNOB_2D_SHIFT_DOWN_LEFT_3**

```
public static final int XK_KNOB_2D_SHIFT_DOWN_LEFT_3
```

See Also:[Constant Field Values](#)**XK_KNOB_2D_SHIFT_PUSH_0**

```
public static final int XK_KNOB_2D_SHIFT_PUSH_0
```

See Also:[Constant Field Values](#)**XK_KNOB_2D_SHIFT_PUSH_1**

```
public static final int XK_KNOB_2D_SHIFT_PUSH_1
```

See Also:

[Constant Field Values](#)

XK_KNOB_2D_SHIFT_PUSH_2

```
public static final int XK_KNOB_2D_SHIFT_PUSH_2
```

See Also:

[Constant Field Values](#)

XK_KNOB_2D_SHIFT_PUSH_3

```
public static final int XK_KNOB_2D_SHIFT_PUSH_3
```

See Also:

[Constant Field Values](#)

XK_KNOB_2D_SHIFT_PULL_0

```
public static final int XK_KNOB_2D_SHIFT_PULL_0
```

See Also:

[Constant Field Values](#)

XK_KNOB_2D_SHIFT_PULL_1

```
public static final int XK_KNOB_2D_SHIFT_PULL_1
```

See Also:

[Constant Field Values](#)

XK_KNOB_2D_SHIFT_PULL_2

```
public static final int XK_KNOB_2D_SHIFT_PULL_2
```

See Also:

[Constant Field Values](#)

XK_KNOB_2D_SHIFT_PULL_3

```
public static final int XK_KNOB_2D_SHIFT_PULL_3
```

See Also:

[Constant Field Values](#)

XK_KNOB_2D_ROTATE_x_0

```
public static final int XK_KNOB_2D_ROTATE_x_0
```

See Also:

[Constant Field Values](#)

XK_KNOB_2D_ROTATE_x_1

```
public static final int XK_KNOB_2D_ROTATE_x_1
```

See Also:[Constant Field Values](#)**XK_KNOB_2D_ROTATE_x_2**

```
public static final int XK_KNOB_2D_ROTATE_x_2
```

See Also:[Constant Field Values](#)**XK_KNOB_2D_ROTATE_x_3**

```
public static final int XK_KNOB_2D_ROTATE_x_3
```

See Also:[Constant Field Values](#)**XK_KNOB_2D_ROTATE_X_0**

```
public static final int XK_KNOB_2D_ROTATE_X_0
```

See Also:[Constant Field Values](#)**XK_KNOB_2D_ROTATE_X_1**

```
public static final int XK_KNOB_2D_ROTATE_X_1
```

See Also:[Constant Field Values](#)**XK_KNOB_2D_ROTATE_X_2**

```
public static final int XK_KNOB_2D_ROTATE_X_2
```

See Also:[Constant Field Values](#)**XK_KNOB_2D_ROTATE_X_3**

```
public static final int XK_KNOB_2D_ROTATE_X_3
```

See Also:[Constant Field Values](#)**XK_KNOB_2D_ROTATE_y_0**

```
public static final int XK_KNOB_2D_ROTATE_y_0
```

See Also:

[Constant Field Values](#)

XK_KNOB_2D_ROTATE_y_1

```
public static final int XK_KNOB_2D_ROTATE_y_1
```

See Also:

[Constant Field Values](#)

XK_KNOB_2D_ROTATE_y_2

```
public static final int XK_KNOB_2D_ROTATE_y_2
```

See Also:

[Constant Field Values](#)

XK_KNOB_2D_ROTATE_y_3

```
public static final int XK_KNOB_2D_ROTATE_y_3
```

See Also:

[Constant Field Values](#)

XK_KNOB_2D_ROTATE_Y_0

```
public static final int XK_KNOB_2D_ROTATE_Y_0
```

See Also:

[Constant Field Values](#)

XK_KNOB_2D_ROTATE_Y_1

```
public static final int XK_KNOB_2D_ROTATE_Y_1
```

See Also:

[Constant Field Values](#)

XK_KNOB_2D_ROTATE_Y_2

```
public static final int XK_KNOB_2D_ROTATE_Y_2
```

See Also:

[Constant Field Values](#)

XK_KNOB_2D_ROTATE_Y_3

```
public static final int XK_KNOB_2D_ROTATE_Y_3
```

See Also:

[Constant Field Values](#)

XK_KNOB_2D_ROTATE_z_0

```
public static final int XK_KNOB_2D_ROTATE_z_0
```

See Also:

[Constant Field Values](#)

XK_KNOB_2D_ROTATE_z_1

```
public static final int XK_KNOB_2D_ROTATE_z_1
```

See Also:

[Constant Field Values](#)

XK_KNOB_2D_ROTATE_z_2

```
public static final int XK_KNOB_2D_ROTATE_z_2
```

See Also:

[Constant Field Values](#)

XK_KNOB_2D_ROTATE_z_3

```
public static final int XK_KNOB_2D_ROTATE_z_3
```

See Also:

[Constant Field Values](#)

XK_KNOB_2D_ROTATE_Z_0

```
public static final int XK_KNOB_2D_ROTATE_Z_0
```

See Also:

[Constant Field Values](#)

XK_KNOB_2D_ROTATE_Z_1

```
public static final int XK_KNOB_2D_ROTATE_Z_1
```

See Also:

[Constant Field Values](#)

XK_KNOB_2D_ROTATE_Z_2

```
public static final int XK_KNOB_2D_ROTATE_Z_2
```

See Also:

Constant Field Values

XK_KNOB_2D_ROTATE_Z_3

```
public static final int XK_KNOB_2D_ROTATE_Z_3
```

See Also:

[Constant Field Values](#)

Constructor Detail

VNCMirrorLinkKeys

```
public VNCMirrorLinkKeys()
```

com.realvnc.mirrorlink

Class VNCServerDisplayConfiguration

java.lang.Object

com.realvnc.mirrorlink.DisplayConfiguration

com.realvnc.mirrorlink.VNCServerDisplayConfiguration

```
public class VNCServerDisplayConfiguration
extends DisplayConfiguration
```

Class holding a decoded ServerDisplayConfiguration MirrorLink extension message that has been received from the server.

Field Summary

Fields inherited from class com.realvnc.mirrorlink.DisplayConfiguration

FRAMEBUFFER_CONFIGURATION_DOWNSCALING, FRAMEBUFFER_CONFIGURATION_REPLACE_EMPTY_UPDATES, FRAMEBUFFER_CONFIGURATION_SERVERSIDE_ORIENTATION_SWITCH, FRAMEBUFFER_CONFIGURATION_SERVERSIDE_ROTATION, FRAMEBUFFER_CONFIGURATION_SUPPORTS_FRAMEBUFFER_ALTERNATIVE_TEXT, FRAMEBUFFER_CONFIGURATION_UPSCALING, PIXELFORMAT_SUPPORT_ANY_16, PIXELFORMAT_SUPPORT_ANY_24, PIXELFORMAT_SUPPORT_ANY_32, PIXELFORMAT_SUPPORT_ARGB888_32, PIXELFORMAT_SUPPORT_GRAYSCALE_16, PIXELFORMAT_SUPPORT_GRAYSCALE_8, PIXELFORMAT_SUPPORT_NONE, PIXELFORMAT_SUPPORT_RGB_343_16, PIXELFORMAT_SUPPORT_RGB444_16, PIXELFORMAT_SUPPORT_RGB555_16, PIXELFORMAT_SUPPORT_RGB565_16, PIXELFORMAT_SUPPORT_RGB888_32, RESIZE_FACTOR_1_1, RESIZE_FACTOR_1_10, RESIZE_FACTOR_1_16, RESIZE_FACTOR_1_2, RESIZE_FACTOR_1_3, RESIZE_FACTOR_1_32, RESIZE_FACTOR_1_4, RESIZE_FACTOR_1_5, RESIZE_FACTOR_1_6, RESIZE_FACTOR_1_8, RESIZE_FACTOR_2_3, RESIZE_FACTOR_3_4, RESIZE_FACTOR_NONE

Constructor Summary

Constructors

Constructor and Description
<code>VNCServerDisplayConfiguration(int serverMajorVersion, int serverMinorVersion, int framebufferConfiguration, int relativePixelWidth, int relativePixelHeight, int pixelFormatSupport)</code> Construct a VNCServerDisplayConfiguration object.

Method Summary

Methods

Modifier and Type	Method and Description
int	<code>getFramebufferConfiguration()</code> Retrieves a bitmask of the framebuffer configuration.
int	<code>getPixelFormatSupport()</code> Retrieves a bitmask of the supported pixel formats.
int	<code>getRelativePixelHeight()</code> Retreives the relative pixel height.
int	<code>getRelativePixelWidth()</code> Retrieves the relative pixel width.
int	<code>getServerMajorVersion()</code> Retrieves the major server version.
int	<code>getServerMinorVersion()</code>

Retrieves the minor server version.
<code>java.lang.String toString()</code>
Retrives a textual representation of this object.

Methods inherited from class java.lang.Object

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

Constructor Detail

VNCServerDisplayConfiguration

```
public VNCServerDisplayConfiguration(int serverMajorVersion,
                                     int serverMinorVersion,
                                     int framebufferConfiguration,
                                     int relativePixelWidth,
                                     int relativePixelHeight,
                                     int pixelFormatSupport)
```

Construct a VNCServerDisplayConfiguration object.

Note that several of these fields have been deprecated in MirrorLink 1.3. Acceptable values are as follows:

- `framebufferConfiguration`: 0 or `DisplayConfiguration.FRAMEBUFFER_CONFIGURATION_DOWNSCALING`
- `relativePixelWidth`: 1
- `relativePixelHeight`: 1
- `pixelFormatSupport`: `DisplayConfiguration.PIXELFORMAT_SUPPORT_RGB565_16`, `DisplayConfiguration.PIXELFORMAT_SUPPORT_ARGB888_32`, or both

Parameters:

`serverMajorVersion` - The major server version.

`serverMinorVersion` - The minor server version.

`framebufferConfiguration` - The framebuffer configuration, as a bitmask.

`relativePixelWidth` - The pixel width, relative height.

`relativePixelHeight` - The pixel height, relative to width

`pixelFormatSupport` - The pixel formats supported, as a bitmask.

Method Detail

getServerMajorVersion

```
public int getServerMajorVersion()
```

Retrieves the major server version.

Returns:

The major version number of the MirrorLink specification followed by the server.

getServerMinorVersion

```
public int getServerMinorVersion()
```

Retrieves the minor server version.

Returns:

The minor version number of the MirrorLink specification followed by the server.

getFramebufferConfiguration

```
public int getFramebufferConfiguration()
```

Retrieves a bitmask of the framebuffer configuration.

Some capabilities have been deprecated in MirrorLink 1.3. Your application should not attempt to use these capabilities if it supports MirrorLink 1.3 or above. See [DisplayConfiguration](#) for further information.

Returns:

A bitfield made up of FRAMEBUFFER_CONFIGURATION_* constants describing the server's capabilities with respect to its framebuffer.

getRelativePixelWidth

```
public int getRelativePixelWidth()
```

Retrieves the relative pixel width.

This field was deprecated in MirrorLink 1.1. Your application should set (or assume) a relative pixel width value of 1 if it supports MirrorLink 1.1 or above.

Returns:

The width of each pixel in the server display relative to its height. This has no effect on the Viewer SDK, but you may wish to take note of it when displaying the framebuffer to the user.

getRelativePixelHeight

```
public int getRelativePixelHeight()
```

Retrieves the relative pixel height.

This field was deprecated in MirrorLink 1.1. Your application should set (or assume) a relative pixel height value of 1 if it supports MirrorLink 1.1 or above.

Returns:

The height of each pixel in the server display relative to its width. This has no effect on the Viewer SDK, but you may wish to take note of it when displaying the framebuffer to the user.

getPixelFormatSupport

```
public int getPixelFormatSupport()
```

Retrieves a bitmask of the supported pixel formats.

Some pixel formats have been deprecated in MirrorLink 1.3. Your application should not attempt to use these pixel formats if it supports MirrorLink 1.3 or above. See [DisplayConfiguration](#) for further information.

Returns:

A bitfield made up of PIXELFORMAT_SUPPORT_* constants that indicates the pixel formats supported by the server.

toString

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

Retrives a textual representation of this object.

Overrides:

`toString` in class `java.lang.Object`

Returns:

A textual representation of this object.

com.realvnc.mirrorlink

Class VNCServerEventConfiguration

java.lang.Object
com.realvnc.mirrorlink.EventConfiguration
com.realvnc.mirrorlink.VNCServerEventConfiguration

Direct Known Subclasses:

VNCClientEventConfiguration

```
public class VNCServerEventConfiguration
extends EventConfiguration
```

Class holding a decoded ServerEventConfiguration MirrorLink extension message that has been received from the server.

Field Summary

Fields	
Modifier and Type	Field and Description
protected int	deviceKeySupport Contains a bitmask of device key support.
protected java.lang.String	keyboardCountry Contains the keyboard layout country code.
protected java.lang.String	keyboardLanguage Contains the keyboard layout language code.
protected int	knobKeySupport Contains a bitmask of knob key support.
protected int	miscKeySupport Contains a bitmask of miscellaneous support.
protected int	multimediaKeySupport Contains a bitmask of multimedia key support.
protected int	pointerSupport Contains a bitmask of pointer support.
protected java.lang.String	uiCountry Contains the user interface country code
protected java.lang.String	uiLanguage Contains the user interface language code.

Fields inherited from class com.realvnc.mirrorlink.EventConfiguration
DEVICE_KEY_SUPPORT_ALL, DEVICE_KEY_SUPPORT_APPLICATION, DEVICE_KEY_SUPPORT_BACKWARD, DEVICE_KEY_SUPPORT_CLEAR, DEVICE_KEY_SUPPORT_DELETE, DEVICE_KEY_SUPPORT_FORWARD, DEVICE_KEY_SUPPORT_HOME, DEVICE_KEY_SUPPORT_MENU, DEVICE_KEY_SUPPORT_OK, DEVICE_KEY_SUPPORT_PHONE_CALL, DEVICE_KEY_SUPPORT_PHONE_END, DEVICE_KEY_SUPPORT_SEARCH, DEVICE_KEY_SUPPORT_SOFT_LEFT, DEVICE_KEY_SUPPORT_SOFT_MIDDLE, DEVICE_KEY_SUPPORT_SOFT_RIGHT, DEVICE_KEY_SUPPORT_ZOOM_IN, DEVICE_KEY_SUPPORT_ZOOM_OUT, KNOB_KEY_SUPPORT_PULL_Z_0, KNOB_KEY_SUPPORT_PULL_Z_1, KNOB_KEY_SUPPORT_PULL_Z_2, KNOB_KEY_SUPPORT_PULL_Z_3, KNOB_KEY_SUPPORT_PUSH_Z_0, KNOB_KEY_SUPPORT_PUSH_Z_1, KNOB_KEY_SUPPORT_PUSH_Z_2, KNOB_KEY_SUPPORT_PUSH_Z_3, KNOB_KEY_SUPPORT_ROTATE_X_0, KNOB_KEY_SUPPORT_ROTATE_X_1, KNOB_KEY_SUPPORT_ROTATE_X_2, KNOB_KEY_SUPPORT_ROTATE_X_3, KNOB_KEY_SUPPORT_ROTATE_Y_0, KNOB_KEY_SUPPORT_ROTATE_Y_1, KNOB_KEY_SUPPORT_ROTATE_Y_2, KNOB_KEY_SUPPORT_ROTATE_Y_3, KNOB_KEY_SUPPORT_ROTATE_Z_0, KNOB_KEY_SUPPORT_ROTATE_Z_1, KNOB_KEY_SUPPORT_ROTATE_Z_2, KNOB_KEY_SUPPORT_ROTATE_Z_3, KNOB_KEY_SUPPORT_SHIFT_X_0, KNOB_KEY_SUPPORT_SHIFT_X_1, KNOB_KEY_SUPPORT_SHIFT_X_2, KNOB_KEY_SUPPORT_SHIFT_X_3, KNOB_KEY_SUPPORT_SHIFT_XY_0, KNOB_KEY_SUPPORT_SHIFT_XY_1, KNOB_KEY_SUPPORT_SHIFT_XY_2, KNOB_KEY_SUPPORT_SHIFT_XY_3, KNOB_KEY_SUPPORT_SHIFT_Y_0, KNOB_KEY_SUPPORT_SHIFT_Y_1, KNOB_KEY_SUPPORT_SHIFT_Y_2, KNOB_KEY_SUPPORT_SHIFT_Y_3, MISC_KEY_SUPPORT_EVENT_MAPPING, MISC_KEY_SUPPORT_FUNCTION_KEY_0, MISC_KEY_SUPPORT_FUNCTION_KEY_1, MISC_KEY_SUPPORT_FUNCTION_KEY_2, MISC_KEY_SUPPORT_FUNCTION_KEY_3, MISC_KEY_SUPPORT_FUNCTION_KEY_4, MISC_KEY_SUPPORT_FUNCTION_KEY_5, MISC_KEY_SUPPORT_FUNCTION_KEY_6,

```

MISC_KEY_SUPPORT_FUNCTION_KEY_7, MISC_KEY_SUPPORT_FUNCTION_KEY_MASK,
MISC_KEY_SUPPORT_FUNCTION_KEY_SHIFT, MISC_KEY_SUPPORT_ITU, MISC_KEY_SUPPORT_KEY_EVENT_LISTING,
MISC_KEY_SUPPORT_KEY_MAPPING_MASK, MISC_KEY_SUPPORT_KEY_MAPPING_SHIFT,
MISC_KEY_SUPPORT_VIRTUAL_KEYBOARD_TRIGGER, MULTIMEDIA_KEY_SUPPORT_FORWARD,
MULTIMEDIA_KEY_SUPPORT_MUTE, MULTIMEDIA_KEY_SUPPORT_NEXT, MULTIMEDIA_KEY_SUPPORT_PAUSE,
MULTIMEDIA_KEY_SUPPORT_PHOTO, MULTIMEDIA_KEY_SUPPORT_PLAY, MULTIMEDIA_KEY_SUPPORT_PREVIOUS,
MULTIMEDIA_KEY_SUPPORT_REWIND, MULTIMEDIA_KEY_SUPPORT_STOP, MULTIMEDIA_KEY_SUPPORT_UNMUTE,
POINTER_SUPPORT_POINTER_BUTTON_1, POINTER_SUPPORT_POINTER_BUTTON_2, POINTER_SUPPORT_POINTER_BUTTON_3,
POINTER_SUPPORT_POINTER_BUTTON_4, POINTER_SUPPORT_POINTER_BUTTON_5, POINTER_SUPPORT_POINTER_BUTTON_6,
POINTER_SUPPORT_POINTER_BUTTON_7, POINTER_SUPPORT_POINTER_BUTTON_8,
POINTER_SUPPORT_POINTER_BUTTON_MASK, POINTER_SUPPORT_POINTER_EVENTS, POINTER_SUPPORT_TOUCH_COUNT_1,
POINTER_SUPPORT_TOUCH_COUNT_10, POINTER_SUPPORT_TOUCH_COUNT_2, POINTER_SUPPORT_TOUCH_COUNT_3,
POINTER_SUPPORT_TOUCH_COUNT_4, POINTER_SUPPORT_TOUCH_COUNT_5, POINTER_SUPPORT_TOUCH_COUNT_6,
POINTER_SUPPORT_TOUCH_COUNT_7, POINTER_SUPPORT_TOUCH_COUNT_8, POINTER_SUPPORT_TOUCH_COUNT_9,
POINTER_SUPPORT_TOUCH_COUNT_MASK, POINTER_SUPPORT_TOUCH_COUNT_MASK_SHIFT,
POINTER_SUPPORT_TOUCH_EVENT_PRESSURE_MASK, POINTER_SUPPORT_TOUCH_EVENT_PRESSURE_MASK_SHIFT,
POINTER_SUPPORT_TOUCH_EVENTS

```

Constructor Summary

Constructors

Constructor and Description

VNCServerEventConfiguration(java.lang.String keyboardLanguage, java.lang.String keyboardCountry, java.lang.String uiLanguage, java.lang.String uiCountry, int knobKeySupport, int deviceKeySupport, int multimediaKeySupport, int miscKeySupport, int pointerSupport)

Constructs a new server event configuration object.

Method Summary

Methods

Modifier and Type	Method and Description
int	getDeviceKeySupport() Retrieves the device key support as a bitmask.
java.lang.String	getKeyboardCountry() Retrieves the keyboard layout country code.
java.lang.String	getKeyboardLanguage() Retrieves the keyboard layout language code.
int	getKnobKeySupport() Retrieves the knob key support as a bitmask.
int	getMiscKeySupport() Retrieves the miscellaneous key support as a bitmask.
int	getMultimediaKeySupport() Retrieves the multimedia key support as a bitmask.
int	getNumFunctionKeysSupported() Retrieves the number of function keys supported.
int	getPointerSupport() Retrieves the pointer / touchscreen support as a bitmask.
java.lang.String	getUiCountry() Retrieves the user interface country code.
java.lang.String	getUiLanguage() Retrieves the user interface language code.
boolean	isEventMappingSupported() Check whether event mapping is supported.
boolean	isITUKeySupported() Check whether ITU keypad events are supported.

boolean	<code>isKeyEventListingSupported()</code> Check whether the key event listing is supported.
boolean	<code>isVirtualKeyboardTriggerSupported()</code> Check whether the virtual keyboard trigger is supported.
java.lang.String	<code>toString()</code> Returns a textual representation of this object.

Methods inherited from class java.lang.Object

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

Field Detail

keyboardLanguage

protected java.lang.String keyboardLanguage

Contains the keyboard layout language code.

keyboardCountry

protected java.lang.String keyboardCountry

Contains the keyboard layout country code.

uiLanguage

protected java.lang.String uiLanguage

Contains the user interface language code.

uiCountry

protected java.lang.String uiCountry

Contains the user interface country code

knobKeySupport

protected int knobKeySupport

Contains a bitmask of knob key support.

deviceKeySupport

protected int deviceKeySupport

Contains a bitmask of device key support.

multimediaKeySupport

protected int multimediaKeySupport

Contains a bitmask of multimedia key support.

miscKeySupport

```
protected int miscKeySupport
```

Contains a bitmask of miscellaneous support.

pointerSupport

```
protected int pointerSupport
```

Contains a bitmask of pointer support.

Constructor Detail**VNCServerEventConfiguration**

```
public VNCServerEventConfiguration(java.lang.String keyboardLanguage,  
                                   java.lang.String keyboardCountry,  
                                   java.lang.String uiLanguage,  
                                   java.lang.String uiCountry,  
                                   int knobKeySupport,  
                                   int deviceKeySupport,  
                                   int multimediaKeySupport,  
                                   int miscKeySupport,  
                                   int pointerSupport)
```

Constructs a new server event configuration object.

Note that several of these fields have been deprecated in MirrorLink 1.3 and above. In particular, `miscKeySupport` should not indicate support for `EventConfiguration.MISC_KEY_SUPPORT_ITU`, `EventConfiguration.MISC_KEY_SUPPORT_VIRTUAL_KEYBOARD_TRIGGER` or `EventConfiguration.MISC_KEY_SUPPORT_KEY_EVENT_LISTING`.

Additionally, certain fields are required to be present in MirrorLink 1.3 and above. In particular, `deviceKeySupport` must indicate support for `EventConfiguration.DEVICE_KEY_SUPPORT_BACKWARD`, and `knobKeySupport` must indicate support for the following:

- `EventConfiguration.KNOB_KEY_SUPPORT_SHIFT_X_0`
- `EventConfiguration.KNOB_KEY_SUPPORT_SHIFT_Y_0`
- `EventConfiguration.KNOB_KEY_SUPPORT_PUSH_Z_0`
- `EventConfiguration.KNOB_KEY_SUPPORT_ROTATE_Z_0`

Parameters:

`keyboardLanguage` - The keyboard layout language code.

`keyboardCountry` - The keyboard layout country code.

`uiLanguage` - The user interface language code.

`uiCountry` - The user interface country code.

`knobKeySupport` - The knob key support, as a bitmask.

`deviceKeySupport` - The device key support, as a bitmask.

`multimediaKeySupport` - The multimedia key support, as a bitmask.

`miscKeySupport` - The miscellaneous key support, as a bitmask.

`pointerSupport` - The pointer / touchscreen support, as a bitmask.

Method Detail

getKeyboardLanguage

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

Retrieves the keyboard layout language code. The SDK normalizes the case of this field so that the value is always lowercase.

Returns:

The ISO 639-1 language code for the server's keyboard layout.

getKeyboardCountry

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

Retrieves the keyboard layout country code. For example, if the server has an American keyboard layout, then keyboardLanguage will be 'en' and keyboardCountry will be 'US'. The SDK normalizes the case of this field so that the value is always uppercase.

Returns:

The ISO 3166-1 country code for the server's keyboard layout.

getUiLanguage

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

Retrieves the user interface language code. The SDK normalizes the case of this field so that the value is always lowercase.

Returns:

The ISO 639-1 language code for the server's user interface language.

getUiCountry

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

Retrieves the user interface country code. For example, if the server's user interface is US English, then uiLanguage will be 'en' and uiCountry will be 'US'. The SDK normalizes the case of this field so that the value is always uppercase.

Returns:

The ISO 3166-1 country code for the server's user interface language.

getKnobKeySupport

```
public int getKnobKeySupport()
```

Retrieves the knob key support as a bitmask. The value is a bitfield made up of KNOB_KEY_SUPPORT_* values. For example, if the KNOB_KEY_SUPPORT_PUSH_Z_0 bit is set, then the server will respond to KeyEvents of type VNCMirrorLinkKeys.XK_KNOB_2D_SHIFT_PUSH_0.

Returns:

The server's support for knob key input.

getDeviceKeySupport

```
public int getDeviceKeySupport()
```

Retrieves the device key support as a bitmask. The value is a bitfield made up of DEVICE_KEY_SUPPORT_* values. For example, if the DEVICE_KEY_SUPPORT_SOFT_LEFT bit is set, then the server will response to KeyEvents of type VNCMirrorLinkKeys.XK_DEVICE_SOFT_LEFT.

Returns:

The server's support for MirrorLink device key input.

getMultimediaKeySupport

```
public int getMultimediaKeySupport()
```

Retrieves the multimedia key support as a bitmask. The value is a bitfield made up of MULTIMEDIA_KEY_SUPPORT_* values. For example, if the MULTIMEDIA_KEY_SUPPORT_PLAY bit is set, then the server will respond to KeyEvents of type VNCMirrorLinkKeys.XK_MULTIMEDIA_PLAY.

Returns:

The server's support for multimedia key input.

getMiscKeySupport

```
public int getMiscKeySupport()
```

Retrieves the miscellaneous key support as a bitmask.

The value is a bitfield made up of MISC_KEY_SUPPORT_* values. For example, if the MISC_KEY_SUPPORT_ITU bit is set, then the server will respond to KeyEvents with ITU keypad key symbols (e.g. VNCMirrorLinkKeys.XK_ITU_KEY_0 etc.).

Some features have been deprecated in MirrorLink 1.3. Viewer and server application should not attempt to use these features if they support MirrorLink 1.3 or above. See [EventConfiguration](#) for further information.

Returns:

The server's support for miscellaneous MirrorLink key input.

isITUKeySupported

```
public boolean isITUKeySupported()
```

Check whether ITU keypad events are supported.

Returns:

true if ITU keypad events are supported and false otherwise.

isEventMappingSupported

```
public boolean isEventMappingSupported()
```

Check whether event mapping is supported.

Returns:

true if event mapping is supported and false otherwise.

isVirtualKeyboardTriggerSupported

```
public boolean isVirtualKeyboardTriggerSupported()
```

Check whether the virtual keyboard trigger is supported.

Returns:

true if the virtual keyboard trigger is supported and false otherwise.

isKeyEventListingSupported

```
public boolean isKeyEventListingSupported()
```

Check whether the key event listing is supported.

Returns:

true if the key event listing is supported and false otherwise.

getNumFunctionKeysSupported

```
public int getNumFunctionKeysSupported()
```

Retrieves the number of function keys supported. The value is a byte (bits [15:8]) in misKeySupport.

Returns:

The number of function keys supported.

getPointerSupport

```
public int getPointerSupport()
```

Retrieves the pointer / touchscreen support as a bitmask. The value is a bitfield made up of POINTER_SUPPORT_* values. For example, if the POINTER_SUPPORT_POINTER_BUTTON_1 bit is set, then the server will respond to PointerEvents with the VNCPointerDevice.BUTTON_LEFT bit set.

Returns:

The server's support for pointer / touchscreen input.

toString

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

Returns a textual representation of this object.

Overrides:

toString in class java.lang.Object

Returns:

A textual representation of this object.

com.realvnc.mirrorlink

Class VNCViewerEventConfiguration

java.lang.Object
com.realvnc.mirrorlink.VNCViewerEventConfiguration

Deprecated.

```
@Deprecated
public class VNCViewerEventConfiguration
extends java.lang.Object
```

This class is deprecated and is of no use.

Constructor Summary

Constructors
Constructor and Description
<code>VNCViewerEventConfiguration()</code> Deprecated.

Method Summary

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

Constructor Detail

VNCViewerEventConfiguration
<code>public VNCViewerEventConfiguration()</code> Deprecated.

com.realvnc.util

Class IniFile.BadFormatException

java.lang.Object
 java.lang.Throwable
 java.lang.Exception
 com.realvnc.util.IniFile.BadFormatException

All Implemented Interfaces:

java.io.Serializable

Enclosing class:

IniFile

```
public class IniFile.BadFormatException
extends java.lang.Exception
```

See Also:

[Serialized Form](#)

Field Summary

Fields	
Modifier and Type	Field and Description
static long	<code>serialVersionUID</code>

Constructor Summary

Constructors	
Constructor and Description	
<code>IniFile.BadFormatException()</code>	

Method Summary

Methods inherited from class java.lang.Throwable	
<code>addSuppressed, fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, getSuppressed, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString</code>	
Methods inherited from class java.lang.Object	
<code>clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait</code>	

Field Detail

<code>serialVersionUID</code>

```
public static final long serialVersionUID
```

See Also:

[Constant Field Values](#)

Constructor Detail

IniFile.BadFormatException

```
public IniFile.BadFormatException()
```

com.realvnc.util

Class IniFile

java.lang.Object
com.realvnc.util.IniFile

public class **IniFile**
extends java.lang.Object

Nested Class Summary

Nested Classes

Modifier and Type	Class and Description
class	IniFile.BadFormatException

Field Summary

Fields

Modifier and Type	Field and Description
protected java.util.Vector<java.lang.String>	sectionNames
protected java.util.Hashtable<java.lang.String, java.util.Hashtable<java.lang.String, java.lang.String>>	sections

Constructor Summary

Constructors

Constructor and Description
IniFile()

Method Summary

Methods

Modifier and Type	Method and Description
java.util.Hashtable<java.lang.String, java.util.Hashtable<java.lang.String, java.lang.String>>	addSections()
java.lang.String	get(java.lang.String section, java.lang.String item)
void	parse(java.io.InputStream data)

Methods inherited from class java.lang.Object

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

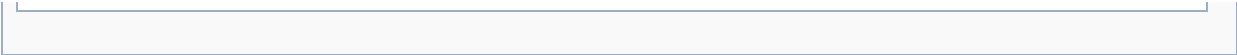
Field Detail

sections

protected java.util.Hashtable<java.lang.String, java.util.Hashtable<java.lang.String, java.lang.String>>
sections

sectionNames

protected java.util.Vector<java.lang.String> sectionNames



Constructor Detail

IniFile

```
public IniFile()
```

Method Detail

parse

```
public void parse(java.io.InputStream data)
    throws java.io.IOException,
           IniFile.BadFormatException
```

Throws:

- java.io.IOException
- IniFile.BadFormatException

get

```
public java.lang.String get(java.lang.String section,
                           java.lang.String item)
```

allSections

```
public java.util.Hashtable<java.lang.String,java.util.Hashtable<java.lang.String,java.lang.String>> allSections()
```

com.realvnc.util

Class VncLog

java.lang.Object
com.realvnc.util.VncLog

```
public class VncLog
extends java.lang.Object
```

Constructor Summary

Constructors
Constructor and Description
VncLog ()

Method Summary

Methods	
Modifier and Type	Method and Description
static void	<code>destroy()</code>
static void	<code>init(java.lang.String filePath)</code>
static void	<code>init(java.lang.String filePath, boolean newThread)</code>
static void	<code>setTestingMode(boolean testMode)</code>

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

Constructor Detail

VncLog
public VncLog ()

Method Detail

init
public static void init(java.lang.String filePath)

init
public static void init(java.lang.String filePath, boolean newThread)

destroy

```
public static void destroy()
```

setTestingMode

```
public static void setTestingMode(boolean testMode)
```


com.realvnc.vncserver.android

Class CustomRemoteControlServiceRequests

java.lang.Object
com.realvnc.vncserver.android.CustomRemoteControlServiceRequests

public class CustomRemoteControlServiceRequests
extends java.lang.Object

Custom requests supported by Remote Control Service implementations provided by RealVNC for Android platforms.

See Also:

VncServer.customRemoteControlServiceRequest(java.lang.String, android.os.Bundle),
VncServerCallbackHandler.customRemoteControlServiceCb(java.lang.String, android.os.Bundle),
RemoteControlServiceCodes

Field Summary

Fields

Modifier and Type	Field and Description
static java.lang.String	ENABLE_HEADS_UP_NOTIFICATIONS Globally enables/disables Android heads up notifications.
static java.lang.String	ENABLE_REMOTE_CONTROL Enables remote control.
static java.lang.String	SET_STATUS_BAR_INFO Disables (or re-enables) information items on the Android status bar.
static java.lang.String	SET_SYSTEM_UI_VISIBILITY Requests that the visibility of the status or navigation bars be changed, for all windows across all applications.

Constructor Summary

Constructors

Constructor and Description
CustomRemoteControlServiceRequests()

Method Summary

Methods inherited from class java.lang.Object

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

Field Detail

ENABLE_REMOTE_CONTROL

public static final java.lang.String ENABLE_REMOTE_CONTROL

Enables remote control.

The SDK will automatically enable remote control when starting a VNC session. However, if your application needs to make custom Remote Control Service requests before a VNC session is started, then you should send this request first.

This request requires no payload data to be provided by the caller.

If the request is understood, then a payload shall be returned with the following key:

- `err` with a `RemoteControlServiceCodes` return code describing the result of this request.

See Also:

[Constant Field Values](#)

SET_STATUS_BAR_INFO

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

Disables (or re-enables) information items on the Android status bar.

This request can be used to hide various information items on the status bar. This allows MirrorLink 1.3 servers and above to hide server status information that a MirrorLink client is already showing locally. Please see the MirrorLink specification for further information.

This request requires a payload to be provided by the caller with the following keys. If an optional key is not provided, then the state of the corresponding status bar information item is not changed.

- `v` with int value 1 for future compatibility.
- **Optional:** `disableBatteryLevel` with a boolean value, set to `true` if the battery level indicator should be disabled, `false` otherwise.
- **Optional:** `disableLocalTime` with a boolean value, set to `true` if the local time information item should be disabled, `false` otherwise.
- **Optional:** `disableNetworkOperator` with a boolean value, set to `true` if the network operator information item should be disabled, `false` otherwise.
- **Optional:** `disableNetworkSignalStrength` with a boolean value, set to `true` if the network signal strength indicator should be disabled, `false` otherwise.

If the request is understood, then a payload shall be returned with the following key:

- `err` with a `RemoteControlServiceCodes` return code describing the result of this request.

See Also:

[Constant Field Values](#)

SET_SYSTEM_UI_VISIBILITY

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

Requests that the visibility of the status or navigation bars be changed, for all windows across all applications.

This request can be used to globally hide the status and/or navigation bars. MirrorLink 1.3 servers and above should hide these system bars if their presence would reduce the MirrorLink client screen area available for applications to below that of the MirrorLink Reference Screen Display.

This request requires a payload to be provided by the caller with the following keys. If an optional key is not provided, then the visibility of the corresponding system bar is not changed.

- `v` with int value 1 for future compatibility.
- **Optional:** `hideStatusBar` with a boolean value, set to `true` if the status bar should be hidden, `false` otherwise.
- **Optional:** `hideNavigationBar` with a boolean value, set to `true` if the navigation bar should be hidden, `false` otherwise.

If the request is understood, then a payload shall be returned with the following key:

- `err` with a `RemoteControlServiceCodes` return code describing the result of this request.

See Also:

[Constant Field Values](#)**ENABLE_HEADS_UP_NOTIFICATIONS**

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

Globally enables/disables Android heads up notifications.

This request can be used to globally enable/disable Android heads up notifications. This can be useful in MirrorLink for example, where the server application has to comply with the driver distraction regulation.

This request requires a payload to be provided by the caller with the following keys:

- `v` with int value 1 for future compatibility.
- **Optional:** `enable` with a boolean value, set to `true` if heads up notifications should be enabled, `false` otherwise.

If the request is understood, then a payload shall be returned with the following key:

- `err` with a [RemoteControlServiceCodes](#) return code describing the result of this request.

See Also:[Constant Field Values](#)**Constructor Detail****CustomRemoteControlServiceRequests**

```
public CustomRemoteControlServiceRequests()
```

com.realvnc.vncserver.android

Class MirrorLinkCallbackHandler

java.lang.Object
com.realvnc.vncserver.android.VncServerCallbackHandler
com.realvnc.vncserver.android.MirrorLinkCallbackHandler

All Implemented Interfaces:

VncServerListener, VncServerMirrorLinkListener, VncServerOrientationListener

public abstract class **MirrorLinkCallbackHandler**
extends VncServerCallbackHandler
implements VncServerMirrorLinkListener

Base class for callbacks received for a MirrorLink server.

All of the callbacks are made from the main system dispatch thread after all pending events have been processed. This means that although the VncServer object uses a number of threads internally you can be sure that the callbacks will be made from a single thread and so will not be made while your application is mid-call in the server.

The callbacks defined here have a default empty implementation, so it is not needed for a MirrorLink Server to provide an implementation if the callback is not of interest.

As the display orientation interface is of interest to the MirrorLink servers, this class also implements that interface.

All the methods from VncServerMirrorLinkListener are kept abstract, but any new methods defined there will have a default implementation.

Constructor Summary

Constructors

Constructor and Description

MirrorLinkCallbackHandler()

Method Summary

Methods

Modifier and Type

Method and Description

void

mlFramebufferUnblockedCb()

The framebuffer is no longer blocked by the client.

Methods inherited from class com.realvnc.vncserver.android.VncServerCallbackHandler

customRemoteControlServiceCb, restoreOrientationLockCb

Methods inherited from class java.lang.Object

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

Methods inherited from interface com.realvnc.vncserver.android.VncServerMirrorLinkListener

mlAudioBlockingNotificationReceivedCb, mlClientDisplayConfigurationReceivedCb, mlClientEventConfigurationReceivedCb, mlConnectionReceivedCb, mlDeviceStatusSendNeededCb, mlEventMappingRequestReceivedCb, mlFramebufferBlockingNotificationReceivedCb

Methods inherited from interface com.realvnc.vncserver.android.VncServerListener

authCb, connectedCb, connectingCb, disconnectedCb, errorCallback, keygenCb, listeningCb, loginCb, remoteControlAvailableCb, remoteKeyCb, runningCb

Methods inherited from**interface com.realvnc.vncserver.android.VncServerOrientationListener**

displayOrientationChangedCb, displayOrientationChangeNeededCb

Constructor Detail**MirrorLinkCallbackHandler**

```
public MirrorLinkCallbackHandler()
```

Method Detail**mlFramebufferUnblockedCb**

```
public void mlFramebufferUnblockedCb()
```

The framebuffer is no longer blocked by the client.

This is called by the VNC Server SDK when the MirrorLink Client stops sending Blocking Notifications. This means that the framebuffer is being retrieved by the client without any part of it being blocked, or out of focus.

The Client does not send a explicit Unblock message, but the SDK will work out when the framebuffer is unblocked by monitoring the Blocking Notifications and the Framebuffer Update Requests.

See Also:

```
VncServerMirrorLinkListener.mlFramebufferBlockingNotificationReceivedCb(android.graphics.Rect,  
com.realvnc.mirrorlink.VNCFramebufferBlockingNotification)
```

com.realvnc.vncserver.android

Class RemoteControlServiceCodes

java.lang.Object
com.realvnc.vncserver.android.RemoteControlServiceCodes

```
public class RemoteControlServiceCodes
extends java.lang.Object
```

Return or error codes that may be reported by Remote Control Service implementations provided by RealVNC for Android platforms.

See Also:

```
VncServer.customRemoteControlServiceRequest(java.lang.String, android.os.Bundle),
VncServerCallbackHandler.customRemoteControlServiceCb(java.lang.String, android.os.Bundle),
CustomRemoteControlServiceRequests
```

Field Summary

Fields	
Modifier and Type	Field and Description
static int	RC_CAPTURE_TEMPORARILY_UNAVAILABLE Return code indicating failure due to a transient reason.
static int	RC_DEVICE_ADMIN_NOT_ENABLED Return code indicating failure due to administration not being enabled.
static int	RC_DISCONNECTED Return code indicating failure due to being disconnected from the service.
static int	RC_INCREMENTAL_UPDATES_UNAVAILABLE Return code indicating failure due to incremental updates not being available.
static int	RC_PERMISSION_DENIED Return code indicating failure due to the calling package not being granted permissions.
static int	RC_SERVICE_ILLEGAL_ARGUMENT Return code indicating failure due to the an illegal argument.
static int	RC_SERVICE_ITSELF_LACKING_PERMISSIONS Return code indicating failure due to the application implementing the service not having sufficient permissions.
static int	RC_SERVICE_LACKING_OTHER_OS_FACILITIES Return code indicating failure due to the operating system not providing the required functionality.
static int	RC_SERVICE_UNAVAILABLE Return code indicating failure due to the service not being available.
static int	RC_SUCCESS Return code indicating success.

Constructor Summary

Constructors	
Constructor and Description	
RemoteControlServiceCodes()	

Method Summary

Methods inherited from class java.lang.Object

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

Field Detail**RC_SUCCESS**

```
public static final int RC_SUCCESS
```

Return code indicating success.

See Also:

[Constant Field Values](#)

RC_CAPTURE_TEMPORARILY_UNAVAILABLE

```
public static final int RC_CAPTURE_TEMPORARILY_UNAVAILABLE
```

Return code indicating failure due to a transient reason.

This value indicates that the failure is only temporary, e.g. attempting to capture the screen when a secure window is visible.

See Also:

[Constant Field Values](#)

RC_PERMISSION_DENIED

```
public static final int RC_PERMISSION_DENIED
```

Return code indicating failure due to the calling package not being granted permissions.

This may be due to the user rejecting the request.

See Also:

[Constant Field Values](#)

RC_DEVICE_ADMIN_NOT_ENABLED

```
public static final int RC_DEVICE_ADMIN_NOT_ENABLED
```

Return code indicating failure due to administration not being enabled.

See Also:

[Constant Field Values](#)

RC_SERVICE_UNAVAILABLE

```
public static final int RC_SERVICE_UNAVAILABLE
```

Return code indicating failure due to the service not being available.

See Also:

[Constant Field Values](#)

RC_DISCONNECTED

```
public static final int RC_DISCONNECTED
```

Return code indicating failure due to being disconnected from the service.

See Also:

[Constant Field Values](#)

RC_INCREMENTAL_UPDATES_UNAVAILABLE

```
public static final int RC_INCREMENTAL_UPDATES_UNAVAILABLE
```

Return code indicating failure due to incremental updates not being available.

See Also:

[Constant Field Values](#)

RC_SERVICE_ITSELF_LACKING_PERMISSIONS

```
public static final int RC_SERVICE_ITSELF_LACKING_PERMISSIONS
```

Return code indicating failure due to the application implementing the service not having sufficient permissions.

See Also:

[Constant Field Values](#)

RC_SERVICE_LACKING_OTHER_OS_FACILITIES

```
public static final int RC_SERVICE_LACKING_OTHER_OS_FACILITIES
```

Return code indicating failure due to the operating system not providing the required functionality.

See Also:

[Constant Field Values](#)

RC_SERVICE_ILLEGAL_ARGUMENT

```
public static final int RC_SERVICE_ILLEGAL_ARGUMENT
```

Return code indicating failure due to the an illegal argument.

See Also:

[Constant Field Values](#)

Constructor Detail**RemoteControlServiceCodes**

```
public RemoteControlServiceCodes()
```


com.realvnc.vncserver.android

Class VncCommandString

java.lang.Object
com.realvnc.vncserver.core.VncCommandStringBase
com.realvnc.vncserver.android.VncCommandString

public class VncCommandString
extends VncCommandStringBase

Android implementation encapsulating a VNC command string. Note that in the past the terms "connection string" and "command string" have been used more-or-less interchangeably; both terms refer to the same thing.

Field Summary

Fields inherited from class com.realvnc.vncserver.core.VncCommandStringBase
fields, TYPE, VERSION

Constructor Summary

Constructors
Constructor and Description
VncCommandString() Create a new object representing an initially empty command string.

Method Summary

Methods

Modifier and Type	Method and Description
protected byte[]	<code>decodeBase64</code> (java.lang.String input) Take a base 64 encoded string and return the decoded output.

Methods inherited from class com.realvnc.vncserver.core.VncCommandStringBase

<code>getBase64Value</code> , <code>getBoolean</code> , <code>getInt</code> , <code>getString</code> , <code>parameterPresent</code> , <code>parse</code> , <code>putField</code>

Methods inherited from class java.lang.Object

<code>clone</code> , <code>equals</code> , <code>finalize</code> , <code>getClass</code> , <code>hashCode</code> , <code>notify</code> , <code>notifyAll</code> , <code>toString</code> , <code>wait</code> , <code>wait</code> , <code>wait</code>

Constructor Detail

VncCommandString
public VncCommandString() Create a new object representing an initially empty command string.

Method Detail

decodeBase64

```
protected byte[] decodeBase64(java.lang.String input)
                        throws java.lang.Exception
```

Take a base 64 encoded string and return the decoded output. Pads the input to a 4 character boundary before decoding.

Specified by:

`decodeBase64` in class `VncCommandStringBase`

Parameters:

`input` - base 64 encoded string to be decoded

Returns:

decoded output

Throws:

`java.lang.Exception` - if an error occurs during the decoding

com.realvnc.vncserver.android

Interface VncContextInformationManager.AccessibilityServiceProvider

Enclosing class:

VncContextInformationManager

public static interface VncContextInformationManager.AccessibilityServiceProvider

This interface represents a class that facilitates the usage of an accessibility service for context information gathering.

Method Summary

Methods	
Modifier and Type	Method and Description
void	accessibilityServiceRequired() Called when the accessibility service is required to receive context information.

Method Detail

accessibilityServiceRequired
void accessibilityServiceRequired() Called when the accessibility service is required to receive context information.

com.realvnc.vncserver.android

Interface VncContextInformationManager.CapturedContextInformation

Enclosing class:

VncContextInformationManager

public static interface VncContextInformationManager.CapturedContextInformation

Interface describing the context information for an area of the device screen.

Method Summary

Methods	
Modifier and Type	Method and Description
android.content.ComponentName	<code>getActivity()</code> Retrieve the component name of the activity responsible for the area of the screen.
int	<code>getFlags()</code> Retrieve a combination of flags describing this context information.
android.graphics.Rect	<code>getRect()</code> Retrives the area of the screen covered by this context information.

Method Detail

getActivity

```
android.content.ComponentName getActivity()
```

Retrieve the component name of the activity responsible for the area of the screen.

This will not correspond to an installed package if the `VncContextInformationManager.CONTEXT_FLAG_SYSTEM_UI` flag is set. Instead one of the constants defined in `VncContextInformationManager` will be used.

Returns:

The class name and package of the activity.

getRect

```
android.graphics.Rect getRect()
```

Retrives the area of the screen covered by this context information.

As this only describes a rectangle, not all of the pixels contained within the rectangle will necessarily be created by the component described by this object.

Returns:

The rectangle of the screen.

getFlags

```
int getFlags()
```

Retrieve a combination of flags describing this context information.

Returns:

A bitwise-or of the applicable flags.

com.realvnc.vncserver.android

Class VncContextInformationManager

java.lang.Object
com.realvnc.vncserver.android.VncContextInformationManager

public abstract class VncContextInformationManager
extends java.lang.Object

This class provides management of the context information for the applications, view and windows present on the display.

Nested Class Summary

Nested Classes

Modifier and Type	Class and Description
static interface	VncContextInformationManager.AccessibilityServiceProvider This interface represents a class that facilitates the usage of an accessibility service for context information gathering.
static interface	VncContextInformationManager.CapturedContextInformation Interface describing the context information for an area of the device screen.
static interface	VncContextInformationManager.Listener This interface allows objects to be notified of changes to the context information for the visual elements of the screen.
static class	VncContextInformationManager.ListenerPriority Enum to indicate priority of listeners.

Field Summary

Fields

Modifier and Type	Field and Description
static int	CHANGE_FLAG_ESTIMATED Constant flag used in VncContextInformationManager.Listener.contextInformationChanged(java.util.List,int) to indicate that the context information provided in the callback is only an estimate of the pixel contents of the screen.
static int	CHANGE_FLAG_POLLED Constant flag used in VncContextInformationManager.Listener.contextInformationChanged(java.util.List,int) to indicate that the context information is being polled and may not correspond to the exact pixel data being sent to the viewer.
static int	CHANGE_FLAG_SYNCHRONOUS Constant flag used in VncContextInformationManager.Listener.contextInformationChanged(java.util.List,int) to indicate that the pixel data for the captured context information won't be sent until the callback has returned.
static java.lang.String	CLASS_BUTTON_BAR Constant used as the class name for the button bar softkeys on the display.
static java.lang.String	CLASS_KEYGUARD Constant used as the class name for the keyguard component.
static java.lang.String	CLASS_STATUS_BAR Constant used as the class name for the status bar, which contains notification.
static java.lang.String	CLASS_TOAST Constant used as the class name for toast messages provided by android.widget.Toast.
static int	CONTEXT_FLAG_SYSTEM_UI Constant flag used in VncContextInformationManager.CapturedContextInformation.getFlags() to indicate that the context rectangle is from a system UI element.
static java.lang.String	PACKAGE_SYSTEM Constant used as the package name for system UI elements which are part of the system and so do not have an associated Android package.

Constructor Summary

Constructors

Constructor and Description
<code>VncContextInformationManager()</code>

Method Summary

Methods	
Modifier and Type	Method and Description
abstract void	<code>addAccessibilityServiceProvider(VncContextInformationManager.AccessibilityServiceProvider provider)</code> Add a new accessibility service provider.
abstract void	<code>addListener(VncContextInformationManager.Listener listener)</code> Add a new listener to receive notification of changes to context information.
abstract void	<code>addListener(VncContextInformationManager.Listener listener, VncContextInformationManager.ListenerPriority priority)</code> Add a new listener to receive notification of changes to context information.
abstract void	<code>removeAccessibilityServiceProvider(VncContextInformationManager.AccessibilityServiceProvider provider)</code> Removes a previously added accessibility service provider.
abstract void	<code>removeListener(VncContextInformationManager.Listener listener)</code> Removes a previously added listener, preventing it from receiving notification of changes to context information.
Methods inherited from class java.lang.Object	
<code>clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait</code>	

Field Detail

PACKAGE_SYSTEM
<code>public static final java.lang.String PACKAGE_SYSTEM</code> Constant used as the package name for system UI elements which are part of the system and so do not have an associated Android package. Only used when <code>CONTEXT_FLAG_SYSTEM_UI</code> is set. See Also: Constant Field Values
CLASS_KEYGUARD
<code>public static final java.lang.String CLASS_KEYGUARD</code> Constant used as the class name for the keyguard component. This is required on Android systems where the keyguard does not have an associated Android package. Only used when <code>CONTEXT_FLAG_SYSTEM_UI</code> is set. See Also: Constant Field Values
CLASS_TOAST
<code>public static final java.lang.String CLASS_TOAST</code> Constant used as the class name for toast messages provided by <code>android.widget.Toast</code> . The source package for the toast will be used as the package name. Only used when <code>CONTEXT_FLAG_SYSTEM_UI</code> is set. See Also: Constant Field Values

CLASS_BUTTON_BAR

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

Constant used as the class name for the button bar softkeys on the display.

Only used when `CONTEXT_FLAG_SYSTEM_UI` is set.

See Also:

[Constant Field Values](#)

CLASS_STATUS_BAR

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

Constant used as the class name for the status bar, which contains notification.

Only used when `CONTEXT_FLAG_SYSTEM_UI` is set.

See Also:

[Constant Field Values](#)

CONTEXT_FLAG_SYSTEM_UI

```
public static final int CONTEXT_FLAG_SYSTEM_UI
```

Constant flag used in `VncContextInformationManager.CapturedContextInformation.getFlags()` to indicate that the context rectangle is from a system UI element.

See Also:

[Constant Field Values](#)

CHANGE_FLAG_POLLED

```
public static final int CHANGE_FLAG_POLLED
```

Constant flag used in `VncContextInformationManager.Listener.contextInformationChanged(java.util.List, int)` to indicate that the context information is being polled and may not correspond to the exact pixel data being sent to the viewer.

See Also:

[Constant Field Values](#)

CHANGE_FLAG_SYNCHRONOUS

```
public static final int CHANGE_FLAG_SYNCHRONOUS
```

Constant flag used in `VncContextInformationManager.Listener.contextInformationChanged(java.util.List, int)` to indicate that the pixel data for the captured context information won't be sent until the callback has returned.

See Also:

[Constant Field Values](#)

CHANGE_FLAG_ESTIMATED

```
public static final int CHANGE_FLAG_ESTIMATED
```

Constant flag used in `VncContextInformationManager.Listener.contextInformationChanged(java.util.List, int)` to indicate that the context information provided in the callback is only an estimate of the pixel contents of the screen.

See Also:

[Constant Field Values](#)

Constructor Detail

VncContextInformationManager

```
public VncContextInformationManager()
```

Method Detail**addListener**

```
public abstract void addListener(VncContextInformationManager.Listener listener)
```

Add a new listener to receive notification of changes to context information.

When first added a listener will receive a callback containing the current context information, even if it has not changed. The listener is added with NORMAL priority.

Parameters:

`listener` - The listener to add.

addListener

```
public abstract void addListener(VncContextInformationManager.Listener listener,  
                                VncContextInformationManager.ListenerPriority priority)
```

Add a new listener to receive notification of changes to context information.

When first added a listener will receive a callback containing the current context information, even if it has not changed.

Parameters:

`listener` - The listener to add.

`priority` - The priority of the listener. Listeners with priority HIGHEST are triggered first.

removeListener

```
public abstract void removeListener(VncContextInformationManager.Listener listener)
```

Removes a previously added listener, preventing it from receiving notification of changes to context information.

Parameters:

`listener` - The listener to remove.

addAccessibilityServiceProvider

```
public  
abstract void addAccessibilityServiceProvider(VncContextInformationManager.AccessibilityServiceProvider provider)
```

Add a new accessibility service provider.

When first added a provider will receive a callback requesting the accessibility service if it is already required.

Parameters:

`provider` - The provider to add.

removeAccessibilityServiceProvider

```
public  
abstract void removeAccessibilityServiceProvider(VncContextInformationManager.AccessibilityServiceProvider provider)
```

Removes a previously added accessibility service provider.

Parameters:

`provider` - The provider to remove.

com.realvnc.vncserver.android

Interface VncContextInformationManager.Listener

Enclosing class:

VncContextInformationManager

public static interface VncContextInformationManager.Listener

This interface allows objects to be notified of changes to the context information for the visual elements of the screen.

Method Summary

Methods	
Modifier and Type	Method and Description
void	<code>contextInformationChanged</code> (java.util.List<VncContextInformationManager.CapturedContextInformation> items, int flags) Called when the current context information has changed.

Method Detail

contextInformationChanged

```
void contextInformationChanged(java.util.List<VncContextInformationManager.CapturedContextInformation> items,
                               int flags)
```

Called when the current context information has changed.

The list of items is in increasing z-order: i.e. higher indexed items in the list will generally cover earlier items. However as visual elements on the screen could be transparent this should not be relied upon.

The flags used by this function are defined in `VncContextInformationManager` and all start with `CHANGE_FLAG`.

Parameters:

`items` - A list of the current context information.

`flags` - A bitwise-or of the applicable flags.

com.realvnc.vncserver.android

Enum VncContextInformationManager.ListenerPriority

java.lang.Object
 java.lang.Enum<VncContextInformationManager.ListenerPriority>
 com.realvnc.vncserver.android.VncContextInformationManager.ListenerPriority

All Implemented Interfaces:

java.io.Serializable, java.lang.Comparable<VncContextInformationManager.ListenerPriority>

Enclosing class:

VncContextInformationManager

public static enum **VncContextInformationManager.ListenerPriority**
extends java.lang.Enum<VncContextInformationManager.ListenerPriority>

Enum to indicate priority of listeners. Listeners with higher priority are guaranteed to be executed first. Listeners with equal priority are called in an unspecified order.

Enum Constant Summary

Enum Constants	
Enum Constant and Description	
HIGHEST	
NORMAL	

Method Summary

Methods	
Modifier and Type	Method and Description
static VncContextInformationManager.ListenerPriority	valueOf(java.lang.String name) Returns the enum constant of this type with the specified name.
static VncContextInformationManager.ListenerPriority[]	values() Returns an array containing the constants of this enum type, in the order they are declared.

Methods inherited from class java.lang.Enum
clone, compareTo, equals, finalize, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

Methods inherited from class java.lang.Object
getClass, notify, notifyAll, wait, wait, wait

Enum Constant Detail

HIGHEST
public static final VncContextInformationManager.ListenerPriority HIGHEST

NORMAL

```
public static final VncContextInformationManager.ListenerPriority NORMAL
```

Method Detail**values**

```
public static VncContextInformationManager.ListenerPriority[] values()
```

Returns an array containing the constants of this enum type, in the order they are declared. This method may be used to iterate over the constants as follows:

```
for (VncContextInformationManager.ListenerPriority c :  
VncContextInformationManager.ListenerPriority.values())  
    System.out.println(c);
```

Returns:

an array containing the constants of this enum type, in the order they are declared

valueOf

```
public static VncContextInformationManager.ListenerPriority valueOf(java.lang.String name)
```

Returns the enum constant of this type with the specified name. The string must match *exactly* an identifier used to declare an enum constant in this type. (Extraneous whitespace characters are not permitted.)

Parameters:

`name` - the name of the enum constant to be returned.

Returns:

the enum constant with the specified name

Throws:

`java.lang.IllegalArgumentException` - if this enum type has no constant with the specified name

`java.lang.NullPointerException` - if the argument is null

com.realvnc.vncserver.android

Class VncDisplayInformationManager

java.lang.Object
com.realvnc.vncserver.android.VncDisplayInformationManager

public abstract class VncDisplayInformationManager
extends java.lang.Object

This class provides management of the display information related to the VNC session.

Nested Class Summary

Nested Classes	
Modifier and Type	Class and Description
static interface	VncDisplayInformationManager.Listener Listener interface used by the VNC Server Display Information Manager to notify the application that some of the display information has changed.

Constructor Summary

Constructors	
Constructor and Description	
VncDisplayInformationManager()	

Method Summary

Methods	
Modifier and Type	Method and Description
abstract void	addListener(VncDisplayInformationManager.Listener listener) Add a new listener to receive notification of changes to the display information.
abstract void	removeListener(VncDisplayInformationManager.Listener listener) Removes a previously added listener, preventing it from receiving notification of changes to display information.

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

Constructor Detail

VncDisplayInformationManager
public VncDisplayInformationManager()

Method Detail

addListener

```
public abstract void addListener(VncDisplayInformationManager.Listener listener)
```

Add a new listener to receive notification of changes to the display information.

When first added a listener will receive a callback containing the current context information, even if it has not changed.

Parameters:

`listener` - The listener to add.

removeListener

```
public abstract void removeListener(VncDisplayInformationManager.Listener listener)
```

Removes a previously added listener, preventing it from receiving notification of changes to display information.

Parameters:

`listener` - The listener to remove.

com.realvnc.vncserver.android

Interface VncDisplayInformationManager.Listener

Enclosing class:

VncDisplayInformationManager

public static interface **VncDisplayInformationManager.Listener**

Listener interface used by the VNC Server Display Information Manager to notify the application that some of the display information has changed.

All of the following callbacks will be made from the Handler provided to the [VncServer](#) constructor after all pending events have been processed. This means that although the [VncServer](#) object uses a number of threads internally you can be sure that the callbacks will be made from a single thread and so will not be made while your application is mid-call in the server.

Method Summary

Methods

Modifier and Type	Method and Description
void	screenSizeChangedCb (android.graphics.Rect newScreenSize) The screen size has changed.
void	sessionPixelFormatChangedCb (VncPixelFormat newPixelFormat) The pixel format used in the RFB session has changed.

Method Detail

screenSizeChangedCb

void screenSizeChangedCb(android.graphics.Rect newScreenSize)

The screen size has changed. This refers to the unscaled screen that gets captured for sending to the Client. This would typically be called when the orientation changes.

Parameters:

newScreenSize - The new size of the screen. This is never null, but if the screen size is not known, then it will be an empty Rect.

sessionPixelFormatChangedCb

void sessionPixelFormatChangedCb([VncPixelFormat](#) newPixelFormat)

The pixel format used in the RFB session has changed.

Parameters:

newPixelFormat - The new pixel format used. If the RFB session gets terminated, this will be null.

See Also:

[VncPixelFormat](#)

com.realvnc.vncserver.android

Interface VncExtension

```
public interface VncExtension
```

This is an opaque object used as a unique handle for an externally registered protocol extension.

It is returned from `registerExtension()` and should be passed to `sendExtensionMessage()` to send an extension message.

com.realvnc.vncserver.android

Interface VncExtensionListener

public interface VncExtensionListener

This interface is used for receiving externally defined protocol extension messages.

To use it, create a class implementing this interface and pass it to `registerExtension()`. When messages arrive, or when the extension is enabled or disabled, the VNC server will invoke these methods.

Method Summary

Methods	
Modifier and Type	Method and Description
void	<code>extensionEnabled(VncServer server, VncExtension extension, boolean enabledFlag)</code> Handle an incoming extension enable or disable message.
void	<code>extensionMessageReceived(VncServer server, VncExtension extension, byte[] payload, int payloadOffset, int payloadLength)</code> Handle an incoming extension message.

Method Detail

extensionEnabled

```
void extensionEnabled(VncServer server,
                     VncExtension extension,
                     boolean enabledFlag)
```

Handle an incoming extension enable or disable message.

Parameters:

- `server` - The VNC server instance which received the message.
- `extension` - The extension handle returned by `registerExtension()`.
- `enabledFlag` - True if the peer has enabled the extension; false if it is now disabled.

extensionMessageReceived

```
void extensionMessageReceived(VncServer server,
                             VncExtension extension,
                             byte[] payload,
                             int payloadOffset,
                             int payloadLength)
```

Handle an incoming extension message.

Parameters:

- `server` - The VNC server instance which received the message.
- `extension` - The extension handle returned by `registerExtension()`.
- `payload` - Byte array containing the received message.
- `payloadOffset` - Starting index of the message within the buffer.

`payloadLength` - Length of the message in bytes.

com.realvnc.vncserver.android

Interface VncH264Encoder.BufferOwner

Enclosing class:

VncH264Encoder

public static interface VncH264Encoder.BufferOwner

Method Summary

Methods	
Modifier and Type	Method and Description
void	<code>giveBuffer</code> (java.nio.ByteBuffer buffer) Transfers ownership of the provided buffer.

Method Detail

giveBuffer
<code>void giveBuffer</code> (java.nio.ByteBuffer buffer) Transfers ownership of the provided buffer. Parameters: buffer - The buffer to be given.

com.realvnc.vncserver.android

Enum VncH264Encoder.FrameType

java.lang.Object
java.lang.Enum<VncH264Encoder.FrameType>
com.realvnc.vncserver.android.VncH264Encoder.FrameType

All Implemented Interfaces:

java.io.Serializable, java.lang.Comparable<VncH264Encoder.FrameType>

Enclosing class:

VncH264Encoder

public static enum **VncH264Encoder.FrameType**
extends java.lang.Enum<VncH264Encoder.FrameType>

Enum Constant Summary

Enum Constants	
Enum Constant and Description	
FRAME_TYPE_IFRAME_AND_PS	The H.264 Parameter Sets (PPS and SPS) followed by an I-frame.
FRAME_TYPE_PFRAME	A P-frame.

Method Summary

Methods	
Modifier and Type	Method and Description
static VncH264Encoder.FrameType	valueOf(java.lang.String name) Returns the enum constant of this type with the specified name.
static VncH264Encoder.FrameType[]	values() Returns an array containing the constants of this enum type, in the order they are declared.

Methods inherited from class java.lang.Enum
clone, compareTo, equals, finalize, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

Methods inherited from class java.lang.Object
getClass, notify, notifyAll, wait, wait, wait

Enum Constant Detail

FRAME_TYPE_IFRAME_AND_PS
public static final VncH264Encoder.FrameType FRAME_TYPE_IFRAME_AND_PS The H.264 Parameter Sets (PPS and SPS) followed by an I-frame.

FRAME_TYPE_PFRAME

```
public static final VncH264Encoder.FrameType FRAME_TYPE_PFRAME
```

A P-frame.

Method Detail**values**

```
public static VncH264Encoder.FrameType[] values()
```

Returns an array containing the constants of this enum type, in the order they are declared. This method may be used to iterate over the constants as follows:

```
for (VncH264Encoder.FrameType c : VncH264Encoder.FrameType.values())  
    System.out.println(c);
```

Returns:

an array containing the constants of this enum type, in the order they are declared

valueOf

```
public static VncH264Encoder.FrameType valueOf(java.lang.String name)
```

Returns the enum constant of this type with the specified name. The string must match *exactly* an identifier used to declare an enum constant in this type. (Extraneous whitespace characters are not permitted.)

Parameters:

`name` - the name of the enum constant to be returned.

Returns:

the enum constant with the specified name

Throws:

`java.lang.IllegalArgumentException` - if this enum type has no constant with the specified name

`java.lang.NullPointerException` - if the argument is null

com.realvnc.vncserver.android

Class VncH264Encoder

java.lang.Object
com.realvnc.vncserver.android.VncH264Encoder

public abstract class VncH264Encoder
extends java.lang.Object

This class can be extended to implement an H.264 encoder.

Nested Class Summary

Nested Classes	
Modifier and Type	Class and Description
static interface	VncH264Encoder.BufferOwner
static class	VncH264Encoder.FrameType
static interface	VncH264Encoder.ScreenGrabHelper

Field Summary

Fields	
Modifier and Type	Field and Description
static int	H264_LEVEL_3_1 H.264 Level 3.1
static int	H264_LEVEL_4_1 H.264 Level 4.1
static int	H264_PROFILE_BASELINE H.264 Baseline Profile

Constructor Summary

Constructors	
Constructor and Description	
VncH264Encoder ()	

Method Summary

Methods	
Modifier and Type	Method and Description
abstract VncH264Encoder.FrameType	encodeFrame(VncH264Encoder.ScreenGrabHelper screenGrabHelper, VncH264Encoder.BufferOwner outputBufferOwner, int reservedBytesAtBufferStart) Encode the contents of the input surface to a ByteBuffer.
abstract int	getCurrentPipelineLag() Returns the number of stages in the encoding pipeline.

abstract VncSizeInt	<code>getNearestSupportedResolution</code> (int width, int height, int h264Level, int h264Profile) Returns the nearest supported resolution to the one specified, or null if no such resolution exists.
abstract boolean	<code>onIFrameRequired</code> () Requests that the encoder generate an I-frame with the H.264 parameter sets in the next call to <code>encodeFrame</code> ().
abstract boolean	<code>queryResolutionSupport</code> (int width, int height, int h264Level, int h264Profile) Check whether the specified resolution and H.264 parameters are supported.
abstract android.view.Surface	<code>startEncoder</code> (int width, int height, int h264Level, int h264Profile) Instructs the encoder to start, with the specified resolution and H.264 level.
abstract void	<code>stopEncoder</code> () Instructs the encoder to stop, and free any allocated resources.

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

Field Detail
<div>H264_LEVEL_3_1 public static final int H264_LEVEL_3_1 H.264 Level 3.1 See Also: Constant Field Values</div>
<div>H264_LEVEL_4_1 public static final int H264_LEVEL_4_1 H.264 Level 4.1 See Also: Constant Field Values</div>
<div>H264_PROFILE_BASELINE public static final int H264_PROFILE_BASELINE H.264 Baseline Profile See Also: Constant Field Values</div>

Constructor Detail
<div>VncH264Encoder public VncH264Encoder()</div>

Method Detail

queryResolutionSupport

```
public abstract boolean queryResolutionSupport(int width,
                                              int height,
                                              int h264Level,
                                              int h264Profile)
```

Check whether the specified resolution and H.264 parameters are supported.

Parameters:

`width` - the width of the frame to be encoded.

`height` - the height of the frame to be encoded.

`h264Level` - should be set to one of the `H264_LEVEL_*` constants. If the level is unrecognised, level 3.1 will be set.

`h264Profile` - should be set to one of the `H264_PROFILE_*` constants. If the profile is unrecognised, use Baseline.

Returns:

true if the specified resolution and H.264 parameters are supported. Otherwise, returns false.

getNearestSupportedResolution

```
public abstract VncSizeInt getNearestSupportedResolution(int width,
                                                         int height,
                                                         int h264Level,
                                                         int h264Profile)
```

Returns the nearest supported resolution to the one specified, or null if no such resolution exists.

The returned resolution must be smaller than the one specified on both axes.

Parameters:

`width` - the width of the frame to be encoded.

`height` - the height of the frame to be encoded.

`h264Level` - should be set to one of the `H264_LEVEL_*` constants. If the level is unrecognised, level 3.1 will be set.

`h264Profile` - should be set to one of the `H264_PROFILE_*` constants. If the profile is unrecognised, use Baseline.

Returns:

The nearest supported resolution to the one provided, or null if no such resolution exists.

startEncoder

```
public abstract android.view.Surface startEncoder(int width,
                                                  int height,
                                                  int h264Level,
                                                  int h264Profile)
```

Instructs the encoder to start, with the specified resolution and H.264 level.

The `h264Level` argument will be set to one of the `H264_LEVEL_*` constants. If the level is unrecognised, use level 3.1.

Parameters:

`width` - the width of the frame to be encoded.

height - the height of the frame to be encoded.

h264Level - should be set to one of the H264_LEVEL_* constants. If the level is unrecognised, level 3.1 will be set.

h264Profile - should be set to one of the H264_PROFILE_* constants. If the profile is unrecognised, use Baseline.

Returns:

the encoder's input surface.

onIFrameRequired

```
public abstract boolean onIFrameRequired()
```

Requests that the encoder generate an I-frame with the H.264 parameter sets in the next call to encodeFrame().

If the encoder cannot ensure that the next encoded frame will be an I-Frame with the H.264 parameter sets, this method must return false. Once false is returned, the SDK will restart the encoder if the next frame to be encoded is not the first one after startEncoder(). The encoder must ensure that the first encoded frame after startEncoder() is an I-Frame with the H.264 parameter sets if this method returns false.

The encoder should return true if it can generate an I-Frame with the H.264 parameter sets in the next encodeFrame() without encoder restarting.

No matter whether this method returns true or false, the encoder must ensure that the first call to encodeFrame() after this method generates an I-Frame with the H.264 parameter sets. The SDK will close VNC session if the encoder fails to do it.

Returns:

true if the encoder can generate an I-Frame in the next frame encoding without encoder restarting, false if the encoder cannot ensure an I-Frame in the next frame encoding but can generate an I-Frame in the first frame encoding after startEncoder().

encodeFrame

```
public
abstract VncH264Encoder.FrameType encodeFrame(VncH264Encoder.ScreenGrabHelper screenGrabHelper,
                                              VncH264Encoder.BufferOwner outputBufferOwner,
                                              int reservedBytesAtBufferStart)
```

Encode the contents of the input surface to a ByteBuffer.

The ByteBuffer to which the data is written must be allocated by the encoder (using ByteBuffer.allocateDirect()), and provided to outputBufferOwner.

The first call to this method after startEncoder() or onIFrameRequired() must cause the H.264 parameter sets to be written to the output buffer, followed by an I-frame. Subsequent calls should result in either a single P-frame or a single I-frame (again prefixed with the SPS and PPS) to be written. All NAL units should be in Annex B format.

The SDK owns the provided buffer until the next call to encodeFrame. The buffer must not be accessed or modified during this time.

Even if the same buffer is reused in consecutive calls, the outputBufferOwner.giveBuffer() method must be invoked on each call.

When this method returns, the contents of the buffer should be as follows:

```
+-----+
| ??? | Reserved for SDK | Encoded H.264 data | ??? |
+-----+
A      B                      C                      D      E
```

```
A = Zero
B = buffer.position()
C = buffer.position() + reservedBytesAtBufferStart
D = buffer.limit()
E = buffer.capacity()
```

```
A -> B      Arbitrary data, ignored by SDK
```

```
B -> C    Arbitrary data, reserved for SDK use
C -> D    Encoded H.264 data, written by encoder
D -> E    Arbitrary data, ignored by SDK
```

The encoder can choose the values B, D, and E, and must set the buffer's limit and position appropriately.

The encoded H.264 data must begin at C, and end at D.

A number of bytes (reservedBytesAtBufferStart) must be reserved between B and C. The SDK will use this space for header data.

For example, if 16 bytes are to be reserved, then the encoded data must start 16 bytes after the buffer's position(). If 100 bytes of encoded data are written in this case, then the buffer's limit must be set to 116 plus the position().

Parameters:

screenGrabHelper - an instance of `VncH264Encoder.ScreenGrabHelper`.

outputBufferOwner - an instance of `VncH264Encoder.BufferOwner` to receive the encoded data.

reservedBytesAtBufferStart - the number of bytes which should be reserved in the buffer before the encoded data.

Returns:

the frame type written.

getCurrentPipelineLag

```
public abstract int getCurrentPipelineLag()
```

Returns the number of stages in the encoding pipeline. This may change between frames.

This should typically be equal to the total number of times `ScreenGrabHelper.forceScreenGrab()` has been called. If the encoder's output does not lag behind the input (i.e. providing one frame as input immediately produces one frame as output) then zero should be returned.

Returns:

the number of stages in the encoding pipeline.

stopEncoder

```
public abstract void stopEncoder()
```

Instructs the encoder to stop, and free any allocated resources.

com.realvnc.vncserver.android

Interface VncH264Encoder.ScreenGrabHelper

Enclosing class:

VncH264Encoder

public static interface VncH264Encoder.ScreenGrabHelper

Method Summary

Methods	
Modifier and Type	Method and Description
boolean	<code>forceScreenGrab()</code> Triggers an extra screen grab, which will be rendered to the input Surface.

Method Detail

forceScreenGrab
<pre>boolean forceScreenGrab()</pre> <p>Triggers an extra screen grab, which will be rendered to the input Surface. Intended to be used where multiple frames are required to fill the encoder's pipeline.</p> <p>Returns:</p> <p>false if the grab failed, true otherwise.</p>

com.realvnc.vncserver.android

Interface VncMirrorLinkKeyEventListener

public interface **VncMirrorLinkKeyEventListener**

Listener interface allowing server SDK users to implement custom handling of MirrorLink key events.

Field Summary

Fields	
Modifier and Type	Field and Description
static int	FLAG_CLIENT_REPEAT Bitwise flag used in mlKeyEventReceived(int, int) to indicate that the key event is an automatic key repeat which has been generated at the client side.
static int	FLAG_KEY_DOWN Bitwise flag used in mlKeyEventReceived(int, int) to indicate that the key event is a key down event.
static int	FLAG_SERVER_REPEAT Bitwise flag used in mlKeyEventReceived(int, int) to indicate that the key event is an automatic key repeat which has been generated at the server side.

Method Summary

Methods	
Modifier and Type	Method and Description
boolean	mlKeyEventReceived (int value, int flags) Callback used to offer a MirrorLink key event to a registered listener.

Field Detail

FLAG_KEY_DOWN
<pre>static final int FLAG_KEY_DOWN</pre> <p>Bitwise flag used inmlKeyEventReceived(int, int) to indicate that the key event is a key down event.</p> <p>If this flag is not set then the key event is a key up event.</p> <p>If a key is repeating then these will be indicated with multiple down events. These can be detected by checking the FLAG_CLIENT_REPEAT and FLAG_SERVER_REPEAT flags.</p> <p>See Also:</p> <p>mlKeyEventReceived(int, int), Constant Field Values</p>
FLAG_CLIENT_REPEAT
<pre>static final int FLAG_CLIENT_REPEAT</pre> <p>Bitwise flag used inmlKeyEventReceived(int, int) to indicate that the key event is an automatic key repeat which has been generated at the client side.</p>

See Also:

`mlKeyEventReceived(int, int)`, [Constant Field Values](#)

FLAG_SERVER_REPEAT

```
static final int FLAG_SERVER_REPEAT
```

Bitwise flag used in `mlKeyEventReceived(int, int)` to indicate that the key event is an automatic key repeat which has been generated at the server side.

See Also:

`mlKeyEventReceived(int, int)`, [Constant Field Values](#)

Method Detail**mlKeyEventReceived**

```
boolean mlKeyEventReceived(int value,  
                           int flags)
```

Callback used to offer a MirrorLink key event to a registered listener.

This callback allows a user of the RealVNC server SDK to implement custom behaviour for MirrorLink keys, as defined in Appendix B of CCC-TS-010 v1.1, v1.2 and v1.3.

If this method returns true then the key event will not be passed to any other listeners and the default key handling behaviour will not be used.

If this method returns false then the key event will be passed onto other registered listeners. If no other listeners consume the key event then the default key handling behaviour of the server SDK will be used.

It is important for listeners to consume all key down and key up events for keys of interest. Failure to do so will lead to unexpected behaviour of the device, possibly in violation of the MirrorLink specification.

If, for example, a listener only wishes to act when the key is released it should still consume the key down event and any subsequent repeats to prevent other listeners or the default key behaviour from being invoked.

It is important to realise that handling of certain keys is a mandatory part of the MirrorLink specification. Consuming one of these key events and then not performing the required action may prevent the server from being certified as a MirrorLink device.

Parameters:

`value` - The value of the XKeySym.

`flags` - Flags describing the key event, see the `FLAG_*` constants defined in this interface for further details. Multiple `FLAG_*` values may be bitwise OR'd together to form this value.

Returns:

True if the event has been handled, false otherwise.

com.realvnc.vncserver.android

Class VncOrientationManager

java.lang.Object

com.realvnc.vncserver.android.VncOrientationManager

```
public abstract class VncOrientationManager
extends java.lang.Object
```

This class provides management of the orientation of the device display.

Field Summary

Fields

Modifier and Type	Field and Description
static int	ORIENTATION_DISABLE_LOCK Constant indicating that the orientation lock should be disabled.
static int	ORIENTATION_LANDSCAPE_LOCK Constant indicating that the orientation should be locked in landscape.
static int	ORIENTATION_PORTRAIT_LOCK Constant indicating that the orientation should be locked in portrait.

Constructor Summary

Constructors

Constructor and Description
VncOrientationManager()

Method Summary

Methods

Modifier and Type	Method and Description
abstract void	lockOrientation (int orientation) Deprecated. <i>Use lockOrientationEx(int) instead</i>
abstract void	lockOrientationEx (int orientation) Requests that the screen orientation is locked to the requested orientation.
abstract boolean	restoreOrientationLock (int orientation) Deprecated. <i>Use restoreOrientationLockEx(int) instead. Requests that the screen orientation lock is restored to the requested value once the VNC connection is over.</i> <i>This method can be called at any point of the VNC connection, but it is recommended to be used as soon as the connection has started, so that when the VNC connection terminates the screen orientation lock holds the same value it did before it started.</i> <i>The orientation provided should be one of the constants defined in this class:</i> ORIENTATION_DISABLE_LOCK , ORIENTATION_LANDSCAPE_LOCK or ORIENTATION_PORTRAIT_LOCK . <i>This feature was introduced in version 3.4. Older versions of the RCS default to disabling screen</i>

orientation lock at the end of a session. For compatibility, if this method is not called, that behaviour is reproduced (i.e. the orientation lock will always be restored to [ORIENTATION_DISABLE_LOCK](#).

If restoring the orientation lock is supported for the current set-up, this method will return `true`. Otherwise it will return `false`.

This method is deprecated because it will fail if invoked too early,

abstract void

`restoreOrientationLockEx(int orientation)`

Requests that the screen orientation lock is restored to the requested value once the VNC connection is over.

Methods inherited from class java.lang.Object

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

Field Detail

ORIENTATION_DISABLE_LOCK

`public static final int ORIENTATION_DISABLE_LOCK`

Constant indicating that the orientation lock should be disabled.

See Also:

`lockOrientation(int)`, [Constant Field Values](#)

ORIENTATION_LANDSCAPE_LOCK

`public static final int ORIENTATION_LANDSCAPE_LOCK`

Constant indicating that the orientation should be locked in landscape.

Landscape orientations are wider than they are tall.

See Also:

`lockOrientation(int)`, [Constant Field Values](#)

ORIENTATION_PORTRAIT_LOCK

`public static final int ORIENTATION_PORTRAIT_LOCK`

Constant indicating that the orientation should be locked in portrait.

Portrait orientations are taller than they are wide.

See Also:

`lockOrientation(int)`, [Constant Field Values](#)

Constructor Detail

VncOrientationManager

`public VncOrientationManager()`

Method Detail

restoreOrientationLock

```
public abstract boolean restoreOrientationLock(int orientation)
```

Deprecated. Use `restoreOrientationLockEx(int)` instead. Requests that the screen orientation lock is restored to the requested value once the VNC connection is over.

This method can be called at any point of the VNC connection, but it is recommended to be used as soon as the connection has started, so that when the VNC connection terminates the screen orientation lock holds the same value it did before it started.

The orientation provided should be one of the constants defined in this class: `ORIENTATION_DISABLE_LOCK`, `ORIENTATION_LANDSCAPE_LOCK` or `ORIENTATION_PORTRAIT_LOCK`.

This feature was introduced in version 3.4. Older versions of the RCS default to disabling screen orientation lock at the end of a session. For compatibility, if this method is not called, that behaviour is reproduced (i.e. the orientation lock will always be restored to `ORIENTATION_DISABLE_LOCK`).

If restoring the orientation lock is supported for the current set-up, this method will return `true`. Otherwise it will return `false`.

This method is deprecated because it will fail if invoked too early.

Parameters:

`orientation` - The orientation which should be restored at the end of the connection.

Returns:

`true` if the restoration is supported, `false` otherwise.

restoreOrientationLockEx

```
public abstract void restoreOrientationLockEx(int orientation)
```

Requests that the screen orientation lock is restored to the requested value once the VNC connection is over.

This method can be called at any point of the VNC connection, but it is recommended to be used as soon as the connection has started, so that when the VNC connection terminates the screen orientation lock holds the same value it did before it started.

The orientation provided should be one of the constants defined in this class: `ORIENTATION_DISABLE_LOCK`, `ORIENTATION_LANDSCAPE_LOCK` or `ORIENTATION_PORTRAIT_LOCK`.

Whether the orientation will be restored is notified asynchronously by `VncServerCallbackHandler.restoreOrientationLockCb(boolean, int)`. If no `VncServerCallbackHandler` object was passed to `VncServer`'s `create()` method, the callback cannot occur.

If several calls to this method are made, the last one to receive a successful `VncServerCallbackHandler.restoreOrientationLockCb(boolean, int)` will determine the orientation that will be restored.

This feature was introduced in version 3.4. Older versions of the RCS default to disabling screen orientation lock at the end of a session. For compatibility, if this method is not called, that behaviour is reproduced (i.e. the orientation lock will always be restored to `ORIENTATION_DISABLE_LOCK`).

Parameters:

`orientation` - The orientation which should be restored at the end of the connection.

lockOrientation

```
public abstract void lockOrientation(int orientation)
```

Deprecated. Use `lockOrientationEx(int)` instead

lockOrientationEx

```
public abstract void lockOrientationEx(int orientation)
                                throws VncException
```

Requests that the screen orientation is locked to the requested orientation.

The orientation provided should be one of the constants defined in this class: `ORIENTATION_DISABLE_LOCK`, `ORIENTATION_LANDSCAPE_LOCK` or `ORIENTATION_PORTRAIT_LOCK`.

To disable a previously requested orientation lock use the `ORIENTATION_DISABLE_LOCK` constant.

Not all applications support every orientation, so the actual orientation of the device should be monitored using a `VncServerOrientationListener`.

Parameters:

`orientation` - The requested orientation.

Throws:

`VncException`

com.realvnc.vncserver.android

Interface VncRemoteControlInfo

public interface VncRemoteControlInfo

Objects implementing this interface are used to provide detailed information about the forms of remote control, and can be obtained through a call to the `VncServer.getRemoteControlInfo()` method.

Method Summary

Methods	
Modifier and Type	Method and Description
boolean	<code>getKeyInjectionSupport()</code> Returns whether this form of remote control supports the injection of key events.
boolean	<code>getMotionInjectionSupport()</code> Returns whether this form of remote control supports the injection of motion events.

Method Detail

getMotionInjectionSupport
<code>boolean getMotionInjectionSupport()</code>
Returns whether this form of remote control supports the injection of motion events.

getKeyInjectionSupport
<code>boolean getKeyInjectionSupport()</code>
Returns whether this form of remote control supports the injection of key events.

com.realvnc.vncserver.android

Interface VncRemoteFeatureCheckListener

public interface VncRemoteFeatureCheckListener

This interface is used for receiving externally defined remote feature checks.

To use it, create a class implementing this interface and pass it to `addRemoteFeatureCheck()`. When a remote feature check is performed on a VNC viewer the VNC server will invoke these methods.

Method Summary

Methods	
Modifier and Type	Method and Description
boolean	<code>remoteFeatureCheckFailed(VncServer server, int featureCheckId)</code> Called when the VNC viewer has failed to pass a remote feature check.
void	<code>remoteFeatureCheckSucceeded(VncServer server, int featureCheckId, int featureId)</code> Called when the VNC viewer has successfully passed a remote feature check.

Method Detail

remoteFeatureCheckSucceeded

```
void remoteFeatureCheckSucceeded(VncServer server,
                                int featureCheckId,
                                int featureId)
```

Called when the VNC viewer has successfully passed a remote feature check.

Parameters:

`server` - The VNC server instance which received the message.

`featureCheckId` - The feature check identifier returned by `addRemoteFeatureCheck()`.

`featureId` - The ID of the feature which the viewer used to pass the check.

remoteFeatureCheckFailed

```
boolean remoteFeatureCheckFailed(VncServer server,
                                int featureCheckId)
```

Called when the VNC viewer has failed to pass a remote feature check.

If this callback returns true then the feature check failure will be treated as critical and the connection to the VNC viewer will be closed with an error indicating that it was not licensed.

Parameters:

`server` - The VNC server instance which received the message.

`featureCheckId` - The feature check identifier returned by `addRemoteFeatureCheck()`.

Returns:

true if this was a critical failure, false otherwise.

--

com.realvnc.vncserver.android

Class VncServerCallbackHandler

java.lang.Object

com.realvnc.vncserver.android.VncServerCallbackHandler

All Implemented Interfaces:

VncServerListener, VncServerOrientationListener

Direct Known Subclasses:

MirrorLinkCallbackHandler

```
public abstract class VncServerCallbackHandler
extends java.lang.Object
implements VncServerOrientationListener
```

Base class for callbacks received for a VNC server.

All of the callbacks are made from the main system dispatch thread after all pending events have been processed. This means that although the `VncServer` object uses a number of threads internally you can be sure that the callbacks will be made from a single thread and so will not be made while your application is mid-call in the server.

The callbacks defined here have a default empty implementation, so it is not needed for a Server to provide an implementation if the callback is not of interest.

All the methods from `VncServerOrientationListener` are kept abstract, but any new methods defined there will have a default implementation.

Constructor Summary

Constructors

Constructor and Description

`VncServerCallbackHandler()`

Method Summary

Methods

Modifier and Type	Method and Description
void	<code>customRemoteControlServiceCb</code> (java.lang.String name, android.os.Bundle payload) A callback indicating an asynchronous reply to a custom request sent to the Remote Control Service.
void	<code>restoreOrientationLockCb</code> (boolean success, int orientation) Called to notify whether a previous <code>VncOrientationManager.restoreOrientationLockEx(int)</code> request will be honoured or not.

Methods inherited from class java.lang.Object

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

Methods inherited from interface com.realvnc.vncserver.android.VncServerOrientationListener

`displayOrientationChangedCb`, `displayOrientationChangeNeededCb`

Methods inherited from interface com.realvnc.vncserver.android.VncServerListener

`authCb`, `connectedCb`, `connectingCb`, `disconnectedCb`, `errorCb`, `keygenCb`, `listeningCb`, `loginCb`, `remoteControlAvailableCb`, `remoteKeyCb`, `runningCb`

Constructor Detail**VncServerCallbackHandler**

```
public VncServerCallbackHandler()
```

Method Detail**customRemoteControlServiceCb**

```
public void customRemoteControlServiceCb(java.lang.String name,  
                                         android.os.Bundle payload)
```

A callback indicating an asynchronous reply to a custom request sent to the Remote Control Service.

This is called by the VNC Server SDK when it receives a callback from the Remote Control Service in response to a custom request.

Parameters:

`name` - The name of the request in standard Java reverse-DNS notation, for example "com.myandroidoem.EnablePhaseTractorBeams".

`payload` - Any payload required for this custom message. This can be null; it's up to the request to interpret this however it likes.

See Also:

`VncServer.customRemoteControlServiceRequest(java.lang.String, android.os.Bundle)`,
`CustomRemoteControlServiceRequests`, `RemoteControlServiceCodes`

restoreOrientationLockCb

```
public void restoreOrientationLockCb(boolean success,  
                                     int orientation)
```

Called to notify whether a previous `VncOrientationManager.restoreOrientationLockEx(int)` request will be honoured or not.

Parameters:

`success` - whether the orientation lock will be restored (true) or not.

`orientation` - the orientation that was requested to be restored.

com.realvnc.vncserver.android

Class VncServer

java.lang.Object
com.realvnc.vncserver.android.VncServer

public abstract class **VncServer**
extends java.lang.Object

This class provides the API for a VNC server. Implementing a VNC server starts with calling the `create` method to create a VNC server object. Once created, the server is controlled by calling the methods described on this page.

Field Summary

Fields	
Modifier and Type	Field and Description
static int	<code>FEATURE_CLIPBOARD</code> Whether or not support should be enabled for transferring clipboard information between the viewer and server.
static int	<code>FEATURE_COMPARE_FB</code> Enable or disable framebuffer comparison.
static int	<code>FEATURE_MIRRORLINK_FORBID_PORTRAIT_ORIENTATION</code> Whether or not to block frame buffer updates for MirrorLink connection when display orientation is portrait.
static int	<code>FEATURE_RFB4</code> Enables or disables RFB4 connections.
static int	<code>FEATURE_SEND_CLIPBOARD_ON_CONNECTION</code> If the clipboard feature is enabled, then defines whether or not the server clipboard contents should be sent to the viewer when a connection is first established.
static int	<code>FEATURE_START_IN_LANDSCAPE</code> Requests that the initial server configuration is reported as landscape.
static int	<code>FEATURE_VIEW_ONLY</code> On Windows mobile systems, tells the server to release all key/pointer presses and ignore further input.
static int	<code>FEATURE_WINCR_SET_DISPLAY_POLL_FREQUENCY</code> On Windows mobile systems, sets a refresh rate for the server screen polling.

Constructor Summary

Constructors	
Modifier	Constructor and Description
protected	<code>VncServer()</code> Do not use the constructor.

Method Summary

Methods	
Modifier and Type	Method and Description
abstract void	<code>accept(boolean ok)</code> Accept or reject a connection from a VNC viewer.
abstract void	<code>acceptRemoteKey(boolean ok)</code> Accept or reject an RSA key from a VNC viewer.
abstract void	<code>addBearer(java.lang.String bearerName, java.lang.Class<? extends VncBearer> bearerClass)</code> Register a new bearer with the SDK.
abstract void	<code>addBearer(java.lang.String bearerName, java.lang.String className)</code> Register a new bearer with the SDK.
abstract byte[]	<code>addLicense(java.lang.String licenseText)</code> Adds a license to the server.
abstract void	<code>addLicenseFeature(int featureId, byte[] featureKey)</code> Adds knowledge of a feature to the server.
abstract int	<code>addRemoteFeatureCheck(int[] featureIds, VncRemoteFeatureCheckListener listener)</code> Adds a remote feature check to this server instance.
abstract void	<code>authenticate(boolean ok)</code> Accept (or reject) a connection based on authentication credentials supplied to <code>authCb</code> .
abstract void	<code>bell()</code> Sends a bell message to the client.
abstract void	<code>blacklistRemoteControl(VncRemoteControlInfo remoteControl, boolean blacklist)</code> Prevents a particular type of remote control method from being used, or reinstates it.
abstract void	<code>checkRemoteControlAvailable()</code> Check whether the device has support for remote control.
abstract void	<code>connect(java.lang.String address, int port)</code> Establish a socket connection to a listening viewer using the VNC TCP/IP outbound pluggable bearer.
abstract void	<code>connect(VncCommandString commandString)</code> Establish a connection using the connection details and bearer specified in the command string.
static VncServer	<code>create(android.content.Context ctx, MirrorLinkCallbackHandler mListenerer)</code> Construct a new VNC server.
static VncServer	<code>create(android.content.Context ctx, MirrorLinkCallbackHandler mListenerer, android.os.Handler handler)</code> Construct a new VNC server.

static VncServer	create(android.content.Context ctx, VncServerCallbackHandler listener)
static VncServer	Construct a new VNC server. create(android.content.Context ctx, VncServerCallbackHandler listener, android.os.Handler handler)
static VncServer	Construct a new VNC server. create(android.content.Context ctx, VncServerListener listener)
static VncServer	Construct a new VNC server. create(android.content.Context ctx, VncServerListener listener, android.os.Handler handler)
static VncServer	Construct a new VNC server. create(android.content.Context ctx, VncServerOrientationListener listener)
static VncServer	Construct a new VNC server. create(android.content.Context ctx, VncServerOrientationListener listener, android.os.Handler handler)
static VncServer	Construct a new VNC server. create(android.content.Context ctx, VncServerOrientationListener listener, VncServerMirrorLinkListener mListener)
static VncServer	Construct a new VNC server. create(android.content.Context ctx, VncServerOrientationListener listener, VncServerMirrorLinkListener mListener, android.os.Handler handler)
abstract android.os.Bundle	Construct a new VNC server. customRemoteControlServiceRequest(java.lang.String name, android.os.Bundle payload)
abstract void	Sends a custom request to the remote control service. destroy()
abstract void	Destroy the the server object and release all associated resources. enableFeature(int feature, boolean enable)
abstract void	Selectively enable or disable optional features. freeze(boolean freeze)
abstract void	Freeze or thaw the server. generateKey(int keySize)
abstract int	Instruct the server to start generation of a new RSA encryption key. getAuthentication()
abstract java.util.Enumeraation<VncBearerInfo>	Get the currently selected authentication type. getBearerInfo()
abstract java.lang.String	Return an enumeration of objects implementing the VncBearerInfo interface. getBluetoothAdapterAddress()
abstract VncContextInformationManager	Gets the hardware address of the local Bluetooth adapter. getContextInformationManager()
abstract VncDisplayInformationManager	Retrieves the context information manager instance for this server object. getDisplayInformationManager()
abstract int	Retrieves the display information manager instance for this server object. getEncryption()
abstract VncOrientationManager	Get the currently selected encryption type. getOrientationManager()
abstract java.util.List<VncRemoteControlInfo>	Retrieves the orientation manager instance for this server object. getRemoteControlInfo()
abstract byte[]	Gets details of the device's remote control support. getServerSignature()
abstract int	Returns the signature for the server's RSA key. getState()
abstract java.lang.Object	Return the current state of the VNC server. getUserData()
abstract java.lang.String	Retrieves the user defined data previously associated with this server. getVersionString()
abstract void	Get the VNC server version string in form "major.minor.patch.revision number" injectKeyEvent(android.view.KeyEvent ev)
abstract boolean	Inject a keyboard event into the system. isConnectionTearingDown()
abstract void	Return true if the tearing down is in progress after either the existing connection is stopped to be processed or the VNC server instance stops listening for new connection. listen(int port)
abstract boolean	Starts this VNC server instance listening for an incoming connection using the VNC TCP/IP inbound pluggable bearer. localFeatureCheck(int[] featureIds)
abstract void	Performs a local feature check. login(java.lang.String username, java.lang.String password)
abstract void	Provide user name and/or password to viewer during reverse authentication. m1FrameBufferBlockingNotificationHandled()
abstract void	Deprecated. <i>Use m1FrameBufferBlockingNotificationHandled (VNCFramebufferBlockingNotification) instead.</i>
abstract void	m1FrameBufferBlockingNotificationHandled (VNCFramebufferBlockingNotification notification)
abstract java.util.Map<java.lang.Integer, VncEventMapping>	Indicates that processing of a framebuffer blocking notification message has completed. m1GetEventMapping()
abstract void	Returns the event mapping presently understood by the server. m1RegisterKeyEventListener (VncMirrorLinkKeyEventListener listener)
abstract void	Registers a MirrorLink key event listener for this server. m1RequestSendDeviceStatus()
abstract void	MirrorLink: Inform the server that a new device status request should be sent. m1SendDeviceStatus (VNCDeviceStatus deviceStatus)
abstract void	MirrorLink: send a 'device status' message to the viewer, in response to a VncServerMirrorLinkListener.m1DeviceStatusSendNeededCb (VNCDeviceStatus latestRequest, VNCDeviceStatus defaultReply) callback. m1SendEventMappingRequestReply (int clientKeyCode)
abstract void	Replies to an event mapping request received from the MirrorLink viewer. m1SendServerDisplayConfiguration (VNCServerDisplayConfiguration sdc)
abstract void	MirrorLink: Send a Server Device Configuration message. m1SendServerEventConfiguration (VNCServerEventConfiguration sec)
abstract void	MirrorLink: Send a Server Event Configuration message.

abstract void	<code>mlSetContextInformation</code> (<code>java.util.List<android.util.Pair<android.graphics.Rect, VNCContextInformation>></code> contextInformation) Tell the VNC Server SDK context about areas of the screen.
abstract void	<code>mlSetContextInformationAndInvalidate</code> (<code>java.util.List<android.util.Pair<android.graphics.Rect, VNCContextInformation>></code> contextInformation) Tell the VNC Server SDK context about areas of the screen.
abstract void	<code>mlSetEventMapping</code> (<code>java.util.Map<java.lang.Integer, java.lang.Integer></code> entries) Sets event mapping entries.
abstract void	<code>mlTriggerFramebufferBlockingNotification</code> (<code>VNCFramebufferBlockingNotification</code> notification) Triggers a server side <code>FramebufferBlockingNotification</code> .
abstract void	<code>mlUnregisterKeyEventListener</code> (<code>VncMirrorLinkKeyListener</code> listener) Unregisters a <code>MirrorLink</code> key event listener from this server.
abstract void	<code>provideLicenseFeature</code> (<code>int</code> featureId) Marks the server as providing the specified license feature.
abstract <code>VncExtension</code>	<code>registerExtension</code> (<code>java.lang.String</code> name, <code>VncExtensionListener</code> handler) Registers a new extension with the SDK.
abstract void	<code>reset</code> () Reset the server core back to a disconnected state.
abstract void	<code>reset</code> (<code>boolean</code> waitForFlush) Reset the server core back to a disconnected state.
abstract void	<code>sendExtensionMessage</code> (<code>VncExtension</code> extension, <code>byte[]</code> payload, <code>int</code> payloadOffset, <code>int</code> payloadLength) Send an extension message.
abstract void	<code>setAuthentication</code> (<code>int</code> authenticationType) Set the type of authentication to use.
abstract void	<code>setDesktopName</code> (<code>java.lang.String</code> name) Sets the VNC desktop name to the given string.
abstract void	<code>setEncryption</code> (<code>int</code> encryptionType) Set the type of encryption to use.
abstract void	<code>setH264Encoder</code> (<code>VncH264Encoder</code> h264Encoder, <code>boolean</code> debugModeEnabled) Register an H.264 encoder with the SDK.
abstract void	<code>setKey</code> (<code>byte[]</code> keyPair) Set key pair to be used for authentication and encryption.
abstract void	<code>setRemoteControlServicePackage</code> (<code>java.lang.String</code> packageName) Sets the package name to use to obtain the remote control service.
abstract void	<code>setUserData</code> (<code>java.lang.Object</code> userData) Set some user defined data to be associated with this server instance.

Methods inherited from class `java.lang.Object`

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

Field Detail

FEATURE_CLIPBOARD

`public static final int` `FEATURE_CLIPBOARD`

Whether or not support should be enabled for transferring clipboard information between the viewer and server.

See Also:

[Constant Field Values](#)

FEATURE_SEND_CLIPBOARD_ON_CONNECTION

`public static final int` `FEATURE_SEND_CLIPBOARD_ON_CONNECTION`

If the clipboard feature is enabled, then defines whether or not the server clipboard contents should be sent to the viewer when a connection is first established.

See Also:

[Constant Field Values](#)

FEATURE_WINCE_SET_DISPLAY_POLL_FREQUENCY

`public static final int` `FEATURE_WINCE_SET_DISPLAY_POLL_FREQUENCY`

On Windows mobile systems, sets a refresh rate for the server screen polling. This feature has no effect on android.

See Also:

[Constant Field Values](#)

FEATURE_VIEW_ONLY

`public static final int` `FEATURE_VIEW_ONLY`

On Windows mobile systems, tells the server to release all key/pointer presses and ignore further input. This feature has no effect on android.

See Also:

[Constant Field Values](#)

FEATURE_RFB4

```
public static final int FEATURE_RFB4
```

Enables or disables RFB4 connections. Enabled by default. Normally you would leave this option turned on, but on some occasions you might want to force connections to run using only RFB3. This is useful (for example) if you've got a MirrorLink viewer and MirrorLink client both supporting RFB4, but you want to force them to use RFB3. When this option is disabled, the very first stage of the RFB protocol negotiation will announce "RFB003.008" instead of "RFB004.001" so that there is no possibility of using RFB4.

See Also:

[Constant Field Values](#)

FEATURE_COMPARE_FB

```
public static final int FEATURE_COMPARE_FB
```

Enable or disable framebuffer comparison. Normally, this would default to true, and the microserver would thus not send any rectangle identical to the last one sent. Use `vncServer.enableFeature(FEATURE_COMPARE_FB, false)` if you wish to disable it; note that this isn't normally helpful!

See Also:

[Constant Field Values](#)

FEATURE_START_IN_LANDSCAPE

```
public static final int FEATURE_START_IN_LANDSCAPE
```

Requests that the initial server configuration is reported as landscape.

This is required for MirrorLink servers from version 1.1.6.

Enabling this feature must be done before a connection is attempted otherwise it will have no effect.

See Also:

[Constant Field Values](#)

FEATURE_MIRRORLINK_FORBID_PORTRAIT_ORIENTATION

```
public static final int FEATURE_MIRRORLINK_FORBID_PORTRAIT_ORIENTATION
```

Whether or not to block frame buffer updates for MirrorLink connection when display orientation is portrait. Disabled by default. This feature is only valid for MirrorLink connection and will be ignored if the connection is not MirrorLink. Enabling/disabling this feature must be done before a connection is established. Enabling/disabling feature will be ignored if the connection has been established.

See Also:

[Constant Field Values](#)

Constructor Detail

VncServer

```
protected VncServer()
```

Do not use the constructor. Call `create(android.content.Context, com.realvnc.vncserver.android.VncServerListener)` instead.

Method Detail

create

```
public static VncServer create(android.content.Context ctx,
                               VncServerListener listener)
```

Construct a new VNC server.

Will throw `java.lang.UnsatisfiedLinkError` if the native library provided with the SDK isn't available.

Parameters:

`ctx` - Android context for the server to use. Normally you should create and control the server from an Android service; this parameter should be your service object.

`listener` - provided by caller so that the server can notify it of significant events.

Returns:

The new `VncServer` object.

create

```
public static VncServer create(android.content.Context ctx,
                               VncServerListener listener,
                               android.os.Handler handler)
```

Construct a new VNC server. This variant allows you to provide your own `android.os.Handler` object, which is useful if you want to receive callbacks from the VNC server in a separate thread.

Will throw `java.lang.UnsatisfiedLinkError` if the native library provided with the SDK isn't available.

Parameters:

`ctx` - Android context for the server to use. Normally you should create and control the server from an Android service; this parameter should be your service object.

`listener` - provided by caller so that the server can notify it of significant events.

`handler` - The `android.os.Handler` object to use for posting callbacks.

Returns:

The new `VncServer` object.

create

```
public static VncServer create(android.content.Context ctx,
                               VncServerOrientationListener listener)
```

Construct a new VNC server.

Will throw `java.lang.UnsatisfiedLinkError` if the native library provided with the SDK isn't available.

Parameters:

`ctx` - Android context for the server to use. Normally you should create and control the server from an Android service; this parameter should be your service object.

`listener` - provided by caller so that the server can notify it of significant events.

Returns:

The new `VncServer` object.

create

```
public static VncServer create(android.content.Context ctx,
                               VncServerOrientationListener listener,
                               android.os.Handler handler)
```

Construct a new VNC server. This variant allows you to provide your own `android.os.Handler` object, which is useful if you want to receive callbacks from the VNC server in a separate thread.

Will throw `java.lang.UnsatisfiedLinkError` if the native library provided with the SDK isn't available.

Parameters:

`ctx` - Android context for the server to use. Normally you should create and control the server from an Android service; this parameter should be your service object.

`listener` - provided by caller so that the server can notify it of significant events.

`handler` - The `android.os.Handler` object to use for posting callbacks.

Returns:

The new `VncServer` object.

create

```
public static VncServer create(android.content.Context ctx,
                               VncServerOrientationListener listener,
                               VncServerMirrorLinkListener mListener)
```

Construct a new VNC server.

Will throw `java.lang.UnsatisfiedLinkError` if the native library provided with the SDK isn't available.

This method has been deprecated. Please use the version which receives a `MirrorLinkCallbackHandler` instance instead.

Parameters:

`ctx` - Android context for the server to use. Normally you should create and control the server from an Android service; this parameter should be your service object.

`listener` - provided by caller so that the server can notify it of significant events.

Returns:

The new `VncServer` object.

See Also:

`create(android.content.Context, MirrorLinkCallbackHandler)`

create

```
public static VncServer create(android.content.Context ctx,
                               VncServerOrientationListener listener,
                               VncServerMirrorLinkListener mListener,
                               android.os.Handler handler)
```

Construct a new VNC server.

This variant allows you to provide your own `android.os.Handler` object, which is useful if you want to receive callbacks from the VNC server in a separate thread.

Will throw `java.lang.UnsatisfiedLinkError` if the native library provided with the SDK isn't available.

This method has been deprecated. Please use the version which receives a `MirrorLinkCallbackHandler` instance instead.

Parameters:

`ctx` - Android context for the server to use. Normally you should create and control the server from an Android service; this parameter should be your service object.

`listener` - provided by caller so that the server can notify it of significant events.

`handler` - The `android.os.Handler` object to use for posting callbacks.

Returns:

The new `VncServer` object.

See Also:

`create(android.content.Context, MirrorLinkCallbackHandler, android.os.Handler)`

create

```
public static VncServer create(android.content.Context ctx,
                               MirrorLinkCallbackHandler mListener)
```

Construct a new VNC server.

Will throw `java.lang.UnsatisfiedLinkError` if the native library provided with the SDK isn't available.

Parameters:

- `ctx` - Android context for the server to use. Normally you should create and control the server from an Android service; this parameter should be your service object.
- `mListener` - provided by caller so that the server can notify it of significant events.

Returns:

The new `VncServer` object.

create

```
public static VncServer create(android.content.Context ctx,
                               MirrorLinkCallbackHandler mListener,
                               android.os.Handler handler)
```

Construct a new VNC server.

This variant allows you to provide your own `android.os.Handler` object, which is useful if you want to receive callbacks from the VNC server in a separate thread.

Will throw `java.lang.UnsatisfiedLinkError` if the native library provided with the SDK isn't available.

Parameters:

- `ctx` - Android context for the server to use. Normally you should create and control the server from an Android service; this parameter should be your service object.
- `mListener` - provided by caller so that the server can notify it of significant events.
- `handler` - The `android.os.Handler` object to use for posting callbacks.

Returns:

The new `VncServer` object.

create

```
public static VncServer create(android.content.Context ctx,
                               VncServerCallbackHandler listener)
```

Construct a new VNC server.

Will throw `java.lang.UnsatisfiedLinkError` if the native library provided with the SDK isn't available.

Parameters:

- `ctx` - Android context for the server to use. Normally you should create and control the server from an Android service; this parameter should be your service object.
- `listener` - provided by caller so that the server can notify it of significant events.

Returns:

The new `VncServer` object.

create

```
public static VncServer create(android.content.Context ctx,
                               VncServerCallbackHandler listener,
                               android.os.Handler handler)
```

Construct a new VNC server. This variant allows you to provide your own `android.os.Handler` object, which is useful if you want to receive callbacks from the VNC server in a separate thread.

Will throw `java.lang.UnsatisfiedLinkError` if the native library provided with the SDK isn't available.

Parameters:

- `ctx` - Android context for the server to use. Normally you should create and control the server from an Android service; this parameter should be your service object.
- `listener` - provided by caller so that the server can notify it of significant events.
- `handler` - The `android.os.Handler` object to use for posting callbacks.

Returns:

The new `VncServer` object.

setH264Encoder

```
public abstract void setH264Encoder(VncH264Encoder h264Encoder,
                                     boolean debugModeEnabled)
    throws VncException
```

Register an H.264 encoder with the SDK.

The H.264 support is only available on Android 5.0 or higher, and OpenGL ES Version 2.0 or higher. This API will throw `VncException` if it is called on a device which Android version is lower than 5.0 or OpenGL ES version is lower than 2.0. If this API is never called, the H.264 support is disabled by default.

The H.264 encoder must be set while the server is in the disconnected state. The API will throw `VncException` if an H.264 encoder is attempted to be set while the server is not in the disconnected state.

The H.264 encoder must be an instance of implementation of `VncH264Encoder` or null. If an H.264 encoder has been set before, then it will be replaced by the new H.264 encoder. Setting a null H.264 encoder will disable the H.264 support.

Parameters:

- `h264Encoder` - An instance of implementation of `VncH264Encoder` or null.

debugModeEnabled - If this is true, then the data provided by the encoder will be checked to ensure that the correct NAL units are present. This may have a negative impact on performance.

Throws:

VncException - If the H.264 encoder set fails.

checkRemoteControlAvailable

```
public abstract void checkRemoteControlAvailable()
                    throws VncException
```

Check whether the device has support for remote control.

The server will respond by calling the remoteControlAvailableCb callback with a parameter if 0 if support is available, or an appropriate error code if not.

This function only checks if the device has support for remote control, it does not check that the application calling this has permissions to use remote control. Permissions to use remote control will be checked when a connection is attempted.

Throws:

VncException - If called while the server is not in the "disconnected" state.

getRemoteControlInfo

```
public abstract java.util.List<VncRemoteControlInfo> getRemoteControlInfo()
                                                    throws VncException
```

Gets details of the device's remote control support.

This method can only be called after the remoteControlAvailableCb callback has been called. Thus, checkRemoteControlAvailable() should be called first.

The List will be empty if no remote control support is available. Otherwise, the first item in the list represents the form of remote control which will be used.

Returns:

A read-only list of objects representing the available remote control methods; potentially null before remoteControlAvailableCb has occurred

Throws:

VncException - If called while the server is not in the "disconnected" state.

blacklistRemoteControl

```
public abstract void blacklistRemoteControl(VncRemoteControlInfo remoteControl,
                                           boolean blacklist)
                                           throws VncException
```

Prevents a particular type of remote control method from being used, or reinstates it.

Parameters:

remoteControl - One of the objects representing a remote control method returned by getRemoteControlInfo(). Must not be null.

blacklist - If true, this remote control method will not be used; if false, it may be attempted.

Throws:

java.lang.NullPointerException - If remoteControl is null.

java.lang.IllegalArgumentException - If remoteControl was not obtained from getRemoteControlInfo().

VncException - If called while the server is not in the "disconnected" state.

addBearer

```
public abstract void addBearer(java.lang.String bearerName,
                              java.lang.String className)
```

Register a new bearer with the SDK. If a bearer is already registered with the given name then it will be replaced with the new details.

Parameters:

bearerName - short name for the bearer - this is the name that will be the name present in the command string used to look up the bearer to handle the command

className - fully qualified name of class that implements the VncBearer interface

addBearer

```
public abstract void addBearer(java.lang.String bearerName,
                              java.lang.Class<? extends VncBearer> bearerClass)
```

Register a new bearer with the SDK. If a bearer is already registered with the given name then it will be replaced with the new details.

Parameters:

bearerName - short name for the bearer - this is the name that will be the name present in the command string used to look up the bearer to handle the command

bearerClass - class that implements the VncBearer interface

getBearerInfo

```
public abstract java.util.Enumeration<VncBearerInfo> getBearerInfo()
```

Return an enumeration of objects implementing the VncBearerInfo interface. The returned objects will describe all of the bearers that are currently known to the SDK, including those that are automatically provided for backwards compatibility with the existing APIs such as the TCP in and TCP out bearers.

enableFeature

```
public abstract void enableFeature(int feature,
                                   boolean enable)
```

Selectively enable or disable optional features.

Parameters:

`feature` - one of the `FEATURE_XXX` constants specifying the optional feature to be enabled or disabled

`enable` - true to enable feature, otherwise false

setUserData

```
public abstract void setUserData(java.lang.Object userData)
```

Set some user defined data to be associated with this server instance.

Parameters:

`userData` - the user data to be associated with this server

See Also:

`getUserData()`

getUserData

```
public abstract java.lang.Object getUserData()
```

Retrieves the user defined data previously associated with this server.

Returns:

The user defined data or null if none set

See Also:

`setUserData(Object)`

getState

```
public abstract int getState()
```

Return the current state of the VNC server. See `VncServerState` for the definitions of the returned values.

isConnectionTearingDown

```
public abstract boolean isConnectionTearingDown()
```

Return true if the tearing down is in progress after either the existing connection is stopped to be processed or the VNC server instance stops listening for new connection. Otherwise, return false.

An existing connection can be stopped either by the fundamental bearer disconnection or errors. The disconnection can be initiated by the server or by the viewer.

Once tearing down starts, this call returns true until the callback `disconnectedCb` is completed.

generateKey

```
public abstract void generateKey(int keySize)
                               throws VncException
```

Instruct the server to start generation of a new RSA encryption key. When the key has been generated `keygenCb` will be invoked to return the key to the application which can then (optionally) store the key for future use. Note that this call will just start generation of the key, it will not automatically cause the server to start using the key. To do this the application must call `setKey`.

Parameters:

`keySize` - the size of the RSA key to be generated in bits

Throws:

`VncException` - When server not disconnected or expecting a key.

setKey

```
public abstract void setKey(byte[] keyPair)
                          throws VncException
```

Set key pair to be used for authentication and encryption.

Parameters:

`keyPair` - the key pair to be used for authentication and encryption. This object must be a key pair instance as returned by `keygenCb`. It is safe to persist and restore instances of this object between calls.

Throws:

`VncException` - When server not disconnected or expecting a key.

getServerSignature

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

Returns the signature for the server's RSA key. Assumes that `setKey()` has been called previously and succeeded, otherwise returns null.

setEncryption

```
public abstract void setEncryption(int encryptionType)
                               throws VncException
```


Set the type of encryption to use. Server defaults to VNC_ENCRYPTION_NONE.

Parameters:

encryptionType - One of the VNC_ENCRYPTION_* constants (see VncEncryptionType)

Throws:

VncException - if the security type is not recognised.

getEncryption

```
public abstract int getEncryption()
```

Get the currently selected encryption type. Server defaults to VNC_ENCRYPTION_NONE.

Returns:

One of the VNC_ENCRYPTION_* constants.

setAuthentication

```
public abstract void setAuthentication(int authenticationType)
                                throws VncException
```

Set the type of authentication to use. Server defaults to VNC_AUTH_NONE.

Parameters:

authenticationType - One of the VNC_AUTH_* constants (see VncAuthType)

Throws:

VncException - if the security type is not recognised.

getAuthentication

```
public abstract int getAuthentication()
```

Get the currently selected authentication type. Server defaults to VNC_AUTH_NONE.

Returns:

One of the VNC_AUTH_* constants.

authenticate

```
public abstract void authenticate(boolean ok)
                                throws VncException
```

Accept (or reject) a connection based on authentication credentials supplied to authCb.

Parameters:

ok - true to accept the connection, false to reject

Throws:

VncException - When server is not in an authenticating state.

listen

```
public abstract void listen(int port)
                                throws VncException
```

Starts this VNC server instance listening for an incoming connection using the VNC TCP/IP inbound pluggable bearer. `listeningCb` will be issued once server has begun to listen. `connectedCb` will be called once a connection has been established, after which `accept` should be called to accept the connection.

This is equivalent to calling `connect(VncCommandString)` with a command string of `vnccmd:v=1;t=C;p=port`, and is provided for backwards compatibility with the pre-pluggable bearer SDK.

Parameters:

port - port on which to listen for a viewer connection

Throws:

VncException - When server is not in a disconnected state.

connect

```
public abstract void connect(java.lang.String address,
                             int port)
                                throws VncException
```

Establish a socket connection to a listening viewer using the VNC TCP/IP outbound pluggable bearer. `connectedCb` will be called once a connection has been established, after which `accept` should be called to accept the connection.

This is equivalent to calling `connect(VncCommandString)` with a command string of `vnccmd:v=1;t=C;a=address;p=port`, and is provided for backwards compatibility with the pre-pluggable bearer SDK.

Parameters:

address - listening viewer to connect to. May be either a host name or an IP address

port - number of port on which the viewer is listening

Throws:

VncException - If this method is called while the server is not in a disconnected state.

connect

```
public abstract void connect(VncCommandString commandString)
                        throws VncException
```

Establish a connection using the connection details and bearer specified in the command string. If the "t" field of the command string is either "L", "C", or "D" then the VNC TCP/IP inbound, outbound, or data relay bearers respectively will be used to establish the connection in the same way as if `listen` or `connect(String, int)` had been called. Any other value of the "t" field will be looked up as a name of a pluggable bearer, and if found that will be used to establish the connection. If a bearer could not be found then the `errorCb` method will be invoked.

Custom pluggable bearers can be added to the SDK using the `addBearer(String, String)` method, and information on the bearers known to the SDK can be found using `getBearerInfo()`.

`connectedCb` will be called once a connection has been established, after which `accept` should be called to accept the connection.

Parameters:

`commandString` - containing the details of the connection to be established

Throws:

`VncException` - If this method is called while the server is not in a disconnected state.

acceptRemoteKey

```
public abstract void acceptRemoteKey(boolean ok)
                        throws VncException
```

Accept or reject an RSA key from a VNC viewer. Should be called by the application after receiving `remoteKeyCb`.

Parameters:

`ok` - true to accept the key, false to reject

Throws:

`VncException` - If this method is called while the server is not waiting for a response.

accept

```
public abstract void accept(boolean ok)
                        throws VncException
```

Accept or reject a connection from a VNC viewer. Should be called by the application after receiving `connectedCb`.

Parameters:

`ok` - true to accept connection, false to reject the connection

Throws:

`VncException` - If this method is called while the server is not in an accepting state.

reset

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

Reset the server core back to a disconnected state. If the server is listening for a connection then it will be stopped from listening, and if the server is connected then the connection will be closed. If the server is already disconnected when called then this method will have no effect.

This will wait for any pending data to be written over a connection before continuing with the disconnection.

reset

```
public abstract void reset(boolean waitForFlush)
```

Reset the server core back to a disconnected state. If the server is listening for a connection then it will be stopped from listening, and if the server is connected then the connection will be closed. If the server is already disconnected when called then this method will have no effect.

The `waitForFlush` parameter allows the caller to specify if the reset should wait for any pending data to be written to a connected viewer before disconnecting.

If the caller knows for certain that the connection in use has been broken, such as the network interface for a tethered USB connection going down, then this should be called with `waitForFlush` set to false.

Parameters:

`waitForFlush` - true if any pending data should be written before closing the connection, false if any pending data should just be discarded.

destroy

```
public abstract void destroy()
```

Destroy the the server object and release all associated resources. Once the server has been destroyed it can no longer be used and any subsequent method calls will cause an `VncException` to be raised. Should be used when the application has finished with the server object and doesn't want to wait for the garbage collector to reclaim the server resources. No callbacks will occur after the server is destroyed.

bell

```
public abstract void bell()
                        throws VncException
```

Sends a bell message to the client.

Throws:

`VncException` - If this method is called while the server is not in an connected state.

login

```
public abstract void login(java.lang.String username,
```

```
java.lang.String password)
    throws VncException
```

Provide user name and/or password to viewer during reverse authentication.

Parameters:

password - to be supplied to viewer

Throws:

VncException - If this method is called while the server is not in an authenticating state.

See Also:

loginCb

freeze

```
public abstract void freeze(boolean freeze)
    throws VncException
```

Freeze or thaw the server. When in the frozen state the server will not send any display updates until it is thawed. Freezes and thaws can be nested; if you freeze the core twice you will have to thaw it twice before it will restart.

Parameters:

freeze - True to freeze, false to thaw

Throws:

VncException - if the server is not connected.

getVersionString

```
public abstract java.lang.String getVersionString()
```

Get the VNC server version string in form "major.minor.patch.revision number"

setDesktopName

```
public abstract void setDesktopName(java.lang.String name)
```

Sets the VNC desktop name to the given string.

Parameters:

name - Desktop name to send to the viewer

registerExtension

```
public abstract VncExtension registerExtension(java.lang.String name,
    VncExtensionListener handler)
    throws VncException
```

Registers a new extension with the SDK.

Valid extension names must be between 1 and 255 bytes in size, must contain at least one '.', and may only contain the characters 'a'-'z', 'A'-'Z', '0'-'9', '_' and '.'.

Extension names should follow the reverse-domain-name naming convention for Java packages, see http://en.wikipedia.org/wiki/Java_package#Package_naming_conventions for further information.

Parameters:

name - Unique identifier for this protocol extension

handler - VncExtensionListener object which will be invoked when messages arrive for this extension.

Returns:

Handle for this extension. Pass to `sendExtensionMessage(com.realvnc.vncserver.android.VncExtension, byte[], int, int)` to send a message.

Throws:

VncException - for error conditions (server not idle, invalid name, extension already registered)

sendExtensionMessage

```
public abstract void sendExtensionMessage(VncExtension extension,
    byte[] payload,
    int payloadOffset,
    int payloadLength)
    throws VncException
```

Send an extension message.

Requests to send an arbitrary message for a previously registered extension.

There is no confirmation when the message is sent. If this is required then extensions will need to implement acknowledgement of receiving messages in their extension protocol.

Extension messages are only guaranteed to be received in the same order as they are sent within a particular extension. The order of messages sent relative to other extensions' messages are not guaranteed to be correct.

That is to say that sending message 1 over extension 1 followed by message 2 over extension 1 is guaranteed to be received in that order. However sending message 1 over extension 1 followed by message 1 over extension 2 is not guaranteed to be received in that order.

As extension messages are sent over the same connection as the other RFB data, care should be taken to not send too many extension messages in quick succession. If too many extension messages are sent in a short period of time then the update frequency and response time of the remote framebuffer will be impacted. In extreme cases this can lead to the remote display not being updated until all outstanding extension messages have been sent.

Reliability

A successful return from `sendExtensionMessage()` is a guarantee that the message has been queued for sending, but this is *not* a guarantee of successful delivery. This is because the session may be lost before the message is received by the peer.

Applications that require a guarantee of successful delivery must implement this themselves, e.g. by using acknowledgements.

Update starvation

Extension messages sent from server to viewer are competing for bandwidth with framebuffer updates. This means that, if an application sends a large volume of extension message data from server to viewer, the server will have less bandwidth available for updates, and the framerate will decrease.

The converse is also true; if the server display updates rapidly, then the resulting framebuffer updates will increase the time it takes to send extension messages to the viewer.

If your server application has a large volume of extension message data to send to the viewer, then you may wish to limit the rate at which extension messages are sent, so that the impact on framebuffer updates is lessened.

Alternatively, if you wish to complete the extension message data transfer as quickly as possible, then you may wish to disable updates entirely until the transfer is complete, and display some alternative viewer-side UI instead (such as a progress bar).

Handling VNCSERVER_ERR_RESOURCES

If an exception with error code `VNCSERVER_ERR_RESOURCES` occurs then this indicates that the server does not have sufficient memory available to send the required extension message. Before retrying to send the extension message some more system memory needs to be made available.

Handling VNCSERVER_ERR_INSUFFICIENT_BUFFER_SPACE

If an exception with error code `VNCSERVER_ERR_INSUFFICIENT_BUFFER_SPACE` is thrown then this indicates that the SDK cannot queue an extension message for sending due to a temporary resource shortage. In this case, the application can make another attempt to queue the message at a later time.

Two possible strategies for avoiding this condition, and handling it should it arise, are:

- Measure the average rate at which extension message data can successfully be queued in your deployment. Queue extension messages at or just below that rate, and, if an attempt to queue a message fails, retry after a short timeout.
- Require the application receiving the extension messages to acknowledge each message. Whenever you receive an acknowledgement for one message, attempt (or re-attempt) to queue the next one. (You may wish to use a sliding window for acknowledgements, as in TCP.)

Parameters:

`extension` - Extension handle as returned by `registerExtension()`.
`payload` - Byte array containing the data to send
`payloadOffset` - Offset of the message within the array
`payloadLength` - Length of the message in bytes

Throws:

`VncException` - for error conditions (server not connected, bad parameter, out of memory)

injectKeyEvent

```
public abstract void injectKeyEvent(android.view.KeyEvent ev)
```

Inject a keyboard event into the system.

This API does not indicate if the key event has been injected successfully.

This API will not inject a key if a connection to a VNC viewer is not currently established or if the server is unable to inject key event.

This API is provided to allow the VNC Sample Server application to inject particular key events. Use of this API in any other way is not guaranteed to be functional and is untested.

Parameters:

`ev` - Event to inject

addLicense

```
public abstract byte[] addLicense(java.lang.String licenseText)
                           throws VncLicenseNotValidException,
                                   VncException
```

Adds a license to the server.

This method may be called as many times as is necessary. All the license features and checks will be combined together.

On successful return the information contained in the license will be added to the server.

This method should only be called when the server is idle.

Parameters:

`licenseText` - The text of the license to add to the server

Returns:

The serial number for the licenses which has been added

Throws:

`VncLicenseNotValidException` - for invalid licenses
`VncException` - for error conditions (invalid state, bad parameters, out of memory)

addLicenseFeature

```
public abstract void addLicenseFeature(int featureId,
                                       byte[] featureKey)
                                       throws VncException
```

Adds knowledge of a feature to the server.

A feature must be added to the server before it can be used in a remote feature check on the viewer or be provided to the viewer as part of the viewers remote feature check on the server.

It is not necessary to add features that have already been defined by a license added by a call to `addLicense(java.lang.String)`. This method allows features to be added without creating a license for the feature.

Multiple calls to this method with the same feature ID will use the feature key provided in the most recent call.

Attempting to add license features from the RealVNC feature range (below 0xfffff) will always fail.

This method should only be called when the server is idle.

Parameters:

`featureId` - The ID of the feature being added

`featureKey` - The secret key for the feature, this should be exactly 16 bytes long

Throws:

`VncException` - for error conditions (invalid state, bad parameters, out of memory)

provideLicenseFeature

```
public abstract void provideLicenseFeature(int featureId)
                                   throws VncException
```

Marks the server as providing the specified license feature.

Calling this method will add a feature to the list of provided features. This will allow the server to respond to remote feature check challenges from the viewer with the specified feature. If a feature is not marked as provided then the server will not offer it in the response to a remote feature check from a viewer, even if the feature has been added by a call to `addLicenseFeature(int, byte[])`.

Attempting to provide license features from the RealVNC feature range (below 0xfffff) will always fail.

Any feature IDs used in this feature check should have been defined as features by calls to `addLicenseFeature(int, byte[])`.

This method should only be called when the server is idle.

Parameters:

`featureId` - The ID of the feature the server should provide.

Throws:

`VncException` - for error conditions (invalid state, bad parameters, out of memory)

addRemoteFeatureCheck

```
public abstract int addRemoteFeatureCheck(int[] featureIds,
                                   VncRemoteFeatureCheckListener listener)
                                   throws VncException
```

Adds a remote feature check to this server instance.

When connecting to a viewer, the viewer will be challenged with the provided feature check. The result of this challenge to the viewer will be provided to the listener provided to this callback.

Any feature IDs used in this feature check should have been defined as features by either licenses added via `addLicense(java.lang.String)` or by calls to `addLicenseFeature(int, byte[])`.

The VNC viewer will be challenged with the list of featureIds provided. The viewer will then pick one of the feature IDs and respond with to the challenge indicating success (and the feature which was successful) or respond indicating failure.

This behaviour means that a list of two features is challenging the viewer to provide either the first or the second feature. If it's required for the viewer to have both features then two separate calls should be made to this method, each with a feature ID list containing one element.

This method should only be called when the server is idle.

Parameters:

`featureIds` - A list of feature IDs to challenge the viewer with when performing the remote feature check.

`listener` - `VncRemoteFeatureCheckListener` object which will be invoked when the remote feature checks are performed.

Returns:

An unique identifier which will be passed to the listener callbacks

Throws:

`VncException` - for error conditions (invalid state, bad parameters, out of memory)

localFeatureCheck

```
public abstract boolean localFeatureCheck(int[] featureIds)
                                   throws VncException
```

Performs a local feature check.

The result is true if any of the feature IDs supplied are provided by a license previously added locally to this server instance using `addLicense()`.

Note that user features added using `provideLicenseFeature`, and features provided by a connected viewer, are not considered when performing local feature checks.

Parameters:

`featureIds` - The array of feature IDs to check.

Returns:

true if feature check passed, false if failed.

Throws:

`VncException` - for error conditions (invalid state, bad parameters, out of memory)

mlSendServerDisplayConfiguration

```
public abstract void mlSendServerDisplayConfiguration(VNCServerDisplayConfiguration sdc)
    throws VncException
```

MirrorLink: Send a Server Device Configuration message. This *must* only be called, a single time, after the receipt of a `VncServerMirrorLinkListener.mlConnectionReceivedCb(com.realvnc.mirrorlink.VNCServerDisplayConfiguration)` callback.

If the MirrorLink Version in the `VNCServerDisplayConfiguration` is set to 1.3 then additional restrictions are applied. In particular, fields deprecated in MirrorLink 1.3 must be set to the values given in the specification. These values are outlined in the documentation for `VNCServerDisplayConfiguration`.

Parameters:

`sdc` - The server display configuration message to send.

Throws:

`VncException` - for error conditions (invalid state, bad parameters, out of memory)

mlSendServerEventConfiguration

```
public abstract void mlSendServerEventConfiguration(VNCServerEventConfiguration sec)
    throws VncException
```

MirrorLink: Send a Server Event Configuration message. This *must* only be called, a single time, after the receipt of a `VncServerMirrorLinkListener.mlClientDisplayConfigurationReceivedCb(com.realvnc.mirrorlink.VNCClientDisplayConfiguration, com.realvnc.mirrorlink.VNCServerEventConfiguration)` callback. See the documentation relating to that callback for discussion of what parameters you might choose to pass into this function.

You should give due consideration to the key events you believe that your implementation supports, and ensure those values are suitable in the `VNCServerEventConfiguration` which you pass into this API.

If support for additional key events are added then it is recommended that a `VncMirrorLinkKeyEventListener` is registered by using `mlRegisterKeyEventListener(VncMirrorLinkKeyEventListener)`. This allows a user of this SDK to implement the correct response to the additional key events that have been added over the default. Failure to do this will lead to the server SDK ignoring the additional key events or otherwise handling them incorrectly.

If the MirrorLink Version in the `VNCServerDisplayConfiguration` was set to 1.3 or above then additional restrictions are applied. In particular, fields deprecated in MirrorLink 1.3 must be set to the values given in the specification. Additionally, support for certain device keys and knob keys is required. These values are outlined in the documentation for `VNCServerEventConfiguration`.

Parameters:

`sec` - The server event configuration message to send.

Throws:

`VncException` - for error conditions (invalid state, bad parameters, out of memory, deprecated field used)

mlRequestSendDeviceStatus

```
public abstract void mlRequestSendDeviceStatus()
    throws VncException
```

MirrorLink: Inform the server that a new device status request should be sent. This may be called at any time when you have a running MirrorLink VNC connection. It will always result in a callback to `VncServerMirrorLinkListener.mlDeviceStatusSendNeededCb(VNCDeviceStatus latestRequest, VNCDeviceStatus defaultReply)`.

If you wish to send a MirrorLink DeviceStatus packet to the client, you should call this function, *not* `mlSendDeviceStatus(VNCDeviceStatus deviceStatus)`. The latter function is to be called *only* in response to a `VncServerMirrorLinkListener.mlDeviceStatusSendNeededCb(VNCDeviceStatus latestRequest, VNCDeviceStatus defaultReply)` callback.

Note: This call used to be mandatory in previous versions (before 3.0) of the SDK if the Server is to send a new Device Status Response to the Client. Now the Server can send the Device Status without needing to call this, but the call is kept for compatibility reasons.

Throws:

`VncException` - for error conditions (invalid state, bad parameters, out of memory)

mlSendDeviceStatus

```
public abstract void mlSendDeviceStatus(VNCDeviceStatus deviceStatus)
    throws VncException
```

MirrorLink: send a 'device status' message to the viewer, in response to a `VncServerMirrorLinkListener.mlDeviceStatusSendNeededCb(VNCDeviceStatus latestRequest, VNCDeviceStatus defaultReply)` callback.

This *must not* be called unless you have received such a callback. If you wish to send a device status message to the client at any other time, do not use this API. Instead, use `mlRequestSendDeviceStatus()`.

Certain fields in the `VNCDeviceStatus` have been deprecated in MirrorLink 1.3. For details, see the documentation for `VNCDeviceStatus`. Setting any deprecated values in deviceStatus when calling this method will result in an exception with error code `VncServerCoreErrors.VNCSEVER_ERR_DEPRECATED_FIELD_USED`.

Parameters:

`deviceStatus` - The new device status

Throws:

`VncException` - for error conditions (invalid state, bad parameters, out of memory)

mlSetContextInformation

```
public abstract void mlSetContextInformation(java.util.List<android.util.Pair<android.graphics.Rect, VNCContextInformation>> contextInformation)
    throws VncException
```

Tell the VNC Server SDK context about areas of the screen.

This information always replaces all prior context information, and should therefore be a complete set of information about all visible parts of the screen.

The coordinates used to express the visible parts of the screen must be expressed relative to the physical device screen size, regardless of any server-side scaling settings. The SDK will make sure that coordinates are translated appropriately when transmitted to the MirrorLink viewer

This method will not cause Context Information to be sent immediately; instead the context information will accompany the next framebuffer update. It *may* schedule a new framebuffer update, but it may not.

The provided information is passed on as-is to the viewer, so care should be taken to ensure that the information provided is compliant with the provisions of the MirrorLink specification. There are two exceptions:

- If any one individual rectangle is null, or has dimensions 0,0,0,0, it will be replaced with a rectangle covering the whole screen.
- If the complete provided information does not cover the entire screen area, the VNC viewer SDK will add an extra rectangle with values of '0' to cover the whole screen. This will be the back-most rectangle, so any rectangles provided to this function will override that.

The server application must call this function *before* the new application becomes visible on the screen, or old applications might become visible briefly on the screen of the MirrorLink viewer.

The list of context information you provide **must not** contain more than 16 entries. This is a hard-coded limit to avoid the need to allocate RAM at runtime inside the server core code, which can cause problems in low-resource situations.

Note that the `VNCContextInformation.contentRulesFollowed` is deprecated in MirrorLink 1.3. MirrorLink 1.3 servers must set this field to 0 or an exception with error code `VncServerCoreErrors.VNCSEVER_ERR_DEPRECATED_FIELD_USED` will be thrown.

Parameters:

`contextInformation` - Information about the screen, or null if no information is known

Throws:

`VncException` - for error conditions (invalid state, bad parameters, out of memory)

mlSetContextInformationAndInvalidate

```
public
abstract void mlSetContextInformationAndInvalidate(java.util.List<android.util.Pair<android.graphics.Rect,VNCContextInformation>> contextInformation)
throws VncException
```

Tell the VNC Server SDK context about areas of the screen.

This is similar to `mlSetContextInformation(List)`, but will invalidate the framebuffer after setting the context information, causing a new framebuffer update to be sent.

Parameters:

`contextInformation` - Information about the screen, or null if no information is known

Throws:

`VncException` - for error conditions (invalid state, bad parameters, out of memory)

mlSetEventMapping

```
public abstract void mlSetEventMapping(java.util.Map<java.lang.Integer,java.lang.Integer> entries)
throws VncException
```

Sets event mapping entries.

This takes a mapping from client-side keycodes to server-side keycodes. Some of these key codes can be found within `VNCMirrorLinkKeys`. You may only specify mappings of 'high' client key codes - the MirrorLink 1.0 specification says "The Event Mapping and Event Mapping Request message pair provides the client with information about the server mapping of high key symbol values.". RealVNC interprets this to mean the knob keys, ITU keys, device keys, function keys and multimedia keys defined in appendix A of the MirrorLink 1.0 specification. Mapping of other types of key event is not allowed. Mapping of events in the 'reserved' ranges in that appendix is also not allowed.

The set of entries provided totally replaces all prior mappings known to the server SDK. It's therefore recommended that you would normally get a copy of the existing mappings by using `mlGetEventMapping()`, and adding or changing entries in the resulting map, before passing the map into this API. (Note that the map returned by `mlGetEventMapping()` may not be mutable, so you will need to clone it).

Even if you're starting from scratch at the beginning of a MirrorLink connection, it's recommended you use the information provided by the `mlGetEventMapping()` API, since the server SDK makes an attempt at providing sufficient mappings to make it possible to control the device.

Providing an entry where the value is 0, is equivalent to simply omitting that mapping from the mapping altogether.

Passing a null object is equivalent to setting a mapping with 0 entries.

The actual policy which you should use for mapping keys is your responsibility. You must stick within the rules of the MirrorLink specification, yet somehow make it possible to control your device from just about any head unit with any combination of physical controls.

If the new mapping is different from the prior mapping then an `EventMapping` message will be sent if the supported version of the MirrorLink server is less than 1.3. If it is 1.3 or greater, no such message shall be sent as a direct result of this call.

You must not call this API before you receive `VncServerMirrorLinkListener.mlClientEventConfigurationReceivedCb(VNCClientEventConfiguration, Map)`.

Parameters:

`entries` - The entries to set

Throws:

`VncException` - for error conditions (invalid state, bad parameters, out of memory)

mlGetEventMapping

```
public abstract java.util.Map<java.lang.Integer,java.lang.Integer> mlGetEventMapping()
throws VncException
```

Returns the event mapping presently understood by the server.

You must not call this API before you receive `VncServerMirrorLinkListener.mlClientEventConfigurationReceivedCb(VNCClientEventConfiguration clientEventConfiguration, Map defaultMappingRecommendation)`.

Changes to the resulting map are ignored. It may not even be mutable. You should inform the server about desired changes by creating a copy and passing it to `mlSetEventMapping(Map entries)`.

Returns:

A map from client key symbol values to server key symbol values

Throws:

`VncException` - for error conditions (invalid state, out of memory)

mlSendEventMappingRequestReply

```
public abstract void mlSendEventMappingRequestReply(int clientKeyCode)
throws VncException
```

Replies to an event mapping request received from the MirrorLink viewer. You must call this whenever you receive a `VncServerMirrorLinkListener.mlEventMappingRequestReceivedCb(int, int)` callback.

This is necessary whenever the client has requested to update the event mapping. (Note that this is not called when the client merely queries the existing event mapping status).

You may or may not wish to grant the request of the client to update the mapping. If you wish to ignore the request, simply call this method. If you wish to grant the request, you should first update the event mapping as follows:

```
Map<Integer, Integer> newMapping = new HashMap<Integer, Integer>(vncServer.mlGetEventMapping());
newMapping.put(clientKeyCode, serverKeyCode);
vncServer.mlSetEventMapping(newMapping);
vncServer.mlSendEventMappingRequestReply(clientKeyCode);
```

You must not call this API at any point other than when you have received `VncServerMirrorLinkListener.mlEventMappingRequestReceivedCb(int, int)`. However, it is not guaranteed that you will receive any error code if this API is called at the wrong time - it's up to you to be careful and call it when such a request has been received.

Parameters:

`clientKeyCode` - The client key code

Throws:

`VncException` - for error conditions (invalid state, out of memory)

mlFrameBufferBlockingNotificationHandled

@Deprecated

```
public abstract void mlFrameBufferBlockingNotificationHandled()
                    throws VncException
```

Deprecated. Use `mlFrameBufferBlockingNotificationHandled(VncFrameBufferBlockingNotification)` instead.

Throws:

`VncException`

mlFrameBufferBlockingNotificationHandled

```
public abstract void mlFrameBufferBlockingNotificationHandled(VncFrameBufferBlockingNotification notification)
                    throws VncException
```

Indicates that processing of a framebuffer blocking notification message has completed.

Some types of framebuffer blocking notification message - especially, those which indicate an unsupported UI mode - require us to totally stop sending framebuffer updates until the situation has been resolved.

It's the responsibility of the application to handle the framebuffer blocking notification, if necessary by rotating the screen and/or arranging to switch to another application.

After it has done so (and the blocked state has been resolved) the application must call this API in order to indicate to the server SDK that framebuffer processing should recommence.

If one, or more, framebuffer blocking notifications are received this needs to be called once for each notification. Individual notifications are differentiated by their unique IDs.

Parameters:

`notification` - The blocking notification message

Throws:

`VncException` - for error conditions (invalid state, out of memory)

mlTriggerFrameBufferBlockingNotification

```
public abstract void mlTriggerFrameBufferBlockingNotification(VncFrameBufferBlockingNotification notification)
                    throws VncException
```

Triggers a server side `FrameBufferBlockingNotification`.

Whenever a server is able to predict a blocking condition expected from the client (when implementing its own drive safe home screen for example), it can use this method to trigger a regular framebuffer blocking notification to take care of the situation on behalf of the client (thus having the client undisturbed at all).

Server side blocking notification are automatically applied to the whole screen

Parameters:

`notification` - The blocking notification

Throws:

`VncException` - for error conditions (invalid state, out of memory)

getOrientationManager

```
public abstract VncOrientationManager getOrientationManager()
```

Retrieves the orientation manager instance for this server object.

Returns:

An object for controlling orientation functionality related to the server.

getContextInformationManager

```
public abstract VncContextInformationManager getContextInformationManager()
```

Retrieves the context information manager instance for this server object.

Returns:

An object for controlling context information functionality related to the server.

getDisplayInformationManager

```
public abstract VncDisplayInformationManager getDisplayInformationManager()
```

Retrieves the display information manager instance for this server object.

Returns:

An object that manages the display information for the server.

See Also:

[VncDisplayInformationManager](#)

mlRegisterKeyEventListener

```
public abstract void mlRegisterKeyEventListener(VncMirrorLinkKeyEventListener listener)
```

Registers a MirrorLink key event listener for this server.

When a MirrorLink key event, as described in Appendix B of CCC-TS-010 v1.1, v1.2 and v1.3, is received then it will be offered to the registered listeners.

Multiple listeners can be registered at once, they will be offered MirrorLink key events in the order that they were registered.

When registering a listener using this method then a server will probably want to modify the supported key events advertised to the client through `mlSendServerEventConfiguration(com.realvnc.mirrorlink.VNCServerEventConfiguration)`. The default server event configuration passed to `VncServerMirrorLinkListener.mlClientDisplayConfigurationReceivedCb(com.realvnc.mirrorlink.VNCClientDisplayConfiguration, com.realvnc.mirrorlink.VNCServerEventConfiguration)` is constructed based on the default functionality of the SDK.

Parameters:

`listener` - The listener to register for events.

See Also:

[VncMirrorLinkKeyEventListener.mlKeyEventReceived\(int, int\)](#)

mlUnregisterKeyEventListener

```
public abstract void mlUnregisterKeyEventListener(VncMirrorLinkKeyEventListener listener)
```

Unregisters a MirrorLink key event listener from this server.

Once this method returns the listener will no longer receive callbacks offering it MirrorLink key events.

Parameters:

`listener` - The listener to unregister for events.

setRemoteControlServicePackage

```
public abstract void setRemoteControlServicePackage(java.lang.String packageName)
```

Sets the package name to use to obtain the remote control service.

Call this API with a null package name to allow the server SDK to decide upon a remote control service from a suitable package. When making this decision the server SDK will only use packages which are signed with the platform keys.

If this value is set to a non-null value then the package will be used regardless of the signatures used to sign that package.

The default value for the package name is null.

Parameters:

`packageName` - The package name to use when resolving the remote control service.

customRemoteControlServiceRequest

```
public abstract android.os.Bundle customRemoteControlServiceRequest(java.lang.String name,
                                                                    android.os.Bundle payload)
                                                                    throws VncException
```

Sends a custom request to the remote control service.

Request names should be in standard Java reverse-DNS notation, for example "com.myandroidoem.EnablePhaseTractorBeams".

Remote control service implementations ignore any request which they don't understand, and return a null Bundle. If the service understands the request, a non-null Bundle is returned and its content is request-specific.

Requests which require to perform tasks that cannot be completed immediately should return their result through the `VncServerCallbackHandler.customRemoteControlServiceCb(java.lang.String, android.os.Bundle)` callback.

In general it's a way for the VNC remote control systems to make use of 'signature' level APIs and permissions, beyond the normal APIs required for basic remote control. Of course, as with basic remote control, all such APIs can only be used thanks to the assistance of the Android OEM.

See [CustomRemoteControlServiceRequests](#) for custom requests supported by remote control service implementations provided by RealVNC.

Parameters:

`name` - The request name; as above this is given in reverse-DNS notation.

`payload` - Any payload required for this custom message. This can be null; it's up to the custom message to interpret this however it likes.

Returns:

null if the remote control service could not understand the message; non-null if the service did understand the message. In which case the content of the Bundle will be whatever is returned by that custom bit of the RemoteControlService.

Throws:

[VncException](#) - If called while the RCS exited or is not available.

See Also:

CustomRemoteControlServiceRequests

getBluetoothAdapterAddress

```
public abstract java.lang.String getBluetoothAdapterAddress()
```

Gets the hardware address of the local Bluetooth adapter.

This method requires the permission 'android.permission.BLUETOOTH'.

Also, from Android 6.0 onwards, this method provides the real address only whilst remote control is enabled or your app uses the system level permission 'android.permission.LOCAL_MAC_ADDRESS'.

This is due to the restriction introduced in Android 6.0 by which the hardware address of the local Bluetooth adapter can be read only by system apps.

The returned address will default to "02:00:00:00:00:00" if the hardware address is not accessible.

Returns:

The hardware address of the local Bluetooth adapter, or null if Bluetooth is not supported on this hardware platform.

Throws:

`java.lang.SecurityException` - if 'android.permission.BLUETOOTH' is not granted.

com.realvnc.vncserver.android

Interface VncServerListener

All Known Subinterfaces:

VncServerMirrorLinkListener, VncServerOrientationListener

All Known Implementing Classes:

MirrorLinkCallbackHandler, VncServerCallbackHandler

public interface **VncServerListener**

Listener interface used by the VNC server to notify the application that certain events have occurred. All of the following callbacks will be made from the main system dispatch thread after all pending events have been processed. This means that although the `VncServer` object uses a number of threads internally you can be sure that the callbacks will be made from a single thread and so will not be made while your application is mid-call in the server.

Method Summary

Methods	
Modifier and Type	Method and Description
void	<code>authCb(VncServer vncServer, java.lang.String username, java.lang.String password)</code> Callback invoked when the viewer has provided some authentication details.
void	<code>connectedCb(VncServer vncServer, java.lang.String address)</code> Called when a viewer (or data relay) has been connected to us, or we have connected to a viewer.
void	<code>connectingCb(VncServer vncServer)</code> Callback to indicate that the VNC server is connecting to a remote viewer.
void	<code>disconnectedCb(VncServer vncServer)</code> Called when the viewer has disconnected from us, or we have disconnected from the viewer.
void	<code>errorCb(VncServer vncServer, int errorCode, java.lang.Exception e)</code> Called when an unexpected error occurred to indicate that the server has returned to the disconnected state and is no longer listening for a connection, connecting, or connected.
void	<code>keygenCb(VncServer vncServer, byte[] keyPair)</code> Called when RSA key pair generation has completed.
void	<code>listeningCb(VncServer vncServer, java.lang.String localAddress)</code> Callback to indicate that the VNC server is listening for an incoming connection.
void	<code>loginCb(VncServer vncServer, boolean usernameReq, boolean passwordReq)</code> Login callback issued during reverse authentication.
void	<code>remoteControlAvailableCb(VncServer vncServer, int errorCode)</code> Called in response to the application calling <code>checkRemoteControlAvailable()</code> .
void	<code>remoteKeyCb(VncServer vncServer, byte[] rsaKey, byte[] signature)</code> Remote Key notification callback.
void	<code>runningCb(VncServer vncServer)</code> Called when the VNC viewer and server are connected together and the VNC session has started.

Method Detail

keygenCb
<code>void keygenCb(VncServer vncServer, byte[] keyPair)</code>

Called when RSA key pair generation has completed. The application can then (optionally) store the key for future use. If the application wishes the server to start using the key then at some point in the future it must call `VncServer.setKey(byte[])`.

Parameters:

`vncServer` - VncServer instance issuing this callback
`keyPair` - an object representing the RSA key pair that was generated.

listeningCb

```
void listeningCb(VncServer vncServer,  
                java.lang.String localAddress)
```

Callback to indicate that the VNC server is listening for an incoming connection.

Parameters:

`vncServer` - VncServer instance issuing this callback
`localAddress` - of the listening server

connectingCb

```
void connectingCb(VncServer vncServer)
```

Callback to indicate that the VNC server is connecting to a remote viewer.

Parameters:

`vncServer` - VncServer instance issuing this callback

remoteKeyCb

```
void remoteKeyCb(VncServer vncServer,  
                 byte[] rsaKey,  
                 byte[] signature)
```

Remote Key notification callback. Provides the details of a remotely supplied key during authentication allowing the application to reject the key as untrusted via `VncServer.acceptRemoteKey`.

Parameters:

`vncServer` - VncServer instance issuing this callback
`rsaKey` - supplied by the viewer
`signature` - of the RSA key supplied by the viewer

authCb

```
void authCb(VncServer vncServer,  
            java.lang.String username,  
            java.lang.String password)
```

Callback invoked when the viewer has provided some authentication details. The application should then call `VncServer.accept` to accept or reject the connection based upon the supplied credentials.

Parameters:

`vncServer` - VncServer instance issuing this callback
`username` - client user-name (may be null)
`password` - client password (should not be null)

loginCb

```
void loginCb(VncServer vncServer,  
            boolean usernameReq,  
            boolean passwordReq)
```

Login callback issued during reverse authentication. The application should supply the requested login details to the VNC server using `VncServer.login`.

Parameters:

`vncServer` - VncServer instance issuing this callback

`usernameReq` - true if the server needs to provide a user-name to login to viewer

`passwordReq` - true if the server needs to provide a password to login to viewer

connectedCb

```
void connectedCb(VncServer vncServer,  
                java.lang.String address)
```

Called when a viewer (or data relay) has been connected to us, or we have connected to a viewer. The application should then call `VncServer.accept` to accept or reject the connection attempt from the given address.

Parameters:

`vncServer` - VncServer instance issuing this callback

`address` - of the viewer or data relay that has connected

disconnectedCb

```
void disconnectedCb(VncServer vncServer)
```

Called when the viewer has disconnected from us, or we have disconnected from the viewer.

Parameters:

`vncServer` - VncServer instance issuing this callback

runningCb

```
void runningCb(VncServer vncServer)
```

Called when the VNC viewer and server are connected together and the VNC session has started.

Parameters:

`vncServer` - VncServer instance issuing this callback

errorCb

```
void errorCb(VncServer vncServer,  
            int errorCode,  
            java.lang.Exception e)
```

Called when an unexpected error occurred to indicate that the server has returned to the disconnected state and is no longer listening for a connection, connecting, or connected. It is not necessary to call `VncServer.reset()` following this call.

Parameters:

`vncServer` - VncServer instance issuing this callback

`errorCode` - VNC specific error code as defined by `VncServerCoreErrors`.

e - the exception that originally caused the error, or null if error was not caused by an exception.

remoteControlAvailableCb

```
void remoteControlAvailableCb(VncServer vncServer,  
                             int errorCode)
```

Called in response to the application calling checkRemoteControlAvailable().

Parameters:

vncServer - VncServer instance issuing this callback

errorCode - Error code as defined by [VncServerCoreErrors](#) or 0 if remote control support is available.

com.realvnc.vncserver.android

Interface VncServerMirrorLinkListener

All Superinterfaces:

VncServerListener

All Known Implementing Classes:

MirrorLinkCallbackHandler

```
public interface VncServerMirrorLinkListener
extends VncServerListener
```

Extension to the VncServerListener class to provide extra callbacks in relation to events using the MirrorLink protocol.

See Also:

```
VncServer.mlSendDeviceStatus(VNCDeviceStatus deviceStatus), VncServer.mlSetContextInformation(List
contextInformation)
```

Method Summary

Methods	
Modifier and Type	Method and Description
void	mlAudioBlockingNotificationReceivedCb (VNCAudioBlockingNotification notification) An Audio Blocking Notification message has been received from the MirrorLink viewer.
void	mlClientDisplayConfigurationReceivedCb (VNCCClientDisplayConfiguration clientDisplayConfiguration, VNCServerEventConfiguration defaultConfig) We have received a ClientDisplayConfiguration message from the MirrorLink viewer.
void	mlClientEventConfigurationReceivedCb (VNCCClientEventConfiguration clientEventConfiguration, java.util.Map<java.lang.Integer,java.lang.Integer> defaultMappingRecommendation) The final step of the MirrorLink handshake has occurred - we have received a client event configuration message from the viewer.
void	mlConnectionReceivedCb (VNCServerDisplayConfiguration defaultConfig) A MirrorLink connection has been established to the server.
void	mlDeviceStatusSendNeededCb (VNCDeviceStatus latestRequest, VNCDeviceStatus defaultReply) There is a need to send a DeviceStatus message to the MirrorLink client.
void	mlEventMappingRequestReceivedCb (int clientKeySymbolValue, int serverKeySymbolValue) An event mapping change request message has been received from the MirrorLink viewer.
void	mlFramebufferBlockingNotificationReceivedCb (android.graphics.Rect rect, VNCFramebufferBlockingNotification notification) A Framebuffer Blocking Notification message has been received from the MirrorLink viewer.

Methods inherited from interface com.realvnc.vncserver.android.VncServerListener

```
authCb, connectedCb, connectingCb, disconnectedCb, errorCallback, keygenCb, listeningCb, loginCb,
remoteControlAvailableCb, remoteKeyCb, runningCb
```

Method Detail

mlConnectionReceivedCb

```
void mlConnectionReceivedCb(VNCServerDisplayConfiguration defaultConfig)
```

A MirrorLink connection has been established to the server.

Implementations must arrange to call `VncServer.mlSendServerDisplayConfiguration(VNCServerDisplayConfiguration)`.

The simplest implementation is to pass in 'defaultConfig'. This is a `VNCServerDisplayConfiguration` pre-populated with values suitable for the RealVNC server SDK which you are using. For example, this is set up with appropriate values for:

- MirrorLink versions
- Frame buffer rotation
- Frame buffer scaling
- Pixel format support

Note that these values will correspond to acceptable values for a MirrorLink 1.1 server. If you are not implementing a MirrorLink 1.1 server then you should edit at least the `VNCServerDisplayConfiguration.serverMinorVersion` field before passing it to `VncServer.mlSendServerDisplayConfiguration(com.realvnc.mirrorlink.VNCServerDisplayConfiguration)`.

Additionally, MirrorLink 1.3 deprecates most of the fields in the `VNCServerDisplayConfiguration`. These must also be updated before passing this object to the

```
VncServer.mlSendServerDisplayConfiguration(com.realvnc.mirrorlink.VNCServerDisplayConfiguration).
```

It is acceptable to edit defaultConfig in-place, as this object is discarded after sending to the viewer, and your edits will therefore have no further effects.

Parameters:

defaultConfig - A suitable `VNCServerDisplayConfiguration` which can be passed to the viewer using `VncServer.mlSendServerDisplayConfiguration(VNCServerDisplayConfiguration)`

mlClientDisplayConfigurationReceivedCb

```
void mlClientDisplayConfigurationReceivedCb(VNCClientDisplayConfiguration clientDisplayConfiguration,
                                           VNCServerEventConfiguration defaultConfig)
```

We have received a `ClientDisplayConfiguration` message from the MirrorLink viewer. You may be expected to respond by calling `VncServer.mlSendServerEventConfiguration(VNCServerEventConfiguration)` in response; you must do this **if and only if** the defaultConfig parameter is not null. The `ClientDisplayConfiguration` message may be received:

- as part of the initial MirrorLink negotiation
- subsequently, in order for the client to adjust its preferences as regards the connection.

These two situations can be distinguished by whether the 'defaultConfig' parameter is null. If this parameter is null, you're receiving a CDC message in the normal run-time mode of MirrorLink (the second situation) and no special response is needed. If 'defaultConfig' is not null, you are receiving this message as part of the initial MirrorLink negotiation - the first situation - and the implementation must arrange to call `VncServer.mlSendServerEventConfiguration(VNCServerEventConfiguration)` in response. This enables the negotiation to proceed to the next stage.

If the contents of the client display configuration are unacceptable, you should pass null to `VncServer.mlSendServerEventConfiguration(VNCServerEventConfiguration)` and the connection will be terminated.

Otherwise, call that function with a suitable `VNCServerEventConfiguration` to send to the viewer. The simplest option is to pass in the 'defaultConfig' parameter provided, which is pre-populated with values suitable for the RealVNC server SDK which you are using. However, you have the opportunity to return an alternative version.

You may wish carefully to think about the set of keys which are supported. The MirrorLink compliance test suite insists that keys have an actual function, and therefore our default `VNCServerEventConfiguration` is fairly conservative in the key support it advertises. As an example, the default `VNCServerEventConfiguration` doesn't claim to support the left and right soft keys, because they're not present on most typical Android phones. If your phone has those keys, you should claim such support in your `VNCServerEventConfiguration` message.

Of course, making these decisions requires awareness of exactly what types of key event the VNC server SDK is able to inject into the Android OS in the first place. If appropriate for your device, you may wish to claim support for the following keys beyond those provided in the default server event configuration. The VNC server SDK does make an attempt to inject these into the OS if asked to do so, but it's not sufficiently reliable across all Android devices for us to claim support for them in the default `VNCServerEventConfiguration`:

- XK_Device_Soft_left
- XK_Device_Soft_right
- XK_Device_Phone_call
- XK_Device_Phone_end

Note that CCC-TS-056 does require Android servers to advertise support for these device keys. If possible, they should be enabled by your application.

It *is* acceptable to edit defaultConfig in-place, as this object is discarded after sending to the viewer, and your edits will therefore have no further effects.

Note that certain fields of the `VNCClientDisplayConfiguration` are deprecated in MirrorLink 1.3. These are outlined in the `VNCClientDisplayConfiguration`. If the client and server have both indicated support for MirrorLink 1.3 or above, then the server SDK will log warnings for any deprecated values that have been set incorrectly by the client in the `VNCClientDisplayConfiguration`. The server SDK will continue normal operation and pass these values unchanged to this method.

Parameters:

`clientDisplayConfiguration` - The client display configuration information received from the MirrorLink client. The client display width and height in pixels is adjusted to the reference display, or previous client display size (if one is known) if it is invalid (either width, or height is zero).

`defaultConfig` - A sensible pre-populated `ServerEventConfiguration` which can be passed to the viewer

mlClientEventConfigurationReceivedCb

```
void mlClientEventConfigurationReceivedCb(VNCClientEventConfiguration clientEventConfiguration,
    java.util.Map<java.lang.Integer,java.lang.Integer> defaultMappingRecommendation)
```

The final step of the MirrorLink handshake has occurred - we have received a client event configuration message from the viewer.

Although this is formally the last stage of the MirrorLink handshake, the MirrorLink specification does state that servers SHOULD sent Event Mapping messages just after the client event configuration is received. To achieve that, you should normally call `VncServer.mlSetEventMapping(Map)` to set up appropriate mappings. It is recommended that you base this mapping on the values passed in the `defaultMappingRecommendation` parameter, which will be pre-configured to be roughly suitable based on the information provided within the client event configuration message.

Parameters:

`clientEventConfiguration` - The client event configuration

`defaultMappingRecommendation` - A default mapping derived from the client event configuration, suitable for the android platform. These can be used as a basis by the server application to derive its own event mapping scheme before invoking `mlSetEventMapping` to transmit them to the client

mlDeviceStatusSendNeededCb

```
void mlDeviceStatusSendNeededCb(VNCDeviceStatus latestRequest,
    VNCDeviceStatus defaultReply)
```

There is a need to send a `DeviceStatus` message to the MirrorLink client.

You must reply immediately using `VncServer.mlSendDeviceStatus(VNCDeviceStatus deviceStatus)`. The simplest way to reply is to call this function using the provided 'defaultReply'.

This function may be called for a variety of reasons:

- The client has requested an update to the device status using a `DeviceStatusRequest` message.
- You yourself have requested an update, using the `VncServer.mlRequestSendDeviceStatus()` API.
- The VNC server SDK has detected a change in device status and wishes to ensure the client is informed.

It is important to call this in a timely fashion, because it may be time-critical to inform the VNC client of changes to the screen rotation. The VNC server SDK may be unable to provide any further image updates to the client until you have responded to this callback.

The RealVNC server SDK will handle some features internally:

- Framebuffer rotation
- Framebuffer orientation
- Framebuffer scaling

Other features will not be handled by the VNC server SDK and should be handled by the VNC server application. Note that there may be sample implementations of some of these features in our sample server application code.

Note that some features are deprecated for MirrorLink 1.3 and above. These should not be used. Full detail is given in the documentation for `VncDeviceStatus`.

The 'latestRequest' parameter is always the latest device status request received from the viewer. It may be null, if no such request has ever been received.

The 'defaultReply' parameter will always contain valid values for the version of MirrorLink that the server has indicated support for.

Parameters:

latestRequest - The device status request

defaultReply - A proposed response

mlFramebufferBlockingNotificationReceivedCb

```
void mlFramebufferBlockingNotificationReceivedCb(android.graphics.Rect rect,
VncFramebufferBlockingNotification notification)
```

A Framebuffer Blocking Notification message has been received from the MirrorLink viewer.

It is the responsibility of the VNC server application to handle this according to the MirrorLink specification, which may for example involve minimising the current application. After this has been achieved, the VNC server application must reply as soon as possible by calling `VncServer.mlFramebufferBlockingNotificationHandled()`. Until this is called, all normal VNC processing is suspended - so it's important to resolve the situation as fast as possible to avoid the user seeing a jarring experience.

Parameters:

rect - The screen rect to which this applies

notification - The blocking notification

mlAudioBlockingNotificationReceivedCb

```
void mlAudioBlockingNotificationReceivedCb(VncAudioBlockingNotification notification)
```

An Audio Blocking Notification message has been received from the MirrorLink viewer.

It is the responsibility of the VNC server application to handle this according to the MirrorLink specification.

Parameters:

notification - The blocking notification

mlEventMappingRequestReceivedCb

```
void mlEventMappingRequestReceivedCb(int clientKeySymbolValue,
int serverKeySymbolValue)
```

An event mapping **change** request message has been received from the MirrorLink viewer.

Important: this is not called for every event mapping request message received by the server. Those messages which simply request information about the present mapping are handled automatically by the server SDK, and result in no callback up to the server application. Only those messages which request changes are reported to the server application by this callback.

In addition, you will not receive a callback here if the client has requested no actual changes to the mapping (i.e. they have requested a new mapping which happens to exactly match the mapping already in use). If you receive this callback, you genuinely know that a change has been requested by the client.

The VNC server application must reply using `VncServer.mlSendEventMappingRequestReply(int)`. It is important to provide this reply in a timely fashion, because no further VNC messages can be passed until the reply has been sent to the client. If you wish to actually grant the request and update the key mapping, you should first do so using `VncServer.mlSetEventMapping(Map)` - see the documentation of `VncServer.mlSendEventMappingRequestReply(int)` for sample code.

Parameters:

clientKeySymbolValue - The client key symbol value

<div>serverKeySymbolValue - The server key symbol value</div>

com.realvnc.vncserver.android

Interface VncServerOrientationListener

All Superinterfaces:

VncServerListener

All Known Implementing Classes:

MirrorLinkCallbackHandler, VncServerCallbackHandler

```
public interface VncServerOrientationListener
extends VncServerListener
```

A type of listener which can be informed of orientation changes detected by the Android VNC server SDK.

Method Summary

Methods	
Modifier and Type	Method and Description
void	<code>displayOrientationChangedCb(int orientation)</code> Called when the orientation of the VNC server framebuffer has changed.
void	<code>displayOrientationChangeNeededCb(int orientNeeded)</code> Called when the SDK requires the application to change the current display orientation.

Methods inherited from interface com.realvnc.vncserver.android.VncServerListener

`authCb, connectedCb, connectingCb, disconnectedCb, errorCallback, keygenCb, listeningCb, loginCb, remoteControlAvailableCb, remoteKeyCb, runningCb`

Method Detail

displayOrientationChangedCb

`void displayOrientationChangedCb(int orientation)`

Called when the orientation of the VNC server framebuffer has changed.

This is called after the internals of the server SDK have been made aware of the orientation change.

Parameters:

 `orientation` - The new orientation, specified as one of android.view.Surface.ROTATION_*

displayOrientationChangeNeededCb

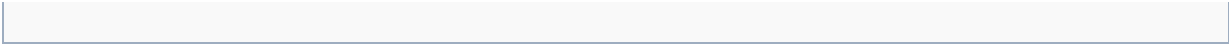
`void displayOrientationChangeNeededCb(int orientNeeded)`

Called when the SDK requires the application to change the current display orientation.

This callback can occur when the connected viewer has indicated that it can't support the screen being resized or rotated after the connection has been established.

Parameters:

 `orientNeeded` - The orientation the display should be changed into, specified as one of android.view.Surface.ROTATION_*



com.realvnc.vncserver.android

Class VncSizeInt

java.lang.Object

com.realvnc.vncserver.android.VncSizeInt

```
public class VncSizeInt
extends java.lang.Object
```

Represents a width and height.

Constructor Summary

Constructors

Constructor and Description

`VncSizeInt(int width, int height)`

Constructs a `VncSizeInt` object with the specified width and height.

Method Summary

Methods

Modifier and Type	Method and Description
boolean	<code>equals(java.lang.Object o)</code>
int	<code>getHeight()</code> Returns the height contained in this object.
int	<code>getWidth()</code> Returns the width contained in this object.
int	<code>hashCode()</code>
java.lang.String	<code>toString()</code>

Methods inherited from class java.lang.Object

`clone, finalize, getClass, notify, notifyAll, wait, wait, wait`

Constructor Detail

VncSizeInt

```
public VncSizeInt(int width,
                  int height)
```

Constructs a `VncSizeInt` object with the specified width and height. If the width or height are negative, an `IllegalArgumentException` will be thrown.

Method Detail

getWidth

```
public int getWidth()
```

Returns the width contained in this object.

getHeight

```
public int getHeight()
```

Returns the height contained in this object.

equals

```
public boolean equals(java.lang.Object o)
```

Overrides:

equals in class java.lang.Object

hashCode

```
public int hashCode()
```

Overrides:

hashCode in class java.lang.Object

toString

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

Overrides:

toString in class java.lang.Object

com.realvnc.vncserver.core

Class VncAuthType

java.lang.Object
com.realvnc.vncserver.core.VncAuthType

```
public class VncAuthType
extends java.lang.Object
```

Type of authentication to be used by the VNC server.

Field Summary

Fields	
Modifier and Type	Field and Description
static int	VNC_AUTH_NONE No authentication is to be used.
static int	VNC_AUTH_PASS Password authentication - the viewer must provide a password to be authenticated by the server.
static int	VNC_AUTH_REV Reverse Authentication - the VNC server must provide either a password or a username and password to be authenticated by the viewer.
static int	VNC_AUTH_USER_PASS Username and password authentication - the viewer must provide a username and password to be authenticated by the server.

Method Summary

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

Field Detail

VNC_AUTH_REV

```
public static final int VNC_AUTH_REV
```

Reverse Authentication - the VNC server must provide either a password or a username and password to be authenticated by the viewer.

See Also:

[Constant Field Values](#)

VNC_AUTH_NONE

```
public static final int VNC_AUTH_NONE
```

No authentication is to be used.

See Also:

[Constant Field Values](#)**VNC_AUTH_PASS**

```
public static final int VNC_AUTH_PASS
```

Password authentication - the viewer must provide a password to be authenticated by the server.

See Also:[Constant Field Values](#)**VNC_AUTH_USER_PASS**

```
public static final int VNC_AUTH_USER_PASS
```

Username and password authentication - the viewer must provide a username and password to be authenticated by the server.

See Also:[Constant Field Values](#)

com.realvnc.vncserver.core

Interface VncBearerCallbacks

public interface VncBearerCallbacks

Objects implementing this interface are used by the SDK to provide a way for the bearers to call SDK provided functionality.

Method Summary

Methods	
Modifier and Type	Method and Description
boolean	<code>localFeatureCheck(int[] featureIds)</code> Requests that the SDK performs a feature check on the local licenses.

Method Detail

localFeatureCheck
<pre>boolean localFeatureCheck(int[] featureIds) throws VncException</pre> <p>Requests that the SDK performs a feature check on the local licenses.</p> <p>Parameters:</p> <p><code>featureIds</code> - List of feature IDs required.</p> <p>Returns:</p> <p>True if one or more of the features in <code>featureIds</code> are present, false otherwise.</p> <p>Throws:</p> <p><code>VncException</code></p>

com.realvnc.vncserver.core

Interface VncBearer

```
public interface VncBearer
```

Objects implementing this interface are used by the SDK to provide data transport facilities between the server and viewer. TCP/IP, data relay, and USB bearers are provided as part of the SDK, and other custom bearers may be added as required.

Method Summary

Methods

Modifier and Type	Method and Description
<code>VncConnection</code>	<code>createConnection(VncCommandStringBase commandString, VncBearerCallbacks callbacks)</code> Create a new connection object which can be used to establish a new transport session over this bearer.
<code>VncBearerInfo</code>	<code>getInfo()</code> Returns an object containing descriptive information about the bearer.

Method Detail

getInfo

```
VncBearerInfo getInfo()
```

Returns an object containing descriptive information about the bearer.

createConnection

```
VncConnection createConnection(VncCommandStringBase commandString,  
                                VncBearerCallbacks callbacks)  
    throws VncException
```

Create a new connection object which can be used to establish a new transport session over this bearer. This method must either succeed or throw an exception and should not block. This call does not cause the connection attempt to be started - for that the `VncConnection.establish()` method of the `VncConnection` object should be used.

Parameters:

`commandString` - details used for establishing the connection

`callbacks` - object for accessing SDK functionality from within the bearer.

Returns:

`VncConnection` a connection object was successfully created and can be used to establish the connection

Throws:

`VncException` - a connection object could not be created

com.realvnc.vncserver.core

Interface VncBearerInfo

public interface VncBearerInfo

Objects implementing this interface are used to provided detailed information on a pluggable bearer, and can be obtained through a call to the `getBearerInfo` method of the `VncServer` object.

Method Summary

Methods	
Modifier and Type	Method and Description
java.lang.String	<code>getDescription()</code> Returns a description of this bearer and the transports that it supports.
java.lang.String	<code>getFullName()</code> Returns a longer human readable name for this bearer.
java.lang.String	<code>getName()</code> Returns the short name for this bearer.
java.lang.String	<code>getVersionString()</code> Returns the version string for this bearer.

Method Detail

getName
<pre>java.lang.String getName()</pre> <p>Returns the short name for this bearer. This is the name that will be taken from the command string and used to look up a bearer object.</p>
getFullName
<pre>java.lang.String getFullName()</pre> <p>Returns a longer human readable name for this bearer.</p>
getDescription
<pre>java.lang.String getDescription()</pre> <p>Returns a description of this bearer and the transports that it supports.</p>
getVersionString
<pre>java.lang.String getVersionString()</pre> <p>Returns the version string for this bearer.</p>

com.realvnc.vncserver.core

Class VncCommandStringBase

java.lang.Object
com.realvnc.vncserver.core.VncCommandStringBase

Direct Known Subclasses:

VncCommandString

```
public abstract class VncCommandStringBase
extends java.lang.Object
```

A abstract class for encapsulating a VNC command string. Note that in the past the terms "connection string" and "command string" have been used more-or-less interchangeably; both terms refer to the same thing. This class must be subclassed to provide an implementation for the base 64 decoding.

Field Summary

Fields	
Modifier and Type	Field and Description
protected java.util.Hashtable	<code>fields</code> java.lang.String, java.lang.String> Table for associating command string field names with their values.
static java.lang.String	<code>TYPE</code> Names for known command string key/value pairs
static java.lang.String	<code>VERSION</code>

Constructor Summary

Constructors	
Constructor and Description	
<code>VncCommandStringBase()</code>	Create a new object representing an initially empty command string.

Method Summary

Methods	
Modifier and Type	Method and Description
protected abstract byte[]	<code>decodeBase64</code> (java.lang.String input) Take a base 64 encoded string and return the decoded output.
byte[]	<code>getBase64Value</code> (java.lang.String parameterName) Extract the named field from the command string (which is assumed to be base 64 encoded), and decode it.
boolean	<code>getBoolean</code> (java.lang.String parameterName) Extract the named field from the command string and convert to a boolean.
int	<code>getInt</code> (java.lang.String parameterName) Extract the named field from the command string and convert to an integer.
java.lang.String	<code>getString</code> (java.lang.String parameterName) Extract the named field from the command string and return as a string.
boolean	<code>parameterPresent</code> (java.lang.String parameterName)

	Returns true if the named parameter was successfully parsed from the command string, otherwise false.
void	<code>parse(java.lang.String commandString)</code> Parse the given command string and break it down into separate fields which are then checked added as key/value strings to the main hash table.
protected void	<code>putField(java.lang.String key, java.lang.String val)</code> Add the given key and value to the parameter hashtable.

Methods inherited from class java.lang.Object

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

Field Detail

TYPE

`public static java.lang.String TYPE`

Names for known command string key/value pairs

VERSION

`public static java.lang.String VERSION`

fields

`protected java.util.Hashtable<java.lang.String,java.lang.String> fields`

Table for associating command string field names with their values. Keys are values are both strings and are converted to the required type from the get methods.

Constructor Detail

VncCommandStringBase

`public VncCommandStringBase()`

Create a new object representing an initially empty command string.

Method Detail

putField

`protected void putField(java.lang.String key,
 java.lang.String val)
 throws VncException`

Add the given key and value to the parameter hashtable.

Parameters:

`key` - of item to be added

val - item to be added

Throws:

`VncException` - if the key is already present in the table or if the given key and value are not URL safe

parse

```
public void parse(java.lang.String commandString)
    throws VncException
```

Parse the given command string and break it down into separate fields which are then checked added as key/value strings to the main hash table. Each of the individual get routines then converts the stored string value back to the requested type.

Parameters:

commandString - the command string to be parsed

Throws:

`VncException` - when an error occurs during the parsing of the command string

parameterPresent

```
public boolean parameterPresent(java.lang.String parameterName)
```

Returns true if the named parameter was successfully parsed from the command string, otherwise false.

getInt

```
public int getInt(java.lang.String parameterName)
    throws VncException
```

Extract the named field from the command string and convert to an integer.

Parameters:

parameterName - name of the field to be returned

Returns:

the value of the named field converted to an integer

Throws:

`VncException` - the named field was not present or could be converted into a integer

getBoolean

```
public boolean getBoolean(java.lang.String parameterName)
    throws VncException
```

Extract the named field from the command string and convert to a boolean. Note that we treat 1 as true, and *everything* else as false.

Parameters:

parameterName - name of the field to be returned

Returns:

the value of the named field converted to an boolean

Throws:

`VncException` - the named field was not present or could be converted into a integer

getString

```
public java.lang.String getString(java.lang.String parameterName)
    throws VncException
```

Extract the named field from the command string and return as a string. Note that this method will not automatically decode any fields which are encoded as base 64.

Parameters:

parameterName - name of the field to be returned

Returns:

the value of the named field converted to an boolean

Throws:

`VncException` - the named field was not present

getBase64Value

```
public byte[] getBase64Value(java.lang.String parameterName)
    throws VncException
```

Extract the named field from the command string (which is assumed to be base 64 encoded), and decode it.

Parameters:

parameterName - name of the field to be returned

Returns:

the value of the named field converted to an boolean

Throws:

`VncException` - the named field was not present or could not be decoded

decodeBase64

```
protected abstract byte[] decodeBase64(java.lang.String input)
    throws java.lang.Exception
```

Take a base 64 encoded string and return the decoded output. Implementation should pad the input to a 4 character boundary before decoding.

Parameters:

input - base 64 encoded string to be decoded

Returns:

decoded output

Throws:

`java.lang.Exception` - if an error occurs during the decoding

com.realvnc.vncserver.core

Interface VncConnection

```
public interface VncConnection
```

An object representing a connection across which the server will talk to a VNC viewer. These objects are created by [VncBearers](#) which should return an instance of a class implementing this interface from their [createConnection](#) method.

Method Summary

Methods

Modifier and Type	Method and Description
void	close() If the connection is already established then close it, or if we're still trying to establish a connection give up.
boolean	establish() Establish a connection over the bearer using the connection details in the command string passed to the createConnection method of the object implementing the VncBearer interface.
java.io.InputStream	getInputStream() Once a connection has been established returns an InputStream which can be used to read data over the bearer.
java.lang.String	getLocalAddress() Return the local address associated with this connection.
java.io.OutputStream	getOutputStream() Once a connection has been established returns an OutputStream which can be used to write data over the bearer.
java.lang.String	getRemoteAddress() Return the remote address associated with this connection.

Method Detail

establish

```
boolean establish()  
    throws VncException
```

Establish a connection over the bearer using the connection details in the command string passed to the [createConnection](#) method of the object implementing the [VncBearer](#) interface. This call blocks until the connection is full established or an error occurs. In the case of a listening bearer this would mean that the call blocks until someone connects and completes the connection.

Returns:

false if close was called before the connection could be established, or true if the connection was successfully established.

Throws:

[VncException](#) - an error occurred during the attempt to establish the connection

close

```
void close()
```

If the connection is already established then close it, or if we're still trying to establish a connection give up. This will cause any blocked calls to [establish\(\)](#) to return false at some point in the future but not necessarily immediately.

getInputStream

```
java.io.InputStream getInputStream()
```

Once a connection has been established returns an InputStream which can be used to read data over the bearer.

Returns:

InputStream or null if the connection was not established

getOutputStream

```
java.io.OutputStream getOutputStream()
```

Once a connection has been established returns an OutputStream which can be used to write data over the bearer.

Returns:

OutputStream or null if the connection was not established

getLocalAddress

```
java.lang.String getLocalAddress()
```

Return the local address associated with this connection. By default this is null which indicates that the connection is an out bound one. An implementation that provides a listening connection should override to provide details on the local address.

getRemoteAddress

```
java.lang.String getRemoteAddress()
```

Return the remote address associated with this connection. This will be null if the connection has not yet been established.

com.realvnc.vncserver.core

Class VncEncryptionType

java.lang.Object
com.realvnc.vncserver.core.VncEncryptionType

```
public class VncEncryptionType
extends java.lang.Object
```

Type of authentication to be used by the VNC server.

Field Summary

Fields	
Modifier and Type	Field and Description
static int	VNC_ENCRYPTION_AES_128 Use 128 bit AES encryption.
static int	VNC_ENCRYPTION_NONE Don't use encryption.

Method Summary

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

Field Detail

VNC_ENCRYPTION_NONE

```
public static final int VNC_ENCRYPTION_NONE
```

Don't use encryption.

See Also:

[Constant Field Values](#)

VNC_ENCRYPTION_AES_128

```
public static final int VNC_ENCRYPTION_AES_128
```

Use 128 bit AES encryption.

See Also:

[Constant Field Values](#)

com.realvnc.vncserver.core

Class VncException

java.lang.Object
 java.lang.Throwable
 java.lang.Exception
 com.realvnc.vncserver.core.VncException

All Implemented Interfaces:

java.io.Serializable

Direct Known Subclasses:

VncLicenseNotValidException

public class **VncException**
extends java.lang.Exception

An exception class to describe errors using standard VNC error codes.

See Also:

VncServerCoreErrors, Serialized Form

Field Summary

Fields

Modifier and Type	Field and Description
java.lang.Exception	cause
int	errorCode
static long	serialVersionUID

Constructor Summary

Constructors

Constructor and Description
VncException (int errorCode)
VncException (int errorCode, java.lang.Exception e)
VncException (int errorCode, java.lang.String s)
VncException (int errorCode, java.lang.String s, java.lang.Exception e)
VncException (int errorCode, java.lang.String s, java.lang.Throwable e)
VncException (int errorCode, java.lang.Throwable e)

Method Summary

Methods

Modifier and Type	Method and Description
java.lang.String	getMessage ()

Methods inherited from class java.lang.Throwable

addSuppressed, fillInStackTrace, getCause, getLocalizedMessage, getStackTrace, getSuppressed, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.Object

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

Field Detail**serialVersionUID**

```
public static final long serialVersionUID
```

See Also:

[Constant Field Values](#)

errorCode

```
public int errorCode
```

cause

```
public java.lang.Exception cause
```

Constructor Detail**VncException**

```
public VncException(int errorCode)
```

VncException

```
public VncException(int errorCode,  
                    java.lang.String s)
```

VncException

```
public VncException(int errorCode,  
                    java.lang.Exception e)
```

VncException

```
public VncException(int errorCode,  
                    java.lang.Throwable e)
```

VncException

```
public VncException(int errorCode,  
                    java.lang.String s,  
                    java.lang.Exception e)
```

VncException

```
public VncException(int errorCode,  
                    java.lang.String s,  
                    java.lang.Throwable e)
```

Method Detail**getMessage**

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

Overrides:

getMessage in class java.lang.Throwable

com.realvnc.vncserver.core

Class VncLicenseNotValidException

java.lang.Object
 java.lang.Throwable
 java.lang.Exception
 com.realvnc.vncserver.core.VncException
 com.realvnc.vncserver.core.VncLicenseNotValidException

All Implemented Interfaces:

java.io.Serializable

```
public class VncLicenseNotValidException
extends VncException
```

An exception class to describe license not valid errors.

See Also:

[Serialized Form](#)

Field Summary

Fields

Modifier and Type	Field and Description
static long	<code>serialVersionUID</code>

Fields inherited from class com.realvnc.vncserver.core.VncException

`cause`, `errorCode`

Constructor Summary

Constructors

Constructor and Description
<code>VncLicenseNotValidException()</code> Constructs a license not valid exception for license errors where the serial number of the license couldn't be determined.
<code>VncLicenseNotValidException(byte[] serialNumber)</code> Constructs a license not valid exception for license errors where the serial number of the license could be determined.

Method Summary

Methods

Modifier and Type	Method and Description
byte[]	<code>getSerialNumber()</code> Retrieves the serial number of the license which was invalid.

Methods inherited from class com.realvnc.vncserver.core.VncException

`getMessage`

Methods inherited from class java.lang.Throwable

addSuppressed, fillInStackTrace, getCause, getLocalizedMessage, getStackTrace, getSuppressed, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.Object

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

Field Detail**serialVersionUID**

```
public static final long serialVersionUID
```

See Also:

[Constant Field Values](#)

Constructor Detail**VncLicenseNotValidException**

```
public VncLicenseNotValidException()
```

Constructs a license not valid exception for license errors where the serial number of the license couldn't be determined.

VncLicenseNotValidException

```
public VncLicenseNotValidException(byte[] serialNumber)
```

Constructs a license not valid exception for license errors where the serial number of the license could be determined.

Parameters:

`serialNumber` - The serial number of the invalid license

Method Detail**getSerialNumber**

```
public byte[] getSerialNumber()
```

Retrieves the serial number of the license which was invalid.

Returns:

The invalid serial number, null if the serial number couldn't be determined

com.realvnc.vncserver.core

Class VncPixelFormat

java.lang.Object
com.realvnc.vncserver.core.VncPixelFormat

```
public class VncPixelFormat
extends java.lang.Object
```

Defines the format of the pixels in a framebuffer.

Byte Order

Pixel data can be big-endian or little-endian. If the byte order differs from the byte order that is used by your application's framebuffer, then you must swap the byte order of every pixel before further interpreting it.

Using True Color

If `getTrueColorFlag()` is non-zero, then the red, green and blue components of each pixel are contained directly in the pixel data. Each component can be extracted using the maximum and shift values for that component.

The maximum value for each component is $(2^n - 1)$, where n is the number of bits used to store that component. For example, if a pixel's red component is equal to `getRedMax()`, then its red component is at full intensity.

The shift value for each component is the number of right-shifts that must be applied to each pixel value to move the least-significant-bit of that component component into the least-significant-bit of the result. For example, after dealing with the pixel byte order, the red component can be extracted by calculating `((pixel >> getRedShift()) & getRedMax())`.

For example, if `bitsPerPixel` is 8 and each pixel has the format 00rrggbb, then:

- `getRedMax()`, `getGreenMax()` and `getBlueMax()` are all equal to 3
- `getRedShift()` is 4
- `getBlueShift()` is 2
- `getGreenShift()` is 0

Using an Indexed Pixel Format

If `getTrueColorFlag()` is zero, then the pixel format is said to be 'indexed'. This means that each value in the pixel data is an index into a color map chosen by the VNC Server. Indexed pixel formats often look better than true color pixel formats when the number of distinct pixel values is small.

Field Summary

Fields

Modifier and Type	Field and Description
static VncPixelFormat	PAL_8 A <code>VncPixelFormat</code> instance representing a pixel format where each component has an 8-bit representation.
static VncPixelFormat	RGB_332 A <code>VncPixelFormat</code> instance representing RGB332.
static VncPixelFormat	RGB_565 A <code>VncPixelFormat</code> instance representing RGB565.
static VncPixelFormat	RGB_888 A <code>VncPixelFormat</code> instance representing RGB888.

Constructor Summary

Constructors

Constructor and Description

`VncPixelFormat()`

Constructs an empty VncPixelFormat instance.

`VncPixelFormat(int depth, int bitsPerPixel, boolean bigEndianFlag, boolean trueColorFlag, int redShift, int greenShift, int blueShift, int redMax, int greenMax, int blueMax)`

Constructs and populates a VncPixelFormat instance.

`VncPixelFormat(VncPixelFormat src)`

Copies a VncPixelFormat instance.

Method Summary

Methods

Modifier and Type	Method and Description
boolean	<code>equals(java.lang.Object o)</code> Tests if this VncPixelFormat instance is equal to the given Object.
boolean	<code>getBigEndianFlag()</code> Returns the big endian flag for this pixel format.
int	<code>getBitsPerPixel()</code> Returns the bits per pixel for this pixel format.
int	<code>getBlueMax()</code> Returns the maximum value of the blue component for this pixel format.
int	<code>getBlueShift()</code> Returns the blue shift value for this pixel format.
int	<code>getDepth()</code> Returns the depth for this pixel format.
int	<code>getGreenMax()</code> Returns the maximum value of the green component for this pixel format.
int	<code>getGreenShift()</code> Returns the green shift value for this pixel format.
int	<code>getRedMax()</code> Returns the maximum value of the red component for this pixel format.
int	<code>getRedShift()</code> Returns the red shift value for this pixel format.
boolean	<code>getTrueColorFlag()</code> Returns the true color flag for this pixel format.
int	<code>hashCode()</code> Calculates a hash code for the object.
void	<code>set(VncPixelFormat src)</code> Sets the instance to have the same values as another VncPixelFormat instance.
<code>VncPixelFormat</code>	<code>setBigEndianFlag(boolean bigEndianFlag)</code> Sets the big endian flag for this pixel format.
<code>VncPixelFormat</code>	<code>setBitsPerPixel(int bitsPerPixel)</code> Sets the bits per pixel for this pixel format.
<code>VncPixelFormat</code>	<code>setBlueMax(int blueMax)</code> Sets the maximum value of the blue component of each pixel for this pixel format.
<code>VncPixelFormat</code>	<code>setBlueShift(int blueShift)</code> Sets the shift value of the blue component of each pixel for this pixel format.
<code>VncPixelFormat</code>	<code>setDepth(int depth)</code> Sets the depth for this pixel format.

<code>VncPixelFormat</code>	<code>setGreenMax(int greenMax)</code> Sets the maximum value of the green component of each pixel for this pixel format.
<code>VncPixelFormat</code>	<code>setGreenShift(int greenShift)</code> Sets the shift value of the green component of each pixel for this pixel format.
<code>VncPixelFormat</code>	<code>setRedMax(int redMax)</code> Sets the maximum value of the red component of each pixel for this pixel format.
<code>VncPixelFormat</code>	<code>setRedShift(int redShift)</code> Sets the shift value of the red component of each pixel for this pixel format.
<code>VncPixelFormat</code>	<code>setTrueColorFlag(boolean trueColorFlag)</code> Sets the true color flag for this pixel format.

Methods inherited from class `java.lang.Object`

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

Field Detail

RGB_888

```
public static final VncPixelFormat RGB_888
```

A `VncPixelFormat` instance representing RGB888.

RGB_565

```
public static final VncPixelFormat RGB_565
```

A `VncPixelFormat` instance representing RGB565.

RGB_332

```
public static final VncPixelFormat RGB_332
```

A `VncPixelFormat` instance representing RGB332.

PAL_8

```
public static final VncPixelFormat PAL_8
```

A `VncPixelFormat` instance representing a pixel format where each component has an 8-bit representation.

Constructor Detail

VncPixelFormat

```
public VncPixelFormat(int depth,
                     int bitsPerPixel,
                     boolean bigEndianFlag,
                     boolean trueColorFlag,
                     int redShift,
                     int greenShift,
                     int blueShift,
                     int redMax,
                     int greenMax,
                     int blueMax)
```

Constructs and populates a VncPixelFormat instance.

Parameters:

`depth` - The depth for this pixel format.

`bitsPerPixel` - The bits per pixel for this pixel format.

`bigEndianFlag` - The big endian flag for this pixel format.

`trueColorFlag` - The true color flag for this pixel format.

`redShift` - The red shift value for this pixel format.

`greenShift` - The green shift value for this pixel format.

`blueShift` - The blue shift value for this pixel format.

`redMax` - The maximum value of the red component for this pixel format.

`greenMax` - The maximum value of the green component for this pixel format.

`blueMax` - The maximum value of the blue component for this pixel format.

VncPixelFormat

```
public VncPixelFormat()
```

Constructs an empty VncPixelFormat instance.

VncPixelFormat

```
public VncPixelFormat(VncPixelFormat src)
```

Copies a VncPixelFormat instance.

Parameters:

`src` - The VncPixelFormat instance to copy.

Method Detail**set**

```
public void set(VncPixelFormat src)
```

Sets the instance to have the same values as another VncPixelFormat instance.

Parameters:

`src` - The VncPixelFormat instance to copy.

setDepth

```
public VncPixelFormat setDepth(int depth)
```

Sets the depth for this pixel format.

This is the number of bits in each pixel that are significant (i.e. the number of colors that can be represented). This must be less than or equal to `getBitsPerPixel()`. (For example, if 32 bits are used to represent pixels that have 8 bits each for red, green and blue, then `getBitsPerPixel()` is 32 and depth is 24.

Parameters:

depth - The depth for this pixel format.

Returns:

This VncPixelFormat instance.

getDepth

```
public int getDepth()
```

Returns the depth for this pixel format.

Returns:

The depth for this pixel format.

See Also:

```
setDepth(int)
```

setBitsPerPixel

```
public VncPixelFormat setBitsPerPixel(int bitsPerPixel)
```

Sets the bits per pixel for this pixel format.

This is number of bits of pixel data used for each pixel. This must be either 8, 16, or 32.

Your application's framebuffer must be aligned so that the address of each pixel is a multiple of (bitsPerPixel / 8).

Parameters:

bitsPerPixel - The bits per pixel for this pixel format.

Returns:

This VncPixelFormat instance.

getBitsPerPixel

```
public int getBitsPerPixel()
```

Returns the bits per pixel for this pixel format.

Returns:

The bits per pixel for this pixel format.

See Also:

```
setBitsPerPixel(int)
```

setBigEndianFlag

```
public VncPixelFormat setBigEndianFlag(boolean bigEndianFlag)
```

Sets the big endian flag for this pixel format.

If this flag is set to true, then the pixels are big-endian. Otherwise, they are little-endian.

Parameters:

bigEndianFlag - The big endian flag for this pixel format.

Returns:

This VncPixelFormat instance.

getBigEndianFlag

```
public boolean getBigEndianFlag()
```

Returns the big endian flag for this pixel format.

Returns:

The big endian flag for this pixel format.

See Also:

```
setBigEndianFlag(boolean)
```

setTrueColorFlag

```
public VncPixelFormat setTrueColorFlag(boolean trueColorFlag)
```

Sets the true color flag for this pixel format.

If this flag is set to true, then the pixel data contains actual pixel values. Otherwise, the pixel data contains indices into the accompanying color map.

Parameters:

`trueColorFlag` - The true color flag for this pixel format.

Returns:

This VncPixelFormat instance.

getTrueColorFlag

```
public boolean getTrueColorFlag()
```

Returns the true color flag for this pixel format.

Returns:

The true color flag for this pixel format.

See Also:

```
setTrueColorFlag(boolean)
```

setRedShift

```
public VncPixelFormat setRedShift(int redShift)
```

Sets the shift value of the red component of each pixel for this pixel format.

Parameters:

`redShift` - The red shift value for this pixel format.

Returns:

This VncPixelFormat instance.

getRedShift

```
public int getRedShift()
```

Returns the red shift value for this pixel format.

Returns:

The red shift value for this pixel format.

See Also:

`setRedShift(int)`

setGreenShift

```
public VncPixelFormat setGreenShift(int greenShift)
```

Sets the shift value of the green component of each pixel for this pixel format.

Parameters:

`greenShift` - The green shift value for this pixel format.

Returns:

This VncPixelFormat instance.

getGreenShift

```
public int getGreenShift()
```

Returns the green shift value for this pixel format.

Returns:

The green shift value for this pixel format.

See Also:

`setGreenShift(int)`

setBlueShift

```
public VncPixelFormat setBlueShift(int blueShift)
```

Sets the shift value of the blue component of each pixel for this pixel format.

Parameters:

`blueShift` - The blue shift value for this pixel format.

Returns:

This VncPixelFormat instance.

getBlueShift

```
public int getBlueShift()
```

Returns the blue shift value for this pixel format.

Returns:

The blue shift value for this pixel format.

See Also:

`setBlueShift(int)`

setRedMax

```
public VncPixelFormat setRedMax(int redMax)
```

Sets the maximum value of the red component of each pixel for this pixel format.

Parameters:

`redMax` - The maximum value of the red component for this pixel format.

Returns:

This VncPixelFormat instance.

getRedMax

```
public int getRedMax()
```

Returns the maximum value of the red component for this pixel format.

Returns:

The maximum value of the red component for this pixel format.

See Also:

`setRedMax(int)`

setGreenMax

```
public VncPixelFormat setGreenMax(int greenMax)
```

Sets the maximum value of the green component of each pixel for this pixel format.

Parameters:

`greenMax` - The maximum value of the green component for this pixel format.

Returns:

This VncPixelFormat instance.

getGreenMax

```
public int getGreenMax()
```

Returns the maximum value of the green component for this pixel format.

Returns:

The maximum value of the green component for this pixel format.

See Also:

`setGreenMax(int)`

setBlueMax

```
public VncPixelFormat setBlueMax(int blueMax)
```

Sets the maximum value of the blue component of each pixel for this pixel format.

Parameters:

`blueMax` - The maximum value of the blue component for this pixel format.

Returns:

This VncPixelFormat instance.

getBlueMax

```
public int getBlueMax()
```

Returns the maximum value of the blue component for this pixel format.

Returns:

The maximum value of the blue component for this pixel format.

See Also:

```
setBlueMax(int)
```

equals

```
public boolean equals(java.lang.Object o)
```

Tests if this VncPixelFormat instance is equal to the given Object.

Overrides:

`equals` in class `java.lang.Object`

Parameters:

o - The Object to compare this VncPixelFormat instance with.

Returns:

True if this VncPixelFormat instance is equal to the Object, otherwise false.

hashCode

```
public int hashCode()
```

Calculates a hash code for the object.

Overrides:

`hashCode` in class `java.lang.Object`

Returns:

The hash code of the object.

com.realvnc.vncserver.core

Class VncServerCoreErrors

java.lang.Object
com.realvnc.vncserver.core.VncServerCoreErrors

public class VncServerCoreErrors
extends java.lang.Object

VNC specific error codes to be returned from the VNC server.

See Also:

VncException

Field Summary

Fields	
Modifier and Type	Field and Description
static int	VNCSERVER_ERR_ALREADY_EXISTS A custom extension with the same name has already been registered.
static int	VNCSERVER_ERR_BAD_CHALLENGE VNC Data Relay could not authenticate the server.
static int	VNCSERVER_ERR_BAD_CRYPT RFB protocol or AES checksum is corrupt, or VNC Mobile Viewer did not have a matching private key.
static int	VNCSERVER_ERR_BAD_MESSAGE VNC Data Relay received an invalid message from the server.
static int	VNCSERVER_ERR_BAD_PIXEL_FORMAT VNC Mobile Viewer specified an unsupported pixel color depth.
static int	VNCSERVER_ERR_BAD_PORT Invalid port number.
static int	VNCSERVER_ERR_BAD_SESSION_ID Either the command string contained an invalid VNC Data Relay session ID, or the communication channel to which it refers is no longer reserved.
static int	VNCSERVER_ERR_BEARER_NOT_FOUND Transport mechanism specified in command string missing or corrupt.
static int	VNCSERVER_ERR_CAPTURE_FRAME_BUFFER_NOT_IMPLEMENTED Screen capture is not implemented in this platform.
static int	VNCSERVER_ERR_COMMAND_FETCH_FAILED HTTP or HTTPS request to command string web service failed.
static int	VNCSERVER_ERR_COMMAND_SUPERSEDED A command string for a different remote control session is received before the device user accepts the prompt authorizing the original session.
static int	VNCSERVER_ERR_CONNECTION_CLOSED VNC Mobile Viewer terminated the remote control session.
static int	VNCSERVER_ERR_CONNECTION_REFUSED Port could not be contacted.
static int	VNCSERVER_ERR_CRITICAL_CAPABILITY_UNSUPPORTED The connection has failed because the VNC Viewer does not support a capability which is critical to the operation of this server.
static int	VNCSERVER_ERR_DEPRECATED_FIELD_USED A deprecated field has been set.
static int	VNCSERVER_ERR_ENVIRONMENT The application environment is unsupported.
static int	VNCSERVER_ERR_FEATURE_NOT_LICENSED

	The requested operation could not be completed due to the feature not being licensed.
static int	VNCSERVER_ERR_HOST_UNREACHABLE IP address could not be contacted.
static int	VNCSERVER_ERR_INSUFFICIENT_BUFFER_SPACE The requested operation could not be completed due to insufficient buffer space.
static int	VNCSERVER_ERR_INTERNAL_ERROR General error.
static int	VNCSERVER_ERR_INVALID_COMMAND_STRING Invalid command string.
static int	VNCSERVER_ERR_INVALID_PARAMETER An invalid parameter was passed to an API call.
static int	VNCSERVER_ERR_KEY_GENERATION The RSA key generation algorithm failed.
static int	VNCSERVER_ERR_KEY_TOO_BIG The RSA key is too large.
static int	VNCSERVER_ERR_LICENSE_NOT_VALID The requested operation could not be completed due to the provided license not being valid.
static int	VNCSERVER_ERR_LOGIN_REJECTED User rejected authentication credentials.
static int	VNCSERVER_ERR_NAME_LOOKUP_FAILED Domain name could not be resolved.
static int	VNCSERVER_ERR_NETWORK General network error.
static int	VNCSERVER_ERR_NETWORK_LOST No network connection.
static int	VNCSERVER_ERR_NO_ENCODINGS VNC Mobile Viewer specified an unsupported encoding.
static int	VNCSERVER_ERR_NO_SUITABLE_RCS No suitable remote control service found.
static int	VNCSERVER_ERR_NONE No error.
static int	VNCSERVER_ERR_NOT_LICENSED_FOR_VIEWER License incompatible with that of VNC Mobile Viewer.
static int	VNCSERVER_ERR_PEER_TIMEOUT VNC Mobile Viewer did not connect to the other end of the reserved VNC Data Relay communication channel in time.
static int	VNCSERVER_ERR_PERMISSIONS Insufficient device permissions.
static int	VNCSERVER_ERR_PORT_IN_USE Port is in use.
static int	VNCSERVER_ERR_PROTOCOL_MISMATCH Protocol incompatible with that of VNC Mobile Viewer.
static int	VNCSERVER_ERR_RCS_EXITED Remote control service exited.
static int	VNCSERVER_ERR_RCS_LACKS_PERMISSIONS Remote control service does not have the required permissions.
static int	VNCSERVER_ERR_RCS_LIBRARY_NOT_FOUND Required remote control support not present.
static int	VNCSERVER_ERR_RCS_NOT_ENABLED Remote control service has not been enabled yet.
static int	VNCSERVER_ERR_RESET The server has been reset.
static int	VNCSERVER_ERR_RESOURCES Insufficient system resources.
static int	VNCSERVER_ERR_SIGNATURE_REJECTED

	VNC Mobile Viewer signature specified in command string not the same as that of the actual VNC Mobile Viewer that connects.
static int	VNCSERVER_ERR_STATE An invalid API call was made.
static int	VNCSERVER_ERR_TIMED_OUT A general network time-out occurred.
static int	VNCSERVER_ERR_TOO_LOW_ANDROID_VERSION The Android version is too low.
static int	VNCSERVER_ERR_TOO_LOW_OPENGL_ES_VERSION The OpenGL ES version is too low.
static int	VNCSERVER_ERR_TOO_MANY_EXTENSIONS The maximum number of custom extensions (8) have already been registered.
static int	VNCSERVER_ERR_TOO_MANY_EXTERNAL_ENCODERS Add external encoder had failed because the VNC Server limit on the number of external encoders is exceeded.
static int	VNCSERVER_ERR_UNABLE_TO_START_SERVICE The underlying VNC Mobile Server service could not be started.
static int	VNCSERVER_ERR_UNDERLYING_LIBRARY_NOT_FOUND Underlying Library Not Found.
static int	VNCSERVER_ERR_UNSUPPORTED_AUTH Invalid authentication type.
static int	VNCSERVER_ERR_USB_NOT_CONNECTED USB Not Connected.
static int	VNCSERVER_ERR_USER_REFUSED_CONNECTION Device user rejected prompt authorizing remote control.

Constructor Summary

Constructors
Constructor and Description
VncServerCoreErrors()

Method Summary

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

Field Detail

VNCSERVER_ERR_NONE
<pre>public static final int VNCSERVER_ERR_NONE</pre> <p>No error.</p> <p>See Also:</p> <p>Constant Field Values</p>

VNCSERVER_ERR_RESOURCES

```
public static final int VNCSERVER_ERR_RESOURCES
```

Insufficient system resources.

When the device has insufficient resources to satisfy a request then this error will be reported. For instance if the device does not have enough free memory available then an API may fail with this error code.

See Also:

[Constant Field Values](#)

VNCSERVER_ERR_STATE

```
public static final int VNCSERVER_ERR_STATE
```

An invalid API call was made.

Some API calls are only valid when the server is in a particular state. For instance it is illegal to ask the server to connect while it is already connected. This error will be reported in such circumstances.

See Also:

[Constant Field Values](#)

VNCSERVER_ERR_PERMISSIONS

```
public static final int VNCSERVER_ERR_PERMISSIONS
```

Insufficient device permissions.

VNC Mobile Server for Android can only operate if it has permission to read from the frame buffer and inject input events.

See Also:

[Constant Field Values](#)

VNCSERVER_ERR_NETWORK

```
public static final int VNCSERVER_ERR_NETWORK
```

General network error.

See Also:

[Constant Field Values](#)

VNCSERVER_ERR_HOST_UNREACHABLE

```
public static final int VNCSERVER_ERR_HOST_UNREACHABLE
```

IP address could not be contacted.

See Also:

[Constant Field Values](#)

VNCSERVER_ERR_CONNECTION_REFUSED

```
public static final int VNCSERVER_ERR_CONNECTION_REFUSED
```

Port could not be contacted.

See Also:[Constant Field Values](#)**VNCSERVER_ERR_NAME_LOOKUP_FAILED**

```
public static final int VNCSERVER_ERR_NAME_LOOKUP_FAILED
```

Domain name could not be resolved.

See Also:[Constant Field Values](#)**VNCSERVER_ERR_PORT_IN_USE**

```
public static final int VNCSERVER_ERR_PORT_IN_USE
```

Port is in use.

This error occurs when the server is told to listen on a port which is already being used by another application.

See Also:[Constant Field Values](#)**VNCSERVER_ERR_BAD_PORT**

```
public static final int VNCSERVER_ERR_BAD_PORT
```

Invalid port number.

Valid TCP port numbers range from 1 to 65535 inclusive.

See Also:[Constant Field Values](#)**VNCSERVER_ERR_NETWORK_LOST**

```
public static final int VNCSERVER_ERR_NETWORK_LOST
```

No network connection.

See Also:[Constant Field Values](#)**VNCSERVER_ERR_TIMED_OUT**

```
public static final int VNCSERVER_ERR_TIMED_OUT
```

A general network time-out occurred.

See Also:[Constant Field Values](#)**VNCSERVER_ERR_USB_NOT_CONNECTED**

```
public static final int VNCSERVER_ERR_USB_NOT_CONNECTED
```

USB Not Connected.

There is nothing connected via USB or the device is unable to communicate via USB.

See Also:

[Constant Field Values](#)

VNCSERVER_ERR_UNDERLYING_LIBRARY_NOT_FOUND

```
public static final int VNCSERVER_ERR_UNDERLYING_LIBRARY_NOT_FOUND
```

Underlying Library Not Found.

Failed to load a library for some particular functionality, for example OEM software for driving a particular type of communications.

See Also:

[Constant Field Values](#)

VNCSERVER_ERR_PROTOCOL_MISMATCH

```
public static final int VNCSERVER_ERR_PROTOCOL_MISMATCH
```

Protocol incompatible with that of VNC Mobile Viewer.

This error can occur if VNC Mobile Server is attempting to connect to VNC Viewer from a RealVNC edition other than VNC Mobile Solution, to a VNC-compatible application from a third party, or to something other than a VNC viewer (eg a HTTP server).

See Also:

[Constant Field Values](#)

VNCSERVER_ERR_LOGIN_REJECTED

```
public static final int VNCSERVER_ERR_LOGIN_REJECTED
```

User rejected authentication credentials.

See Also:

[Constant Field Values](#)

VNCSERVER_ERR_NOT_LICENSED_FOR_VIEWER

```
public static final int VNCSERVER_ERR_NOT_LICENSED_FOR_VIEWER
```

License incompatible with that of VNC Mobile Viewer.

See Also:

[Constant Field Values](#)

VNCSERVER_ERR_CONNECTION_CLOSED

```
public static final int VNCSERVER_ERR_CONNECTION_CLOSED
```

VNC Mobile Viewer terminated the remote control session.

See Also:

[Constant Field Values](#)

VNCSERVER_ERR_INVALID_COMMAND_STRING

```
public static final int VNCSERVER_ERR_INVALID_COMMAND_STRING
```

Invalid command string.

See Also:

[Constant Field Values](#)

VNCSERVER_ERR_UNSUPPORTED_AUTH

```
public static final int VNCSERVER_ERR_UNSUPPORTED_AUTH
```

Invalid authentication type.

The connection is encrypted but VNC Mobile Server did not provide RSA keys. Alternatively, VNC Mobile Viewer specified an unsupported authentication type.

See Also:

[Constant Field Values](#)

VNCSERVER_ERR_KEY_TOO_BIG

```
public static final int VNCSERVER_ERR_KEY_TOO_BIG
```

The RSA key is too large.

See Also:

[Constant Field Values](#)

VNCSERVER_ERR_BAD_CRYPT

```
public static final int VNCSERVER_ERR_BAD_CRYPT
```

RFB protocol or AES checksum is corrupt, or VNC Mobile Viewer did not have a matching private key.

See Also:

[Constant Field Values](#)

VNCSERVER_ERR_NO_ENCODINGS

```
public static final int VNCSERVER_ERR_NO_ENCODINGS
```

VNC Mobile Viewer specified an unsupported encoding.

See Also:

[Constant Field Values](#)

VNCSERVER_ERR_BAD_PIXEL_FORMAT

```
public static final int VNCSERVER_ERR_BAD_PIXEL_FORMAT
```

VNC Mobile Viewer specified an unsupported pixel color depth.

See Also:

[Constant Field Values](#)

VNCSERVER_ERR_BEARER_NOT_FOUND

```
public static final int VNCSERVER_ERR_BEARER_NOT_FOUND
```

Transport mechanism specified in command string missing or corrupt.

See Also:

[Constant Field Values](#)

VNCSERVER_ERR_SIGNATURE_REJECTED

```
public static final int VNCSERVER_ERR_SIGNATURE_REJECTED
```

VNC Mobile Viewer signature specified in command string not the same as that of the actual VNC Mobile Viewer that connects.

See Also:

[Constant Field Values](#)

VNCSERVER_ERR_INSUFFICIENT_BUFFER_SPACE

```
public static final int VNCSERVER_ERR_INSUFFICIENT_BUFFER_SPACE
```

The requested operation could not be completed due to insufficient buffer space.

See Also:

[Constant Field Values](#)

VNCSERVER_ERR_LICENSE_NOT_VALID

```
public static final int VNCSERVER_ERR_LICENSE_NOT_VALID
```

The requested operation could not be completed due to the provided license not being valid.

See Also:

[Constant Field Values](#)

VNCSERVER_ERR_FEATURE_NOT_LICENSED

```
public static final int VNCSERVER_ERR_FEATURE_NOT_LICENSED
```

The requested operation could not be completed due to the feature not being licensed.

See Also:

[Constant Field Values](#)

VNCSERVER_ERR_CRITICAL_CAPABILITY_UNSUPPORTED

```
public static final int VNCSERVER_ERR_CRITICAL_CAPABILITY_UNSUPPORTED
```

The connection has failed because the VNC Viewer does not support a capability which is critical to the operation of this server.

This can occur, for instance, when attempting to connect a view-only server to a viewer which is not aware of the existence of view-only servers.

See Also:

[Constant Field Values](#)

VNCSERVER_ERR_TOO_MANY_EXTERNAL_ENCODERS

```
public static final int VNCSERVER_ERR_TOO_MANY_EXTERNAL_ENCODERS
```

Add external encoder had failed because the VNC Server limit on the number of external encoders is exceeded.

See Also:

[Constant Field Values](#)

VNCSERVER_ERR_INVALID_PARAMETER

```
public static final int VNCSERVER_ERR_INVALID_PARAMETER
```

An invalid parameter was passed to an API call.

This can occur when registering a custom extension with an invalid name, or sending an extension message with an invalid length.

See Also:

[Constant Field Values](#)

VNCSERVER_ERR_KEY_GENERATION

```
public static final int VNCSERVER_ERR_KEY_GENERATION
```

The RSA key generation algorithm failed.

See Also:

[Constant Field Values](#)

VNCSERVER_ERR_UNABLE_TO_START_SERVICE

```
public static final int VNCSERVER_ERR_UNABLE_TO_START_SERVICE
```

The underlying VNC Mobile Server service could not be started.

See Also:

[Constant Field Values](#)

VNCSERVER_ERR_ALREADY_EXISTS

```
public static final int VNCSERVER_ERR_ALREADY_EXISTS
```

A custom extension with the same name has already been registered.

See Also:

[Constant Field Values](#)

VNCSERVER_ERR_TOO_MANY_EXTENSIONS

```
public static final int VNCSERVER_ERR_TOO_MANY_EXTENSIONS
```

The maximum number of custom extensions (8) have already been registered.

See Also:

[Constant Field Values](#)

VNCSERVER_ERR_RESET

```
public static final int VNCSERVER_ERR_RESET
```

The server has been reset.

See Also:

[Constant Field Values](#)

VNCSERVER_ERR_DEPRECATED_FIELD_USED

```
public static final int VNCSERVER_ERR_DEPRECATED_FIELD_USED
```

A deprecated field has been set.

See Also:

[Constant Field Values](#)

VNCSERVER_ERR_BAD_MESSAGE

```
public static final int VNCSERVER_ERR_BAD_MESSAGE
```

VNC Data Relay received an invalid message from the server.

See Also:

[Constant Field Values](#)

VNCSERVER_ERR_BAD_SESSION_ID

```
public static final int VNCSERVER_ERR_BAD_SESSION_ID
```

Either the command string contained an invalid VNC Data Relay session ID, or the communication channel to which it refers is no longer reserved.

See Also:

[Constant Field Values](#)

VNCSERVER_ERR_BAD_CHALLENGE

```
public static final int VNCSERVER_ERR_BAD_CHALLENGE
```

VNC Data Relay could not authenticate the server.

See Also:

[Constant Field Values](#)

VNCSERVER_ERR_PEER_TIMEOUT

```
public static final int VNCSERVER_ERR_PEER_TIMEOUT
```

VNC Mobile Viewer did not connect to the other end of the reserved VNC Data Relay communication channel in time.

See Also:

[Constant Field Values](#)

VNCSERVER_ERR_USER_REFUSED_CONNECTION

```
public static final int VNCSERVER_ERR_USER_REFUSED_CONNECTION
```

Device user rejected prompt authorizing remote control.

See Also:

[Constant Field Values](#)

VNCSERVER_ERR_COMMAND_FETCH_FAILED

```
public static final int VNCSERVER_ERR_COMMAND_FETCH_FAILED
```

HTTP or HTTPS request to command string web service failed.

See Also:

[Constant Field Values](#)

VNCSERVER_ERR_INTERNAL_ERROR

```
public static final int VNCSERVER_ERR_INTERNAL_ERROR
```

General error.

See Also:

[Constant Field Values](#)

VNCSERVER_ERR_COMMAND_SUPERSEDED

```
public static final int VNCSERVER_ERR_COMMAND_SUPERSEDED
```

A command string for a different remote control session is received before the device user accepts the prompt authorizing the original session.

See Also:

[Constant Field Values](#)

VNCSERVER_ERR_ENVIRONMENT

```
public static final int VNCSERVER_ERR_ENVIRONMENT
```

The application environment is unsupported.

See Also:

[Constant Field Values](#)

VNCSERVER_ERR_CAPTURE_FRAME_BUFFER_NOT_IMPLEMENTED

```
public static final int VNCSERVER_ERR_CAPTURE_FRAME_BUFFER_NOT_IMPLEMENTED
```

Screen capture is not implemented in this platform.

See Also:

[Constant Field Values](#)

VNCSERVER_ERR_RCS_LIBRARY_NOT_FOUND

```
public static final int VNCSERVER_ERR_RCS_LIBRARY_NOT_FOUND
```

Required remote control support not present.

See Also:

[Constant Field Values](#)

VNCSERVER_ERR_NO_SUITABLE_RCS

```
public static final int VNCSERVER_ERR_NO_SUITABLE_RCS
```

No suitable remote control service found.

See Also:

[Constant Field Values](#)

VNCSERVER_ERR_RCS_LACKS_PERMISSIONS

```
public static final int VNCSERVER_ERR_RCS_LACKS_PERMISSIONS
```

Remote control service does not have the required permissions.

See Also:

[Constant Field Values](#)

VNCSERVER_ERR_RCS_NOT_ENABLED

```
public static final int VNCSERVER_ERR_RCS_NOT_ENABLED
```

Remote control service has not been enabled yet.

See Also:

[Constant Field Values](#)

VNCSERVER_ERR_RCS_EXITED

```
public static final int VNCSERVER_ERR_RCS_EXITED
```

Remote control service exited.

See Also:

[Constant Field Values](#)

VNCSERVER_ERR_TOO_LOW_ANDROID_VERSION

```
public static final int VNCSERVER_ERR_TOO_LOW_ANDROID_VERSION
```

The Android version is too low.

See Also:

[Constant Field Values](#)

VNCSERVER_ERR_TOO_LOW_OPENGL_ES_VERSION

```
public static final int VNCSERVER_ERR_TOO_LOW_OPENGL_ES_VERSION
```

The OpenGL ES version is too low.

See Also:

[Constant Field Values](#)

Constructor Detail**VncServerCoreErrors**

```
public VncServerCoreErrors()
```

com.realvnc.vncserver.core

Class VncServerState

java.lang.Object

com.realvnc.vncserver.core.VncServerState

```
public final class VncServerState
extends java.lang.Object
```

Constants representing the various states that the VNC server can be in.

Field Summary

Fields

Modifier and Type	Field and Description
static int	VNC_STATE_ACCEPT_REMOTE_KEY Server is waiting for a remote key to be accepted
static int	VNC_STATE_ACCEPTING Server is waiting for a connection to be accepted
static int	VNC_STATE_AUTH Server is waiting for viewer credentials to be authenticated by the application.
static int	VNC_STATE_AWAITING_KEY Server is waiting for an encryption key to be set
static int	VNC_STATE_CONNECTING Server is initiating an outbound connection
static int	VNC_STATE_CONNECTING_RELAY Server is performing a data relay handshake
static int	VNC_STATE_DISCONNECTED Server is idle
static int	VNC_STATE_EXITING Server is in the process of exiting
static int	VNC_STATE_GENERATING_KEY Server is generating an encryption key
static int	VNC_STATE_HANDSHAKING Server is processing the RFB handshaking phase
static int	VNC_STATE_LISTENING Server is listening for an incoming connection
static int	VNC_STATE_ML_AWAITING_CLIENT_DISPLAY_CONFIGURATION Server is waiting for a MirrorLink 'client display configuration' message from the viewer.
static int	VNC_STATE_ML_AWAITING_CLIENT_EVENT_CONFIGURATION Server is waiting for a MirrorLink 'client event configuration' message from the viewer.
static int	VNC_STATE_ML_AWAITING_SERVER_DISPLAY_CONFIGURATION Server is waiting for a MirrorLink 'server display configuration' message from the application.
static int	VNC_STATE_ML_AWAITING_SERVER_EVENT_CONFIGURATION Server is waiting for a MirrorLink 'server event configuration' message from the application.
static int	VNC_STATE_REVERSE_AUTH Server is waiting for a reverse authentication password from the application.
static int	VNC_STATE_RUNNING Server is connected to a viewer
static int	VNC_STATE_SETUP Server is setting the parameters for the RFB session

Method Summary

Methods inherited from class java.lang.Object

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

Field Detail

VNC_STATE_DISCONNECTED

```
public static final int VNC_STATE_DISCONNECTED
```

Server is idle

See Also:

[Constant Field Values](#)

VNC_STATE_AWAITING_KEY

```
public static final int VNC_STATE_AWAITING_KEY
```

Server is waiting for an encryption key to be set

See Also:

[Constant Field Values](#)

VNC_STATE_GENERATING_KEY

```
public static final int VNC_STATE_GENERATING_KEY
```

Server is generating an encryption key

See Also:

[Constant Field Values](#)

VNC_STATE_LISTENING

```
public static final int VNC_STATE_LISTENING
```

Server is listening for an incoming connection

See Also:

[Constant Field Values](#)

VNC_STATE_CONNECTING

```
public static final int VNC_STATE_CONNECTING
```

Server is initiating an outbound connection

See Also:

[Constant Field Values](#)

VNC_STATE_CONNECTING_RELAY

```
public static final int VNC_STATE_CONNECTING_RELAY
```

Server is performing a data relay handshake

See Also:

[Constant Field Values](#)

VNC_STATE_ACCEPTING

```
public static final int VNC_STATE_ACCEPTING
```

Server is waiting for a connection to be accepted

See Also:

[Constant Field Values](#)

VNC_STATE_ACCEPT_REMOTE_KEY

```
public static final int VNC_STATE_ACCEPT_REMOTE_KEY
```

Server is waiting for a remote key to be accepted

See Also:

[Constant Field Values](#)

VNC_STATE_AUTH

```
public static final int VNC_STATE_AUTH
```

Server is waiting for viewer credentials to be authenticated by the application.

See Also:

[Constant Field Values](#)

VNC_STATE_REVERSE_AUTH

```
public static final int VNC_STATE_REVERSE_AUTH
```

Server is waiting for a reverse authentication password from the application.

See Also:

[Constant Field Values](#)

VNC_STATE_SETUP

```
public static final int VNC_STATE_SETUP
```

Server is setting the parameters for the RFB session

See Also:

[Constant Field Values](#)

VNC_STATE_HANDSHAKING

```
public static final int VNC_STATE_HANDSHAKING
```

Server is processing the RFB handshaking phase

See Also:

[Constant Field Values](#)

VNC_STATE_RUNNING

```
public static final int VNC_STATE_RUNNING
```

Server is connected to a viewer

See Also:

[Constant Field Values](#)

VNC_STATE_ML_AWAITING_SERVER_DISPLAY_CONFIGURATION

```
public static final int VNC_STATE_ML_AWAITING_SERVER_DISPLAY_CONFIGURATION
```

Server is waiting for a MirrorLink 'server display configuration' message from the application.

See Also:

[Constant Field Values](#)

VNC_STATE_ML_AWAITING_CLIENT_DISPLAY_CONFIGURATION

```
public static final int VNC_STATE_ML_AWAITING_CLIENT_DISPLAY_CONFIGURATION
```

Server is waiting for a MirrorLink 'client display configuration' message from the viewer.

See Also:

[Constant Field Values](#)

VNC_STATE_ML_AWAITING_SERVER_EVENT_CONFIGURATION

```
public static final int VNC_STATE_ML_AWAITING_SERVER_EVENT_CONFIGURATION
```

Server is waiting for a MirrorLink 'server event configuration' message from the application.

See Also:

[Constant Field Values](#)

VNC_STATE_ML_AWAITING_CLIENT_EVENT_CONFIGURATION

```
public static final int VNC_STATE_ML_AWAITING_CLIENT_EVENT_CONFIGURATION
```

Server is waiting for a MirrorLink 'client event configuration' message from the viewer.

See Also:

[Constant Field Values](#)

VNC_STATE_EXITING

```
public static final int VNC_STATE_EXITING
```

Server is in the process of exiting

See Also:

[Constant Field Values](#)

How This API Document Is Organized

This API (Application Programming Interface) document has pages corresponding to the items in the navigation bar, described as follows.

Overview

The [Overview](#) page is the front page of this API document and provides a list of all packages with a summary for each. This page can also contain an overall description of the set of packages.

Package

Each package has a page that contains a list of its classes and interfaces, with a summary for each. This page can contain six categories:

- Interfaces (*italic*)
- Classes
- Enums
- Exceptions
- Errors
- Annotation Types

Class/Interface

Each class, interface, nested class and nested interface has its own separate page. Each of these pages has three sections consisting of a class/interface description, summary tables, and detailed member descriptions:

- Class inheritance diagram
- Direct Subclasses
- All Known Subinterfaces
- All Known Implementing Classes
- Class/interface declaration
- Class/interface description
- Nested Class Summary
- Field Summary
- Constructor Summary
- Method Summary
- Field Detail
- Constructor Detail
- Method Detail

Each summary entry contains the first sentence from the detailed description for that item. The summary entries are alphabetical, while the detailed descriptions are in the order they appear in the source code. This preserves the logical groupings established by the programmer.

Annotation Type

Each annotation type has its own separate page with the following sections:

- Annotation Type declaration
- Annotation Type description
- Required Element Summary
- Optional Element Summary
- Element Detail

Enum

Each enum has its own separate page with the following sections:

- Enum declaration
- Enum description
- Enum Constant Summary
- Enum Constant Detail

Tree (Class Hierarchy)

There is a [Class Hierarchy](#) page for all packages, plus a hierarchy for each package. Each hierarchy page contains a list of classes and a list of interfaces. The classes are organized by inheritance structure starting with `java.lang.Object`. The interfaces do not inherit from `java.lang.Object`.

- When viewing the Overview page, clicking on "Tree" displays the hierarchy for all packages.
- When viewing a particular package, class or interface page, clicking "Tree" displays the hierarchy for only that package.

Deprecated API

The [Deprecated API](#) page lists all of the API that have been deprecated. A deprecated API is not recommended for use, generally due to improvements, and a replacement API is usually given. Deprecated APIs may be removed in future implementations.

Index

The [Index](#) contains an alphabetic list of all classes, interfaces, constructors, methods, and fields.

Prev/Next

These links take you to the next or previous class, interface, package, or related page.

Frames/No Frames

These links show and hide the HTML frames. All pages are available with or without frames.

All Classes

The [All Classes](#) link shows all classes and interfaces except non-static nested types.

Serialized Form

Each serializable or externalizable class has a description of its serialization fields and methods. This information is of interest to re-implementors, not to developers using the API. While there is no link in the navigation bar, you can get to this information by going to any serialized class and clicking "Serialized Form" in the "See also" section of the class description.

Constant Field Values

The [Constant Field Values](#) page lists the static final fields and their values.

This help file applies to API documentation generated using the standard doclet.

A B C D E F G H I K L M O P Q R S T U V X

A**accept(boolean)** - Method in class `com.realvnc.vncserver.android.VncServer`

Accept or reject a connection from a VNC viewer.

acceptRemoteKey(boolean) - Method in class `com.realvnc.vncserver.android.VncServer`

Accept or reject an RSA key from a VNC viewer.

accessibilityServiceRequired() - Method in interface`com.realvnc.vncserver.android.VncContextInformationManager.AccessibilityServiceProvider`

Called when the accessibility service is required to receive context information.

addAccessibilityServiceProvider(VncContextInformationManager.AccessibilityServiceProvider) - Method in class`com.realvnc.vncserver.android.VncContextInformationManager`

Add a new accessibility service provider.

addBearer(String, String) - Method in class `com.realvnc.vncserver.android.VncServer`

Register a new bearer with the SDK.

addBearer(String, Class<? extends VncBearer>) - Method in class `com.realvnc.vncserver.android.VncServer`

Register a new bearer with the SDK.

addLicense(String) - Method in class `com.realvnc.vncserver.android.VncServer`

Adds a license to the server.

addLicenseFeature(int, byte[]) - Method in class `com.realvnc.vncserver.android.VncServer`

Adds knowledge of a feature to the server.

addListener(VncContextInformationManager.Listener) - Method in class `com.realvnc.vncserver.android.VncContextInformationManager`

Add a new listener to receive notification of changes to context information.

addListener(VncContextInformationManager.Listener, VncContextInformationManager.ListenerPriority) - Method in class`com.realvnc.vncserver.android.VncContextInformationManager`

Add a new listener to receive notification of changes to context information.

addListener(VncDisplayInformationManager.Listener) - Method in class `com.realvnc.vncserver.android.VncDisplayInformationManager`

Add a new listener to receive notification of changes to the display information.

addRemoteFeatureCheck(int[], VncRemoteFeatureCheckListener) - Method in class `com.realvnc.vncserver.android.VncServer`

Adds a remote feature check to this server instance.

allSections() - Method in class `com.realvnc.util.JniFile`**APPLICATION_CATEGORY_BROWSER** - Static variable in class `com.realvnc.mirrorlink.VNCContextInformation`

General browser category.

APPLICATION_CATEGORY_BROWSER_APPLICATION_STORE - Static variable in class `com.realvnc.mirrorlink.VNCContextInformation`

Category representing an application store.

APPLICATION_CATEGORY_IMMERSIVE_HOME_SCREEN - Static variable in class `com.realvnc.mirrorlink.VNCContextInformation`

Category representing an immersive home screen.

APPLICATION_CATEGORY_INFORMATION - Static variable in class `com.realvnc.mirrorlink.VNCContextInformation`

General information category.

APPLICATION_CATEGORY_INFORMATION_CLOCK - Static variable in class `com.realvnc.mirrorlink.VNCContextInformation`

Category representing a clock application.

APPLICATION_CATEGORY_INFORMATION_NEWS - Static variable in class `com.realvnc.mirrorlink.VNCContextInformation`

Category representing a news information application.

APPLICATION_CATEGORY_INFORMATION_SPORTS - Static variable in class `com.realvnc.mirrorlink.VNCContextInformation`

Category representing a sports information application.

APPLICATION_CATEGORY_INFORMATION_STOCKS - Static variable in class `com.realvnc.mirrorlink.VNCContextInformation`

Category representing a stocks information application.

APPLICATION_CATEGORY_INFORMATION_TRAVEL - Static variable in class `com.realvnc.mirrorlink.VNCContextInformation`

Category representing a travel information application.

APPLICATION_CATEGORY_INFORMATION_WEATHER - Static variable in class `com.realvnc.mirrorlink.VNCContextInformation`

Category representing a weather information application.

APPLICATION_CATEGORY_MASK - Static variable in class `com.realvnc.mirrorlink.VNCContextInformation`

Used to mask out the category from the subcategory for testing.

APPLICATION_CATEGORY_MEDIA - Static variable in class `com.realvnc.mirrorlink.VNCContextInformation`

General media applications category.

APPLICATION_CATEGORY_MEDIA_GAMING - Static variable in class `com.realvnc.mirrorlink.VNCContextInformation`

Category representing a game or gaming related application.

APPLICATION_CATEGORY_MEDIA_IMAGE - Static variable in class com.realvnc.mirrorlink.VNCContextInformation
Category representing an image application.

APPLICATION_CATEGORY_MEDIA_MUSIC - Static variable in class com.realvnc.mirrorlink.VNCContextInformation
Category representing a music application.

APPLICATION_CATEGORY_MEDIA_VIDEO - Static variable in class com.realvnc.mirrorlink.VNCContextInformation
Category representing a video.

APPLICATION_CATEGORY_MESSAGING - Static variable in class com.realvnc.mirrorlink.VNCContextInformation
General messaging applications category.

APPLICATION_CATEGORY_MESSAGING_EMAIL - Static variable in class com.realvnc.mirrorlink.VNCContextInformation
Category representing an email.

APPLICATION_CATEGORY_MESSAGING_MMS - Static variable in class com.realvnc.mirrorlink.VNCContextInformation
Category representing an MMS.

APPLICATION_CATEGORY_MESSAGING_SMS - Static variable in class com.realvnc.mirrorlink.VNCContextInformation
Category representing an SMS.

APPLICATION_CATEGORY_NAVIGATION - Static variable in class com.realvnc.mirrorlink.VNCContextInformation
General navigation category.

APPLICATION_CATEGORY_NO_UI - Static variable in class com.realvnc.mirrorlink.VNCContextInformation
General UI-less applications category.

APPLICATION_CATEGORY_NO_UI_CLIENT - Static variable in class com.realvnc.mirrorlink.VNCContextInformation
Category representing a client.

APPLICATION_CATEGORY_NO_UI_CONVERSATIONAL_AUDIO - Static variable in class com.realvnc.mirrorlink.VNCContextInformation
Category representing conversational audio.

APPLICATION_CATEGORY_NO_UI_SERVER - Static variable in class com.realvnc.mirrorlink.VNCContextInformation
Category representing a server.

APPLICATION_CATEGORY_NO_UI_VOICE_COMMAND_ENGINE - Static variable in class com.realvnc.mirrorlink.VNCContextInformation
Category representing a voice command engine.

APPLICATION_CATEGORY_PHONE - Static variable in class com.realvnc.mirrorlink.VNCContextInformation
General phone call application category.

APPLICATION_CATEGORY_PHONE_CALL_LOG - Static variable in class com.realvnc.mirrorlink.VNCContextInformation
Category representing a call log.

APPLICATION_CATEGORY_PHONE_CONTACT_LIST - Static variable in class com.realvnc.mirrorlink.VNCContextInformation
Category representing a contact list.

APPLICATION_CATEGORY_PHONE_IMMERSIVE_CALL - Static variable in class com.realvnc.mirrorlink.VNCContextInformation
Category representing an immersive phone call.

APPLICATION_CATEGORY_PIM - Static variable in class com.realvnc.mirrorlink.VNCContextInformation
General Personal Information Management category.

APPLICATION_CATEGORY_PIM_CALENDAR - Static variable in class com.realvnc.mirrorlink.VNCContextInformation
Category representing a calendar application.

APPLICATION_CATEGORY_PIM_NOTES - Static variable in class com.realvnc.mirrorlink.VNCContextInformation
Category representing a notes application.

APPLICATION_CATEGORY_PRODUCTIVITY - Static variable in class com.realvnc.mirrorlink.VNCContextInformation
General productivity category.

APPLICATION_CATEGORY_PRODUCTIVITY_DOCUMENT_EDITOR - Static variable in class com.realvnc.mirrorlink.VNCContextInformation
Category representing a document editor.

APPLICATION_CATEGORY_PRODUCTIVITY_DOCUMENT_VIEWER - Static variable in class com.realvnc.mirrorlink.VNCContextInformation
Category representing a document viewer.

APPLICATION_CATEGORY_SOCIAL_NETWORKING - Static variable in class com.realvnc.mirrorlink.VNCContextInformation
General social networking category.

APPLICATION_CATEGORY_SWITCH_TO_CLIENT_NATIVE_UI - Static variable in class com.realvnc.mirrorlink.VNCContextInformation
Category used within the context information to tell the client is should switch to the native UI, or disconnect the VNC session.

APPLICATION_CATEGORY_SYSTEM - Static variable in class com.realvnc.mirrorlink.VNCContextInformation
General system category.

APPLICATION_CATEGORY_SYSTEM_INPUT_BLUETOOTH_PIN - Static variable in class com.realvnc.mirrorlink.VNCContextInformation
Category representing a Bluetooth PIN code input.

APPLICATION_CATEGORY_SYSTEM_INPUT_OTHER_PASSWORD - Static variable in class com.realvnc.mirrorlink.VNCContextInformation
Category representing a password input.

APPLICATION_CATEGORY_SYSTEM_INPUT_UNLOCK_PIN - Static variable in class com.realvnc.mirrorlink.VNCContextInformation
Category representing a PIN input for device unlock.

APPLICATION_CATEGORY_SYSTEM_VOICE_COMMAND_CONFIRMATION - Static variable in class

`com.realvnc.mirrorlink.VNCContextInformation`

Category representing a voice command confirmation.

APPLICATION_CATEGORY_TESTING_AND_CERTIFICATION - Static variable in class `com.realvnc.mirrorlink.VNCContextInformation`

General testing and certification category.

APPLICATION_CATEGORY_UI - Static variable in class `com.realvnc.mirrorlink.VNCContextInformation`

General UI framework category.

APPLICATION_CATEGORY_UI_APPLICATION_LISTING - Static variable in class `com.realvnc.mirrorlink.VNCContextInformation`

Category representing an application listing.

APPLICATION_CATEGORY_UI_HOME_SCREEN - Static variable in class `com.realvnc.mirrorlink.VNCContextInformation`

Category representing a home screen / start-up screen.

APPLICATION_CATEGORY_UI_MENU - Static variable in class `com.realvnc.mirrorlink.VNCContextInformation`

Category representing a menu.

APPLICATION_CATEGORY_UI_NOTIFICATION - Static variable in class `com.realvnc.mirrorlink.VNCContextInformation`

Category representing a notification.

APPLICATION_CATEGORY_UI_SETTINGS - Static variable in class `com.realvnc.mirrorlink.VNCContextInformation`

Category representing a settings application.

APPLICATION_CATEGORY_UNKNOWN - Static variable in class `com.realvnc.mirrorlink.VNCContextInformation`

The server has no information about the application category.

AUDIO_CONTENT_CATEGORY_MEDIA_AUDIO_IN - Static variable in class `com.realvnc.mirrorlink.VNCAudioInfo`

Audio content category representing media input.

AUDIO_CONTENT_CATEGORY_MEDIA_AUDIO_OUT - Static variable in class `com.realvnc.mirrorlink.VNCAudioInfo`

Audio content category representing media output.

AUDIO_CONTENT_CATEGORY_MISC - Static variable in class `com.realvnc.mirrorlink.VNCAudioInfo`

Audio content category representing miscellaneous audio.

AUDIO_CONTENT_CATEGORY_PHONE_AUDIO - Static variable in class `com.realvnc.mirrorlink.VNCAudioInfo`

Audio content category representing phone audio.

AUDIO_CONTENT_CATEGORY_UNKNOWN - Static variable in class `com.realvnc.mirrorlink.VNCAudioInfo`

Audio content category representing unknown content.

AUDIO_CONTENT_CATEGORY_VOICE_COMMAND_IN - Static variable in class `com.realvnc.mirrorlink.VNCAudioInfo`

Audio content category representing voice command input.

AUDIO_CONTENT_CATEGORY_VOICE_COMMAND_OUT - Static variable in class `com.realvnc.mirrorlink.VNCAudioInfo`

Audio content category representing voice command output.

authCb(VncServer, String, String) - Method in interface `com.realvnc.vncserver.android.VncServerListener`

Callback invoked when the viewer has provided some authentication details.

authenticate(boolean) - Method in class `com.realvnc.vncserver.android.VncServer`

Accept (or reject) a connection based on authentication credentials supplied to `authCb`.

B

bell() - Method in class `com.realvnc.vncserver.android.VncServer`

Sends a bell message to the client.

blacklistRemoteControl(VncRemoteControlInfo, boolean) - Method in class `com.realvnc.vncserver.android.VncServer`

Prevents a particular type of remote control method from being used, or reinstates it.

C

cause - Variable in exception `com.realvnc.vncserver.core.VncException`

CHANGE_FLAG_ESTIMATED - Static variable in class `com.realvnc.vncserver.android.VncContextInformationManager`

Constant flag used in `VncContextInformationManager.Listener.contextInformationChanged(java.util.List, int)` to indicate that the context information provided in the callback is only an estimate of the pixel contents of the screen.

CHANGE_FLAG_POLLED - Static variable in class `com.realvnc.vncserver.android.VncContextInformationManager`

Constant flag used in `VncContextInformationManager.Listener.contextInformationChanged(java.util.List, int)` to indicate that the context information is being polled and may not correspond to the exact pixel data being sent to the viewer.

CHANGE_FLAG_SYNCHRONOUS - Static variable in class `com.realvnc.vncserver.android.VncContextInformationManager`

Constant flag used in `VncContextInformationManager.Listener.contextInformationChanged(java.util.List, int)` to indicate that the pixel data for the captured context information won't be sent until the callback has returned.

checkRemoteControlAvailable() - Method in class `com.realvnc.vncserver.android.VncServer`

Check whether the device has support for remote control.

CLASS_BUTTON_BAR - Static variable in class `com.realvnc.vncserver.android.VncContextInformationManager`

Constant used as the class name for the button bar softkeys on the display.

CLASS_KEYGUARD - Static variable in class `com.realvnc.vncserver.android.VncContextInformationManager`

Constant used as the class name for the keyguard component.

CLASS_STATUS_BAR - Static variable in class `com.realvnc.vncserver.android.VncContextInformationManager`

Constant used as the class name for the status bar, which contains notification.

CLASS_TOAST - Static variable in class `com.realvnc.vncserver.android.VncContextInformationManager`

Constant used as the class name for toast messages provided by `android.widget.Toast`.

close() - Method in interface `com.realvnc.vncserver.core.VncConnection`

If the connection is already established then close it, or if we're still trying to establish a connection give up.

com.realvnc.mirrorlink - package `com.realvnc.mirrorlink`

Provides the classes describing various aspects of a MirrorLink connection.

com.realvnc.util - package `com.realvnc.util`

Provides some utility classes useful in the creation of a server.

com.realvnc.vncserver.android - package `com.realvnc.vncserver.android`

Provides the main classes for use in managing a server instance.

com.realvnc.vncserver.core - package `com.realvnc.vncserver.core`

Provides the core classes which are independent of Android.

connect(String, int) - Method in class `com.realvnc.vncserver.android.VncServer`

Establish a socket connection to a listening viewer using the VNC TCP/IP outbound pluggable bearer.

connect(VncCommandString) - Method in class `com.realvnc.vncserver.android.VncServer`

Establish a connection using the connection details and bearer specified in the command string.

connectedCb(VncServer, String) - Method in interface `com.realvnc.vncserver.android.VncServerListener`

Called when a viewer (or data relay) has been connected to us, or we have connected to a viewer.

connectingCb(VncServer) - Method in interface `com.realvnc.vncserver.android.VncServerListener`

Callback to indicate that the VNC server is connecting to a remote viewer.

CONTEXT_FLAG_SYSTEM_UI - Static variable in class `com.realvnc.vncserver.android.VncContextInformationManager`

Constant flag used in `VncContextInformationManager.CapturedContextInformation.getFlags()` to indicate that the context rectangle is from a system UI element.

contextInformationChanged(List<VncContextInformationManager.CapturedContextInformation>, int) - Method in interface `com.realvnc.vncserver.android.VncContextInformationManager.Listener`

Called when the current context information has changed.

create(Context, VncServerListener) - Static method in class `com.realvnc.vncserver.android.VncServer`

Construct a new VNC server.

create(Context, VncServerListener, Handler) - Static method in class `com.realvnc.vncserver.android.VncServer`

Construct a new VNC server.

create(Context, VncServerOrientationListener) - Static method in class `com.realvnc.vncserver.android.VncServer`

Construct a new VNC server.

create(Context, VncServerOrientationListener, Handler) - Static method in class `com.realvnc.vncserver.android.VncServer`

Construct a new VNC server.

create(Context, VncServerOrientationListener, VncServerMirrorLinkListener) - Static method in class `com.realvnc.vncserver.android.VncServer`

Construct a new VNC server.

create(Context, VncServerOrientationListener, VncServerMirrorLinkListener, Handler) - Static method in class `com.realvnc.vncserver.android.VncServer`

Construct a new VNC server.

create(Context, MirrorLinkCallbackHandler) - Static method in class `com.realvnc.vncserver.android.VncServer`

Construct a new VNC server.

create(Context, MirrorLinkCallbackHandler, Handler) - Static method in class `com.realvnc.vncserver.android.VncServer`

Construct a new VNC server.

create(Context, VncServerCallbackHandler) - Static method in class `com.realvnc.vncserver.android.VncServer`

Construct a new VNC server.

create(Context, VncServerCallbackHandler, Handler) - Static method in class `com.realvnc.vncserver.android.VncServer`

Construct a new VNC server.

createConnection(VncCommandStringBase, VncBearerCallbacks) - Method in interface `com.realvnc.vncserver.core.VncBearer`

Create a new connection object which can be used to establish a new transport session over this bearer.

customRemoteControlServiceCb(String, Bundle) - Method in class `com.realvnc.vncserver.android.VncServerCallbackHandler`

A callback indicating an asynchronous reply to a custom request sent to the Remote Control Service.

customRemoteControlServiceRequest(String, Bundle) - Method in class `com.realvnc.vncserver.android.VncServer`

Sends a custom request to the remote control service.

CustomRemoteControlServiceRequests - Class in `com.realvnc.vncserver.android`

Custom requests supported by Remote Control Service implementations provided by RealVNC for Android platforms.

CustomRemoteControlServiceRequests() - Constructor for class `com.realvnc.vncserver.android.CustomRemoteControlServiceRequests`

D

decodeBase64(String) - Method in class `com.realvnc.vncserver.android.VncCommandString`

Take a base 64 encoded string and return the decoded output.

decodeBase64(String) - Method in class `com.realvnc.vncserver.core.VncCommandStringBase`

Take a base 64 encoded string and return the decoded output.

destroy() - Static method in class `com.realvnc.util.VncLog`

destroy() - Method in class `com.realvnc.vncserver.android.VncServer`

Destroy the the server object and release all associated resources.

DEVICE_KEY_SUPPORT_ALL - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

DEVICE_KEY_SUPPORT_APPLICATION - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

DEVICE_KEY_SUPPORT_BACKWARD - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

DEVICE_KEY_SUPPORT_CLEAR - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

DEVICE_KEY_SUPPORT_DELETE - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

DEVICE_KEY_SUPPORT_FORWARD - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

DEVICE_KEY_SUPPORT_HOME - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

DEVICE_KEY_SUPPORT_MENU - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

DEVICE_KEY_SUPPORT_OK - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

DEVICE_KEY_SUPPORT_PHONE_CALL - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

DEVICE_KEY_SUPPORT_PHONE_END - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

DEVICE_KEY_SUPPORT_SEARCH - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

DEVICE_KEY_SUPPORT_SOFT_LEFT - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

DEVICE_KEY_SUPPORT_SOFT_MIDDLE - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

DEVICE_KEY_SUPPORT_SOFT_RIGHT - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

DEVICE_KEY_SUPPORT_ZOOM_IN - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

DEVICE_KEY_SUPPORT_ZOOM_OUT - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

deviceKeySupport - Variable in class `com.realvnc.mirrorlink.VNCServerEventConfiguration`

Contains a bitmask of device key support.

disconnectedCb(VncServer) - Method in interface `com.realvnc.vncserver.android.VncServerListener`

Called when the viewer has disconnected from us, or we have disconnected from the viewer.

DisplayConfiguration - Class in `com.realvnc.mirrorlink`

Class defining constants for use in `VNCServerDisplayConfiguration` and `VNCClientDisplayConfiguration` classes.

DisplayConfiguration() - Constructor for class `com.realvnc.mirrorlink.DisplayConfiguration`

displayOrientationChangedCb(int) - Method in interface `com.realvnc.vncserver.android.VncServerOrientationListener`

Called when the orientation of the VNC server framebuffer has changed.

displayOrientationChangeNeededCb(int) - Method in interface `com.realvnc.vncserver.android.VncServerOrientationListener`

Called when the SDK requires the application to change the current display orientation.

E

ENABLE_HEADS_UP_NOTIFICATIONS - Static variable in class `com.realvnc.vncserver.android.CustomRemoteControlServiceRequests`

Globally enables/disables Android heads up notifications.

ENABLE_REMOTE_CONTROL - Static variable in class `com.realvnc.vncserver.android.CustomRemoteControlServiceRequests`

Enables remote control.

enableFeature(int, boolean) - Method in class `com.realvnc.vncserver.android.VncServer`

Selectively enable or disable optional features.

encodeFrame(VncH264Encoder.ScreenGrabHelper, VncH264Encoder.BufferOwner, int) - Method in class `com.realvnc.vncserver.android.VncH264Encoder`

Encode the contents of the input surface to a ByteBuffer.

equals(Object) - Method in class `com.realvnc.vncserver.android.VncSizeInt`

equals(Object) - Method in class `com.realvnc.vncserver.core.VncPixelFormat`

Tests if this VncPixelFormat instance is equal to the given Object.

errorCb(VncServer, int, Exception) - Method in interface `com.realvnc.vncserver.android.VncServerListener`

Called when an unexpected error occurred to indicate that the server has returned to the disconnected state and is no longer listening for a connection, connecting, or connected.

errorCode - Variable in exception `com.realvnc.vncserver.core.VncException`

establish() - Method in interface `com.realvnc.vncserver.core.VncConnection`

Establish a connection over the bearer using the connection details in the command string passed to the `createConnection` method of the object implementing the `VncBearer` interface.

EventConfiguration - Class in `com.realvnc.mirrorlink`

Class defining constants for use in `VNCServerEventConfiguration` and `VNCClientEventConfiguration` classes.

EventConfiguration() - Constructor for class `com.realvnc.mirrorlink.EventConfiguration`

extensionEnabled(VncServer, VncExtension, boolean) - Method in interface `com.realvnc.vncserver.android.VncExtensionListener`

Handle an incoming extension enable or disable message.

extensionMessageReceived(VncServer, VncExtension, byte[], int, int) - Method in interface `com.realvnc.vncserver.android.VncExtensionListener`

Handle an incoming extension message.

F

FEATURE_CLIPBOARD - Static variable in class `com.realvnc.vncserver.android.VncServer`

Whether or not support should be enabled for transferring clipboard information between the viewer and server.

FEATURE_COMPARE_FB - Static variable in class `com.realvnc.vncserver.android.VncServer`

Enable or disable framebuffer comparison.

FEATURE_DEVICE_LOCK_DISABLED - Static variable in class `com.realvnc.mirrorlink.VNCDeviceStatus`

Device lock is or should be disabled.

FEATURE_DEVICE_LOCK_ENABLED - Static variable in class `com.realvnc.mirrorlink.VNCDeviceStatus`

Device lock is or should be enabled.

FEATURE_DEVICE_LOCK_IGNORED - Static variable in class `com.realvnc.mirrorlink.VNCDeviceStatus`

This message does not specify the status of the device lock.

FEATURE_DEVICE_LOCK_MASK - Static variable in class `com.realvnc.mirrorlink.VNCDeviceStatus`

Used to mask out the device lock status for testing.

FEATURE_DRIVER_DISTRACTION_AVOIDANCE_DISABLED - Static variable in class `com.realvnc.mirrorlink.VNCDeviceStatus`

Driver distraction avoidance is or should be disabled.

FEATURE_DRIVER_DISTRACTION_AVOIDANCE_ENABLED - Static variable in class `com.realvnc.mirrorlink.VNCDeviceStatus`

Driver distraction avoidance is or should be enabled.

FEATURE_DRIVER_DISTRACTION_AVOIDANCE_IGNORED - Static variable in class `com.realvnc.mirrorlink.VNCDeviceStatus`

This message does not specify the status of driver distraction avoidance.

FEATURE_DRIVER_DISTRACTION_AVOIDANCE_MASK - Static variable in class `com.realvnc.mirrorlink.VNCDeviceStatus`

Used to mask out the driver distraction avoidance status for testing.

FEATURE_FRAMEBUFFER_ROTATION_0_DEGREES - Static variable in class `com.realvnc.mirrorlink.VNCDeviceStatus`

The server framebuffer is not or should not be rotated.

FEATURE_FRAMEBUFFER_ROTATION_180_DEGREES - Static variable in class `com.realvnc.mirrorlink.VNCDeviceStatus`

The server framebuffer is or should be rotated 180 degrees.

FEATURE_FRAMEBUFFER_ROTATION_270_DEGREES - Static variable in class `com.realvnc.mirrorlink.VNCDeviceStatus`

The server framebuffer is or should be rotated 270 degrees clockwise (or, equivalently, 90 degrees counter-clockwise).

FEATURE_FRAMEBUFFER_ROTATION_90_DEGREES - Static variable in class `com.realvnc.mirrorlink.VNCDeviceStatus`

The server framebuffer is or should be rotated 90 degrees clockwise.

FEATURE_FRAMEBUFFER_ROTATION_IGNORED - Static variable in class `com.realvnc.mirrorlink.VNCDeviceStatus`

This message does not specify the framebuffer rotation.

FEATURE_FRAMEBUFFER_ROTATION_MASK - Static variable in class `com.realvnc.mirrorlink.VNCDeviceStatus`

Used to mask out the framebuffer rotation for testing.

FEATURE_KEY_LOCK_DISABLED - Static variable in class `com.realvnc.mirrorlink.VNCDeviceStatus`

Key lock is or should be disabled.

FEATURE_KEY_LOCK_ENABLED - Static variable in class `com.realvnc.mirrorlink.VNCDeviceStatus`

Key lock is or should be enabled.

FEATURE_KEY_LOCK_IGNORED - Static variable in class `com.realvnc.mirrorlink.VNCDeviceStatus`

This message does not specify the status of the key lock.

FEATURE_KEY_LOCK_MASK - Static variable in class `com.realvnc.mirrorlink.VNCDeviceStatus`

Used to mask out the key lock status for testing.

FEATURE_MICROPHONE_INPUT_DISABLED - Static variable in class `com.realvnc.mirrorlink.VNCDeviceStatus`

Microphone input is or should be disabled.

FEATURE_MICROPHONE_INPUT_ENABLED - Static variable in class `com.realvnc.mirrorlink.VNCDeviceStatus`

Microphone input is or should be enabled.

FEATURE_MICROPHONE_INPUT_IGNORED - Static variable in class `com.realvnc.mirrorlink.VNCDeviceStatus`

This message does not specify the status of microphone input.

FEATURE_MICROPHONE_INPUT_MASK - Static variable in class `com.realvnc.mirrorlink.VNCDeviceStatus`

Used to mask out the microphone input status for testing.

FEATURE_MIRRORLINK_FORBID_PORTRAIT_ORIENTATION - Static variable in class `com.realvnc.vncserver.android.VncServer`

Whether or not to block frame buffer updates for MirrorLink connection when display orientation is portrait.

FEATURE_NIGHT_MODE_DISABLED - Static variable in class `com.realvnc.mirrorlink.VNCDeviceStatus`

Night mode is or should be disabled.

FEATURE_NIGHT_MODE_ENABLED - Static variable in class `com.realvnc.mirrorlink.VNCDeviceStatus`

Night mode is or should be enabled.

FEATURE_NIGHT_MODE_IGNORED - Static variable in class `com.realvnc.mirrorlink.VNCDeviceStatus`

This message does not specify the status of night mode.

FEATURE_NIGHT_MODE_MASK - Static variable in class `com.realvnc.mirrorlink.VNCDeviceStatus`

Used to mask out the night mode status for testing.

FEATURE_ORIENTATION_IGNORED - Static variable in class `com.realvnc.mirrorlink.VNCDeviceStatus`

This message does not specify the display orientation.

FEATURE_ORIENTATION_LANDSCAPE - Static variable in class `com.realvnc.mirrorlink.VNCDeviceStatus`

The display orientation is or should be landscape.

FEATURE_ORIENTATION_MASK - Static variable in class `com.realvnc.mirrorlink.VNCDeviceStatus`

Used to mask out the display orientation for testing.

FEATURE_ORIENTATION_PORTRAIT - Static variable in class `com.realvnc.mirrorlink.VNCDeviceStatus`

The display orientation is or should be portrait.

FEATURE_RFB4 - Static variable in class `com.realvnc.vncserver.android.VncServer`

Enables or disables RFB4 connections.

FEATURE_SCREENSAVER_DISABLED - Static variable in class `com.realvnc.mirrorlink.VNCDeviceStatus`

Screensaver is or should be disabled.

FEATURE_SCREENSAVER_ENABLED - Static variable in class `com.realvnc.mirrorlink.VNCDeviceStatus`

Screensaver is or should be enabled.

FEATURE_SCREENSAVER_IGNORED - Static variable in class `com.realvnc.mirrorlink.VNCDeviceStatus`

This message does not specify the status of the screensaver.

FEATURE_SCREENSAVER_MASK - Static variable in class `com.realvnc.mirrorlink.VNCDeviceStatus`

Used to mask out the screensaver status for testing.

FEATURE_SEND_CLIPBOARD_ON_CONNECTION - Static variable in class `com.realvnc.vncserver.android.VncServer`

If the clipboard feature is enabled, then defines whether or not the server clipboard contents should be sent to the viewer when a connection is first established.

FEATURE_START_IN_LANDSCAPE - Static variable in class `com.realvnc.vncserver.android.VncServer`

Requests that the initial server configuration is reported as landscape.

FEATURE_VIEW_ONLY - Static variable in class `com.realvnc.vncserver.android.VncServer`

On Windows mobile systems, tells the server to release all key/pointer presses and ignore further input.

FEATURE_VOICE_INPUT_DISABLED - Static variable in class `com.realvnc.mirrorlink.VNCDeviceStatus`

Voice input is or should be disabled.

FEATURE_VOICE_INPUT_ENABLED - Static variable in class `com.realvnc.mirrorlink.VNCDeviceStatus`

Voice input is or should be enabled.

FEATURE_VOICE_INPUT_IGNORED - Static variable in class `com.realvnc.mirrorlink.VNCDeviceStatus`

This message does not specify the status of voice input.

FEATURE_VOICE_INPUT_MASK - Static variable in class `com.realvnc.mirrorlink.VNCDeviceStatus`

Used to mask out the voice input status for testing.

FEATURE_VOICE_INPUT_REROUTING_DISABLED - Static variable in class `com.realvnc.mirrorlink.VNCDeviceStatus`

Deprecated.

Use `VNCDeviceStatus.FEATURE_MICROPHONE_INPUT_DISABLED` instead.

FEATURE_VOICE_INPUT_REROUTING_ENABLED - Static variable in class `com.realvnc.mirrorlink.VNCDeviceStatus`

Deprecated.

Use `VNCDeviceStatus.FEATURE_MICROPHONE_INPUT_ENABLED` instead.

FEATURE_VOICE_INPUT_REROUTING_IGNORED - Static variable in class `com.realvnc.mirrorlink.VNCDeviceStatus`

Deprecated.

Use `VNCDeviceStatus.FEATURE_MICROPHONE_INPUT_IGNORED` instead.

FEATURE_VOICE_INPUT_REROUTING_MASK - Static variable in class `com.realvnc.mirrorlink.VNCDeviceStatus`

Deprecated.

Use `VNCDeviceStatus.FEATURE_MICROPHONE_INPUT_MASK` instead.

FEATURE_Wince_SET_DISPLAY_POLL_FREQUENCY - Static variable in class `com.realvnc.vncserver.android.VncServer`

On Windows mobile systems, sets a refresh rate for the server screen polling.

fields - Variable in class `com.realvnc.vncserver.core.VncCommandStringBase`

Table for associating command string field names with their values.

FLAG_CLIENT_REPEAT - Static variable in interface `com.realvnc.vncserver.android.VncMirrorLinkKeyEventListener`

Bitwise flag used in `VncMirrorLinkKeyEventListener.mlKeyEventReceived(int, int)` to indicate that the key event is an automatic key repeat which has been generated at the client side.

FLAG_KEY_DOWN - Static variable in interface `com.realvnc.vncserver.android.VncMirrorLinkKeyEventListener`

Bitwise flag used in `VncMirrorLinkKeyEventListener.mlKeyEventReceived(int, int)` to indicate that the key event is a key down event.

FLAG_SERVER_REPEAT - Static variable in interface `com.realvnc.vncserver.android.VncMirrorLinkKeyEventListener`

Bitwise flag used in `VncMirrorLinkKeyEventListener.mlKeyEventReceived(int, int)` to indicate that the key event is an automatic key repeat which has been generated at the server side.

forceScreenGrab() - Method in interface `com.realvnc.vncserver.android.VncH264Encoder.ScreenGrabHelper`

Triggers an extra screen grab, which will be rendered to the input Surface.

FRAMEBUFFER_CONFIGURATION_DOWNSCALING - Static variable in class `com.realvnc.mirrorlink.DisplayConfiguration`

The server is capable of downscaling its framebuffer if its own dimensions are greater than those of the client display, which are specified in the `VNCClientDisplayConfiguration` structure.

FRAMEBUFFER_CONFIGURATION_REPLACE_EMPTY_UPDATES - Static variable in class `com.realvnc.mirrorlink.DisplayConfiguration`

The server is capable of suppressing empty update rectangles, so that the viewer application does not have to deal with them.

FRAMEBUFFER_CONFIGURATION_SERVERSIDE_ORIENTATION_SWITCH - Static variable in class `com.realvnc.mirrorlink.DisplayConfiguration`

The server is capable of switching orientations, as instructed by an appropriate `DeviceStatusRequest` message.

FRAMEBUFFER_CONFIGURATION_SERVERSIDE_ROTATION - Static variable in class `com.realvnc.mirrorlink.DisplayConfiguration`

The server is capable of rotating its framebuffer, as instructed by an appropriate `DeviceStatusRequest` message.

FRAMEBUFFER_CONFIGURATION_SUPPORTS_FRAMEBUFFER_ALTERNATIVE_TEXT - Static variable in class `com.realvnc.mirrorlink.DisplayConfiguration`

The server supports `FramebufferAlternativeText` messages.

FRAMEBUFFER_CONFIGURATION_UPSCALING - Static variable in class `com.realvnc.mirrorlink.DisplayConfiguration`

The server is capable of upscaling its framebuffer if its own dimensions are less than those of the client display, which are specified in the `VNCClientDisplayConfiguration` structure.

freeze(boolean) - Method in class `com.realvnc.vncserver.android.VncServer`

Freeze or thaw the server.

G

generateKey(int) - Method in class `com.realvnc.vncserver.android.VncServer`

Instruct the server to start generation of a new RSA encryption key.

get(String, String) - Method in class `com.realvnc.util.JniFile`

getActivity() - Method in interface `com.realvnc.vncserver.android.VncContextInformationManager.CapturedContextInformation`

Retrieve the component name of the activity responsible for the area of the screen.

getApplicationCategory() - Method in class `com.realvnc.mirrorlink.VNCContextInformation`

The category and sub-category into which the application falls.

getApplicationCategoryTrustLevel() - Method in class `com.realvnc.mirrorlink.VNCContextInformation`

The server's level of trust that the information in the `applicationCategory` field is correct.

getApplicationUniqueId() - Method in class `com.realvnc.mirrorlink.VNCAudioBlockingNotification`

Retrieves the unique ID of the blocked application.

getApplicationUniqueId() - Method in class `com.realvnc.mirrorlink.VNCContextInformation`

Return the application unique ID.

getApplicationUniqueId() - Method in class `com.realvnc.mirrorlink.VNCFramebufferBlockingNotification`

Retrieves the unique ID of a blocked application.

getAuthentication() - Method in class `com.realvnc.vncserver.android.VncServer`

Get the currently selected authentication type.

getBase64Value(String) - Method in class `com.realvnc.vncserver.core.VncCommandStringBase`

Extract the named field from the command string (which is assumed to be base 64 encoded), and decode it.

getBearerInfo() - Method in class `com.realvnc.vncserver.android.VncServer`

Return an enumeration of objects implementing the `VncBearerInfo` interface.

getBigEndianFlag() - Method in class `com.realvnc.vncserver.core.VncPixelFormat`

Returns the big endian flag for this pixel format.

getBitsPerPixel() - Method in class `com.realvnc.vncserver.core.VncPixelFormat`

Returns the bits per pixel for this pixel format.

getBlueMax() - Method in class `com.realvnc.vncserver.core.VncPixelFormat`

Returns the maximum value of the blue component for this pixel format.

getBlueShift() - Method in class `com.realvnc.vncserver.core.VncPixelFormat`

Returns the blue shift value for this pixel format.

getBluetoothAdapterAddress() - Method in class `com.realvnc.vncserver.android.VncServer`

Gets the hardware address of the local Bluetooth adapter.

getBoolean(String) - Method in class `com.realvnc.vncserver.core.VncCommandStringBase`

Extract the named field from the command string and convert to a boolean.

getClientDisplayHeightMillimeters() - Method in class `com.realvnc.mirrorlink.VNCClientDisplayConfiguration`

Retrieves the display height size.

getClientDisplayHeightPixels() - Method in class `com.realvnc.mirrorlink.VNCClientDisplayConfiguration`

Retrieves the display pixel height.

getClientDisplayWidthMillimeters() - Method in class `com.realvnc.mirrorlink.VNCClientDisplayConfiguration`

Retrieves the display width size.

getClientDisplayWidthPixels() - Method in class `com.realvnc.mirrorlink.VNCClientDisplayConfiguration`

Retrieves the display pixel width.

getClientDistanceFromUserMillimeters() - Method in class `com.realvnc.mirrorlink.VNCClientDisplayConfiguration`

Retrieves the expected distance from user.

getClientMajorVersion() - Method in class `com.realvnc.mirrorlink.VNCClientDisplayConfiguration`

Retrieves the major client version.

getClientMinorVersion() - Method in class `com.realvnc.mirrorlink.VNCClientDisplayConfiguration`

Retrieves the minor client version.

getContentCategory() - Method in class `com.realvnc.mirrorlink.VNCContextInformation`

The category of the content that the application is presenting.

getContentCategoryTrustLevel() - Method in class `com.realvnc.mirrorlink.VNCContextInformation`

The server's level of trust that the information in the `contentCategory` field is correct.

getContentRulesFollowed() - Method in class `com.realvnc.mirrorlink.VNCContextInformation`

Deprecated.

Since MirrorLink 1.3. Must be ignored by MirrorLink 1.3 clients.

getContextInformationManager() - Method in class `com.realvnc.vncserver.android.VncServer`

Retrieves the context information manager instance for this server object.

getCurrentPipelineLag() - Method in class `com.realvnc.vncserver.android.VncH264Encoder`

Returns the number of stages in the encoding pipeline.

getDepth() - Method in class `com.realvnc.vncserver.core.VncPixelFormat`

Returns the depth for this pixel format.

getDescription() - Method in interface `com.realvnc.vncserver.core.VncBearerInfo`

Returns a description of this bearer and the transports that it supports.

getDeviceKeySupport() - Method in class `com.realvnc.mirrorlink.VNCServerEventConfiguration`

Retrieves the device key support as a bitmask.

getDisplayInformationManager() - Method in class `com.realvnc.vncserver.android.VncServer`
Retrieves the display information manager instance for this server object.

getEncryption() - Method in class `com.realvnc.vncserver.android.VncServer`
Get the currently selected encryption type.

getFeatures() - Method in class `com.realvnc.mirrorlink.VNCDeviceStatus`
Retrieves a bitmask of device status features.

getFlags() - Method in interface `com.realvnc.vncserver.android.VncContextInformationManager.CapturedContextInformation`
Retrieve a combination of flags describing this context information.

getFramebufferConfiguration() - Method in class `com.realvnc.mirrorlink.VNCClientDisplayConfiguration`
Retrieves the framebuffer configuration.

getFramebufferConfiguration() - Method in class `com.realvnc.mirrorlink.VNCServerDisplayConfiguration`
Retrieves a bitmask of the framebuffer configuration.

getFullName() - Method in interface `com.realvnc.vncserver.core.VncBearerInfo`
Returns a longer human readable name for this bearer.

getGreenMax() - Method in class `com.realvnc.vncserver.core.VncPixelFormat`
Returns the maximum value of the green component for this pixel format.

getGreenShift() - Method in class `com.realvnc.vncserver.core.VncPixelFormat`
Returns the green shift value for this pixel format.

getHeight() - Method in class `com.realvnc.vncserver.android.VncSizeInt`
Returns the height contained in this object.

getInfo() - Method in interface `com.realvnc.vncserver.core.VncBearer`
Returns an object containing descriptive information about the bearer.

getInputStream() - Method in interface `com.realvnc.vncserver.core.VncConnection`
Once a connection has been established returns an `InputStream` which can be used to read data over the bearer.

getInt(String) - Method in class `com.realvnc.vncserver.core.VncCommandStringBase`
Extract the named field from the command string and convert to an integer.

getKeyboardCountry() - Method in class `com.realvnc.mirrorlink.VNCServerEventConfiguration`
Retrieves the keyboard layout country code.

getKeyboardLanguage() - Method in class `com.realvnc.mirrorlink.VNCServerEventConfiguration`
Retrieves the keyboard layout language code.

getKeyInjectionSupport() - Method in interface `com.realvnc.vncserver.android.VncRemoteControlInfo`
Returns whether this form of remote control supports the injection of key events.

getKnobKeySupport() - Method in class `com.realvnc.mirrorlink.VNCServerEventConfiguration`
Retrieves the knob key support as a bitmask.

getLocalAddress() - Method in interface `com.realvnc.vncserver.core.VncConnection`
Return the local address associated with this connection.

getMessage() - Method in exception `com.realvnc.vncserver.core.VncException`

getMiscKeySupport() - Method in class `com.realvnc.mirrorlink.VNCServerEventConfiguration`
Retrieves the miscellaneous key support as a bitmask.

getMotionInjectionSupport() - Method in interface `com.realvnc.vncserver.android.VncRemoteControlInfo`
Returns whether this form of remote control supports the injection of motion events.

getMultimediaKeySupport() - Method in class `com.realvnc.mirrorlink.VNCServerEventConfiguration`
Retrieves the multimedia key support as a bitmask.

getName() - Method in interface `com.realvnc.vncserver.core.VncBearerInfo`
Returns the short name for this bearer.

getNearestSupportedResolution(int, int, int, int) - Method in class `com.realvnc.vncserver.android.VncH264Encoder`
Returns the nearest supported resolution to the one specified, or null if no such resolution exists.

getNumFunctionKeysSupported() - Method in class `com.realvnc.mirrorlink.VNCServerEventConfiguration`
Retrieves the number of function keys supported.

getOrientationManager() - Method in class `com.realvnc.vncserver.android.VncServer`
Retrieves the orientation manager instance for this server object.

getOutputStream() - Method in interface `com.realvnc.vncserver.core.VncConnection`
Once a connection has been established returns an `OutputStream` which can be used to write data over the bearer.

getPixelFormatSupport() - Method in class `com.realvnc.mirrorlink.VNCClientDisplayConfiguration`
Returns the pixel formats supported by the client for use with the Transform encoding.

getPixelFormatSupport() - Method in class `com.realvnc.mirrorlink.VNCServerDisplayConfiguration`
Retrieves a bitmask of the supported pixel formats.

getPointerSupport() - Method in class `com.realvnc.mirrorlink.VNCServerEventConfiguration`
Retrieves the pointer / touchscreen support as a bitmask.

getReason() - Method in class `com.realvnc.mirrorlink.VNCAudioBlockingNotification`
Retrieves the reason for blocking the application.

getReason() - Method in class `com.realvnc.mirrorlink.VNCFramebufferBlockingNotification`
Retrieves the reason for application blocking.

getRect() - Method in interface `com.realvnc.vncserver.android.VncContextInformationManager.CapturedContextInformation`
Retrieves the area of the screen covered by this context information.

getRedMax() - Method in class `com.realvnc.vncserver.core.VncPixelFormat`
Returns the maximum value of the red component for this pixel format.

getRedShift() - Method in class `com.realvnc.vncserver.core.VncPixelFormat`
Returns the red shift value for this pixel format.

getRelativePixelHeight() - Method in class `com.realvnc.mirrorlink.VNCServerDisplayConfiguration`
Retrieves the relative pixel height.

getRelativePixelWidth() - Method in class `com.realvnc.mirrorlink.VNCServerDisplayConfiguration`
Retrieves the relative pixel width.

getRemoteAddress() - Method in interface `com.realvnc.vncserver.core.VncConnection`
Return the remote address associated with this connection.

getRemoteControlInfo() - Method in class `com.realvnc.vncserver.android.VncServer`
Gets details of the device's remote control support.

getResizeFactors() - Method in class `com.realvnc.mirrorlink.VNCClientDisplayConfiguration`
Returns the resize factors supported by the client for use with the Transform encoding.

getSerialNumber() - Method in exception `com.realvnc.vncserver.core.VncLicenseNotValidException`
Retrieves the serial number of the license which was invalid.

getServerMajorVersion() - Method in class `com.realvnc.mirrorlink.VNCServerDisplayConfiguration`
Retrieves the major server version.

getServerMinorVersion() - Method in class `com.realvnc.mirrorlink.VNCServerDisplayConfiguration`
Retrieves the minor server version.

getServerSignature() - Method in class `com.realvnc.vncserver.android.VncServer`
Returns the signature for the server's RSA key.

getState() - Method in class `com.realvnc.vncserver.android.VncServer`
Return the current state of the VNC server.

getString(String) - Method in class `com.realvnc.vncserver.core.VncCommandStringBase`
Extract the named field from the command string and return as a string.

getTrueColorFlag() - Method in class `com.realvnc.vncserver.core.VncPixelFormat`
Returns the true color flag for this pixel format.

getUiCountry() - Method in class `com.realvnc.mirrorlink.VNCServerEventConfiguration`
Retrieves the user interface country code.

getUiLanguage() - Method in class `com.realvnc.mirrorlink.VNCServerEventConfiguration`
Retrieves the user interface language code.

getUniqueId() - Method in class `com.realvnc.mirrorlink.VNCFramebufferBlockingNotification`
Retrieves the unique ID of the notification.

getUserData() - Method in class `com.realvnc.vncserver.android.VncServer`
Retrieves the user defined data previously associated with this server.

getVersionString() - Method in class `com.realvnc.vncserver.android.VncServer`
Get the VNC server version string in form "major.minor.patch.revision number"

getVersionString() - Method in interface `com.realvnc.vncserver.core.VncBearerInfo`
Returns the version string for this bearer.

getWidth() - Method in class `com.realvnc.vncserver.android.VncSizeInt`
Returns the width contained in this object.

giveBuffer(ByteBuffer) - Method in interface `com.realvnc.vncserver.android.VncH264Encoder.BufferOwner`
Transfers ownership of the provided buffer.

H

H264_LEVEL_3_1 - Static variable in class `com.realvnc.vncserver.android.VncH264Encoder`

H.264 Level 3.1

H264_LEVEL_4_1 - Static variable in class `com.realvnc.vncserver.android.VncH264Encoder`

H.264 Level 4.1

H264_PROFILE_BASELINE - Static variable in class `com.realvnc.vncserver.android.VncH264Encoder`
H.264 Baseline Profile

hashCode() - Method in class `com.realvnc.vncserver.android.VncSizeInt`

hashCode() - Method in class `com.realvnc.vncserver.core.VncPixelFormat`
Calculates a hash code for the object.

I

IniFile - Class in `com.realvnc.util`

IniFile() - Constructor for class `com.realvnc.util.IniFile`

IniFile.BadFormatException - Exception in `com.realvnc.util`

IniFile.BadFormatException() - Constructor for exception `com.realvnc.util.IniFile.BadFormatException`

init(String) - Static method in class `com.realvnc.util.VncLog`

init(String, boolean) - Static method in class `com.realvnc.util.VncLog`

injectKeyEvent(KeyEvent) - Method in class `com.realvnc.vncserver.android.VncServer`
Inject a keyboard event into the system.

isConnectionTearingDown() - Method in class `com.realvnc.vncserver.android.VncServer`
Return true if the tearing down is in progress after either the existing connection is stopped to be processed or the VNC server instance stops listening for new connection.

isEventMappingSupported() - Method in class `com.realvnc.mirrorlink.VNCServerEventConfiguration`
Check whether event mapping is supported.

isITUKeySupported() - Method in class `com.realvnc.mirrorlink.VNCServerEventConfiguration`
Check whether ITU keypad events are supported.

isKeyEventListingSupported() - Method in class `com.realvnc.mirrorlink.VNCServerEventConfiguration`
Check whether the key event listing is supported.

isVirtualKeyboardTriggerSupported() - Method in class `com.realvnc.mirrorlink.VNCServerEventConfiguration`
Check whether the virtual keyboard trigger is supported.

K

keyboardCountry - Variable in class `com.realvnc.mirrorlink.VNCServerEventConfiguration`
Contains the keyboard layout country code.

keyboardLanguage - Variable in class `com.realvnc.mirrorlink.VNCServerEventConfiguration`
Contains the keyboard layout language code.

keygenCb(VncServer, byte[]) - Method in interface `com.realvnc.vncserver.android.VncServerListener`
Called when RSA key pair generation has completed.

KNOB_KEY_SUPPORT_PULL_Z_0 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

KNOB_KEY_SUPPORT_PULL_Z_1 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

KNOB_KEY_SUPPORT_PULL_Z_2 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

KNOB_KEY_SUPPORT_PULL_Z_3 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

KNOB_KEY_SUPPORT_PUSH_Z_0 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

KNOB_KEY_SUPPORT_PUSH_Z_1 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

KNOB_KEY_SUPPORT_PUSH_Z_2 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

KNOB_KEY_SUPPORT_PUSH_Z_3 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

KNOB_KEY_SUPPORT_ROTATE_X_0 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

KNOB_KEY_SUPPORT_ROTATE_X_1 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

KNOB_KEY_SUPPORT_ROTATE_X_2 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

KNOB_KEY_SUPPORT_ROTATE_X_3 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

KNOB_KEY_SUPPORT_ROTATE_Y_0 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

KNOB_KEY_SUPPORT_ROTATE_Y_1 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

KNOB_KEY_SUPPORT_ROTATE_Y_2 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

KNOB_KEY_SUPPORT_ROTATE_Y_3 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

KNOB_KEY_SUPPORT_ROTATE_Z_0 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

KNOB_KEY_SUPPORT_ROTATE_Z_1 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

KNOB_KEY_SUPPORT_ROTATE_Z_2 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

KNOB_KEY_SUPPORT_ROTATE_Z_3 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

KNOB_KEY_SUPPORT_SHIFT_X_0 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

KNOB_KEY_SUPPORT_SHIFT_X_1 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

KNOB_KEY_SUPPORT_SHIFT_X_2 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

KNOB_KEY_SUPPORT_SHIFT_X_3 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

KNOB_KEY_SUPPORT_SHIFT_XY_0 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

KNOB_KEY_SUPPORT_SHIFT_XY_1 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

KNOB_KEY_SUPPORT_SHIFT_XY_2 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

KNOB_KEY_SUPPORT_SHIFT_XY_3 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

KNOB_KEY_SUPPORT_SHIFT_Y_0 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

KNOB_KEY_SUPPORT_SHIFT_Y_1 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

KNOB_KEY_SUPPORT_SHIFT_Y_2 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

KNOB_KEY_SUPPORT_SHIFT_Y_3 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

knobKeySupport - Variable in class `com.realvnc.mirrorlink.VNCServerEventConfiguration`

Contains a bitmask of knob key support.

L

listen(int) - Method in class `com.realvnc.vncserver.android.VncServer`

Starts this VNC server instance listening for an incoming connection using the VNC TCP/IP inbound pluggable bearer.

listeningCb(VncServer, String) - Method in interface `com.realvnc.vncserver.android.VncServerListener`

Callback to indicate that the VNC server is listening for an incoming connection.

localFeatureCheck(int[]) - Method in class `com.realvnc.vncserver.android.VncServer`

Performs a local feature check.

localFeatureCheck(int[]) - Method in interface `com.realvnc.vncserver.core.VncBearerCallbacks`

Requests that the SDK performs a feature check on the local licenses.

lockOrientation(int) - Method in class `com.realvnc.vncserver.android.VncOrientationManager`

Deprecated.

Use `VncOrientationManager.lockOrientationEx(int)` instead

lockOrientationEx(int) - Method in class `com.realvnc.vncserver.android.VncOrientationManager`

Requests that the screen orientation is locked to the requested orientation.

login(String, String) - Method in class `com.realvnc.vncserver.android.VncServer`

Provide user name and/or password to viewer during reverse authentication.

loginCb(VncServer, boolean, boolean) - Method in interface `com.realvnc.vncserver.android.VncServerListener`

Login callback issued during reverse authentication.

M

MirrorLinkCallbackHandler - Class in `com.realvnc.vncserver.android`

Base class for callbacks received for a MirrorLink server.

MirrorLinkCallbackHandler() - Constructor for class `com.realvnc.vncserver.android.MirrorLinkCallbackHandler`

MISC_KEY_SUPPORT_EVENT_MAPPING - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

Supports the Event Mapping feature.

MISC_KEY_SUPPORT_FUNCTION_KEY_0 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

Supports Function Key 0.

MISC_KEY_SUPPORT_FUNCTION_KEY_1 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

Supports Function Key 1.

MISC_KEY_SUPPORT_FUNCTION_KEY_2 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

Supports Function Key 2.

MISC_KEY_SUPPORT_FUNCTION_KEY_3 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

Supports Function Key 3.

MISC_KEY_SUPPORT_FUNCTION_KEY_4 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

Supports Function Key 4.

MISC_KEY_SUPPORT_FUNCTION_KEY_5 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

Supports Function Key 5.

MISC_KEY_SUPPORT_FUNCTION_KEY_6 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

Supports Function Key 6.

MISC_KEY_SUPPORT_FUNCTION_KEY_7 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

Supports Function Key 7.

MISC_KEY_SUPPORT_FUNCTION_KEY_MASK - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

MISC_KEY_SUPPORT_FUNCTION_KEY_SHIFT - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

MISC_KEY_SUPPORT_ITU - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

Supports ITU keypad events.

MISC_KEY_SUPPORT_KEY_EVENT_LISTING - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

Supports the Key Event Listing feature.

MISC_KEY_SUPPORT_KEY_MAPPING_MASK - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

MISC_KEY_SUPPORT_KEY_MAPPING_SHIFT - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

MISC_KEY_SUPPORT_VIRTUAL_KEYBOARD_TRIGGER - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

Supports the Virtual Keyboard Trigger feature.

miscKeySupport - Variable in class `com.realvnc.mirrorlink.VNCServerEventConfiguration`

Contains a bitmask of miscellaneous support.

mlAudioBlockingNotificationReceivedCb(VNCAudioBlockingNotification) - Method in interface `com.realvnc.vncserver.android.VncServerMirrorLinkListener`

An Audio Blocking Notification message has been received from the MirrorLink viewer.

mlClientDisplayConfigurationReceivedCb(VNCClientDisplayConfiguration, VNCServerEventConfiguration) - Method in interface `com.realvnc.vncserver.android.VncServerMirrorLinkListener`

We have received a ClientDisplayConfiguration message from the MirrorLink viewer.

mlClientEventConfigurationReceivedCb(VNCClientEventConfiguration, Map<Integer, Integer>) - Method in interface `com.realvnc.vncserver.android.VncServerMirrorLinkListener`

The final step of the MirrorLink handshake has occurred - we have received a client event configuration message from the viewer.

mlConnectionReceivedCb(VNCServerDisplayConfiguration) - Method in interface `com.realvnc.vncserver.android.VncServerMirrorLinkListener`

A MirrorLink connection has been established to the server.

mlDeviceStatusSendNeededCb(VNCDeviceStatus, VNCDeviceStatus) - Method in interface `com.realvnc.vncserver.android.VncServerMirrorLinkListener`

There is a need to send a DeviceStatus message to the MirrorLink client.

mlEventMappingRequestReceivedCb(int, int) - Method in interface `com.realvnc.vncserver.android.VncServerMirrorLinkListener`

An event mapping **change** request message has been received from the MirrorLink viewer.

mlFrameBufferBlockingNotificationHandled() - Method in class `com.realvnc.vncserver.android.VncServer`

Deprecated.

Use `VncServer.mlFrameBufferBlockingNotificationHandled(VNCFramebufferBlockingNotification)` instead.

mlFrameBufferBlockingNotificationHandled(VNCFramebufferBlockingNotification) - Method in class `com.realvnc.vncserver.android.VncServer`

Indicates that processing of a framebuffer blocking notification message has completed.

mlFramebufferBlockingNotificationReceivedCb(Rect, VNCFramebufferBlockingNotification) - Method in interface `com.realvnc.vncserver.android.VncServerMirrorLinkListener`

A Framebuffer Blocking Notification message has been received from the MirrorLink viewer.

mlFramebufferUnblockedCb() - Method in class `com.realvnc.vncserver.android.MirrorLinkCallbackHandler`

The framebuffer is no longer blocked by the client.

mlGetEventMapping() - Method in class `com.realvnc.vncserver.android.VncServer`

Returns the event mapping presently understood by the server.

mlKeyEventReceived(int, int) - Method in interface `com.realvnc.vncserver.android.VncMirrorLinkKeyEventListener`

Callback used to offer a MirrorLink key event to a registered listener.

mlRegisterKeyEventListener(VncMirrorLinkKeyEventListener) - Method in class `com.realvnc.vncserver.android.VncServer`

Registers a MirrorLink key event listener for this server.

mlRequestSendDeviceStatus() - Method in class `com.realvnc.vncserver.android.VncServer`

MirrorLink: Inform the server that a new device status request should be sent.

mlSendDeviceStatus(VNCDeviceStatus) - Method in class `com.realvnc.vncserver.android.VncServer`

MirrorLink: send a 'device status' message to the viewer, in response to a `VncServerMirrorLinkListener.mlDeviceStatusSendNeededCb(VNCDeviceStatus latestRequest, VNCDeviceStatus defaultReply)` callback.

mlSendEventMappingRequestReply(int) - Method in class `com.realvnc.vncserver.android.VncServer`

Replies to an event mapping request received from the MirrorLink viewer.

mlSendServerDisplayConfiguration(VNCServerDisplayConfiguration) - Method in class `com.realvnc.vncserver.android.VncServer`

MirrorLink: Send a Server Device Configuration message.

mlSendServerEventConfiguration(VNCServerEventConfiguration) - Method in class `com.realvnc.vncserver.android.VncServer`

MirrorLink: Send a Server Event Configuration message.

mlSetContextInformation(List<Pair<Rect, VNContextInformation>>) - Method in class `com.realvnc.vncserver.android.VncServer`

Tell the VNC Server SDK context about areas of the screen.

mlSetContextInformationAndInvalidate(List<Pair<Rect, VNContextInformation>>) - Method in class `com.realvnc.vncserver.android.VncServer`

Tell the VNC Server SDK context about areas of the screen.

mlSetEventMapping(Map<Integer, Integer>) - Method in class `com.realvnc.vncserver.android.VncServer`

Sets event mapping entries.

mlTriggerFrameBufferBlockingNotification(VNCFramebufferBlockingNotification) - Method in class `com.realvnc.vncserver.android.VncServer`

Triggers a server side FramebufferBlockingNotification.

mlUnregisterKeyEventListener(VncMirrorLinkKeyEventListener) - Method in class `com.realvnc.vncserver.android.VncServer`

Unregisters a MirrorLink key event listener from this server.

MULTIMEDIA_KEY_SUPPORT_FORWARD - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

MULTIMEDIA_KEY_SUPPORT_MUTE - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

MULTIMEDIA_KEY_SUPPORT_NEXT - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

MULTIMEDIA_KEY_SUPPORT_PAUSE - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

MULTIMEDIA_KEY_SUPPORT_PHOTO - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

MULTIMEDIA_KEY_SUPPORT_PLAY - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

MULTIMEDIA_KEY_SUPPORT_PREVIOUS - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

MULTIMEDIA_KEY_SUPPORT_REWIND - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

MULTIMEDIA_KEY_SUPPORT_STOP - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

MULTIMEDIA_KEY_SUPPORT_UNMUTE - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

multimediaKeySupport - Variable in class `com.realvnc.mirrorlink.VNCServerEventConfiguration`

Contains a bitmask of multimedia key support.

O

onIFrameRequired() - Method in class `com.realvnc.vncserver.android.VncH264Encoder`

Requests that the encoder generate an I-frame with the H.264 parameter sets in the next call to `encodeFrame()`.

ORIENTATION_DISABLE_LOCK - Static variable in class `com.realvnc.vncserver.android.VncOrientationManager`

Constant indicating that the orientation lock should be disabled.

ORIENTATION_LANDSCAPE_LOCK - Static variable in class `com.realvnc.vncserver.android.VncOrientationManager`

Constant indicating that the orientation should be locked in landscape.

ORIENTATION_PORTRAIT_LOCK - Static variable in class `com.realvnc.vncserver.android.VncOrientationManager`

Constant indicating that the orientation should be locked in portrait.

P

PACKAGE_SYSTEM - Static variable in class `com.realvnc.vncserver.android.VncContextInformationManager`

Constant used as the package name for system UI elements which are part of the system and so do not have an associated Android package.

PAL_8 - Static variable in class `com.realvnc.vncserver.core.VncPixelFormat`

A `VncPixelFormat` instance representing a pixel format where each component has an 8-bit representation.

parameterPresent(String) - Method in class `com.realvnc.vncserver.core.VncCommandStringBase`

Returns true if the named parameter was successfully parsed from the command string, otherwise false.

parse(InputStream) - Method in class `com.realvnc.util.JniFile`

parse(String) - Method in class `com.realvnc.vncserver.core.VncCommandStringBase`

Parse the given command string and break it down into separate fields which are then checked added as key/value strings to the main hash table.

PIXELFORMAT_SUPPORT_ANY_16 - Static variable in class `com.realvnc.mirrorlink.DisplayConfiguration`

Supports any other 16-bit true color pixel formats.

PIXELFORMAT_SUPPORT_ANY_24 - Static variable in class `com.realvnc.mirrorlink.DisplayConfiguration`

Supports any other 24-bit true color pixel formats.

PIXELFORMAT_SUPPORT_ANY_32 - Static variable in class `com.realvnc.mirrorlink.DisplayConfiguration`

Supports any other 32-bit true color pixel formats.

PIXELFORMAT_SUPPORT_ARGB888_32 - Static variable in class `com.realvnc.mirrorlink.DisplayConfiguration`

Supports the ARGB32 pixel format.

PIXELFORMAT_SUPPORT_GRAYSCALE_16 - Static variable in class `com.realvnc.mirrorlink.DisplayConfiguration`

Supports 16-bit grayscale.

PIXELFORMAT_SUPPORT_GRAYSCALE_8 - Static variable in class `com.realvnc.mirrorlink.DisplayConfiguration`

Supports 8-bit grayscale.

PIXELFORMAT_SUPPORT_NONE - Static variable in class `com.realvnc.mirrorlink.DisplayConfiguration`

Supports no pixel formats.

PIXELFORMAT_SUPPORT_RGB444_16 - Static variable in class `com.realvnc.mirrorlink.DisplayConfiguration`

Supports the RGB444 pixel formats.

PIXELFORMAT_SUPPORT_RGB555_16 - Static variable in class `com.realvnc.mirrorlink.DisplayConfiguration`

Supports the RGB555 pixel formats.

PIXELFORMAT_SUPPORT_RGB565_16 - Static variable in class `com.realvnc.mirrorlink.DisplayConfiguration`

Supports the RGB565 pixel formats.

PIXELFORMAT_SUPPORT_RGB888_32 - Static variable in class `com.realvnc.mirrorlink.DisplayConfiguration`

Supports the RGB888 pixel format.

PIXELFORMAT_SUPPORT_RGB_343_16 - Static variable in class `com.realvnc.mirrorlink.DisplayConfiguration`

Supports the RGB343 pixel formats.

POINTER_SUPPORT_POINTER_BUTTON_1 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

POINTER_SUPPORT_POINTER_BUTTON_2 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

POINTER_SUPPORT_POINTER_BUTTON_3 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

POINTER_SUPPORT_POINTER_BUTTON_4 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

POINTER_SUPPORT_POINTER_BUTTON_5 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

POINTER_SUPPORT_POINTER_BUTTON_6 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

POINTER_SUPPORT_POINTER_BUTTON_7 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

POINTER_SUPPORT_POINTER_BUTTON_8 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

POINTER_SUPPORT_POINTER_BUTTON_MASK - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

POINTER_SUPPORT_POINTER_EVENTS - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

POINTER_SUPPORT_TOUCH_COUNT_1 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

POINTER_SUPPORT_TOUCH_COUNT_10 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

POINTER_SUPPORT_TOUCH_COUNT_2 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

POINTER_SUPPORT_TOUCH_COUNT_3 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

POINTER_SUPPORT_TOUCH_COUNT_4 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

POINTER_SUPPORT_TOUCH_COUNT_5 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

POINTER_SUPPORT_TOUCH_COUNT_6 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

POINTER_SUPPORT_TOUCH_COUNT_7 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

POINTER_SUPPORT_TOUCH_COUNT_8 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

POINTER_SUPPORT_TOUCH_COUNT_9 - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

POINTER_SUPPORT_TOUCH_COUNT_MASK - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

POINTER_SUPPORT_TOUCH_COUNT_MASK_SHIFT - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

POINTER_SUPPORT_TOUCH_EVENT_PRESSURE_MASK - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

POINTER_SUPPORT_TOUCH_EVENT_PRESSURE_MASK_SHIFT - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

POINTER_SUPPORT_TOUCH_EVENTS - Static variable in class `com.realvnc.mirrorlink.EventConfiguration`

pointerSupport - Variable in class `com.realvnc.mirrorlink.VNCServerEventConfiguration`
Contains a bitmask of pointer support.

provideLicenseFeature(int) - Method in class `com.realvnc.vncserver.android.VncServer`
Marks the server as providing the specified license feature.

putField(String, String) - Method in class `com.realvnc.vncserver.core.VncCommandStringBase`
Add the given key and value to the parameter hashtable.

Q

queryResolutionSupport(int, int, int, int) - Method in class `com.realvnc.vncserver.android.VncH264Encoder`
Check whether the specified resolution and H.264 parameters are supported.

R

RC_CAPTURE_TEMPORARILY_UNAVAILABLE - Static variable in class `com.realvnc.vncserver.android.RemoteControlServiceCodes`
Return code indicating failure due to a transient reason.

RC_DEVICE_ADMIN_NOT_ENABLED - Static variable in class `com.realvnc.vncserver.android.RemoteControlServiceCodes`
Return code indicating failure due to administration not being enabled.

RC_DISCONNECTED - Static variable in class `com.realvnc.vncserver.android.RemoteControlServiceCodes`
Return code indicating failure due to being disconnected from the service.

RC_INCREMENTAL_UPDATES_UNAVAILABLE - Static variable in class `com.realvnc.vncserver.android.RemoteControlServiceCodes`

Return code indicating failure due to incremental updates not being available.

RC_PERMISSION_DENIED - Static variable in class `com.realvnc.vncserver.android.RemoteControlServiceCodes`

Return code indicating failure due to the calling package not being granted permissions.

RC_SERVICE_ILLEGAL_ARGUMENT - Static variable in class `com.realvnc.vncserver.android.RemoteControlServiceCodes`

Return code indicating failure due to the an illegal argument.

RC_SERVICE_ITSELF_LACKING_PERMISSIONS - Static variable in class `com.realvnc.vncserver.android.RemoteControlServiceCodes`

Return code indicating failure due to the application implementing the service not having sufficient permissions.

RC_SERVICE_LACKING_OTHER_OS_FACILITIES - Static variable in class `com.realvnc.vncserver.android.RemoteControlServiceCodes`

Return code indicating failure due to the operating system not providing the required functionality.

RC_SERVICE_UNAVAILABLE - Static variable in class `com.realvnc.vncserver.android.RemoteControlServiceCodes`

Return code indicating failure due to the service not being available.

RC_SUCCESS - Static variable in class `com.realvnc.vncserver.android.RemoteControlServiceCodes`

Return code indicating success.

REASON_APPLICATION_CATEGORY_NOT_ALLOWED - Static variable in class `com.realvnc.mirrorlink.VNCAudioBlockingNotification`

The application's category has been disallowed (for example, by the driver distraction policy).

REASON_APPLICATION_CATEGORY_NOT_ALLOWED - Static variable in class `com.realvnc.mirrorlink.VNCFramebufferBlockingNotification`

The application category has been disallowed (for example, by the driver distraction policy).

REASON_APPLICATION_NOT_TRUSTED - Static variable in class `com.realvnc.mirrorlink.VNCAudioBlockingNotification`

The server's trust in the application category that it reported is not sufficient to satisfy the viewer application.

REASON_APPLICATION_NOT_TRUSTED - Static variable in class `com.realvnc.mirrorlink.VNCFramebufferBlockingNotification`

The server's trust in the application category that it reported is not sufficient to satisfy the viewer application.

REASON_APPLICATION_UNIQUE_ID_NOT_ALLOWED - Static variable in class `com.realvnc.mirrorlink.VNCAudioBlockingNotification`

The server application has been disallowed based on its unique ID.

REASON_APPLICATION_UNIQUE_ID_NOT_ALLOWED - Static variable in class `com.realvnc.mirrorlink.VNCFramebufferBlockingNotification`

The server application has been disallowed based on its unique ID (for example, by the driver distraction policy).

REASON_CONTENT_CATEGORY_NOT_ALLOWED - Static variable in class `com.realvnc.mirrorlink.VNCFramebufferBlockingNotification`

The application's content category has been disallowed (for example, by the driver distraction policy).

REASON_CONTENT_NOT_TRUSTED - Static variable in class `com.realvnc.mirrorlink.VNCFramebufferBlockingNotification`

The server's trust in the content category that it reported is not sufficient to satisfy the viewer application.

REASON_CONTENT_RULES_NOT_FOLLOWED - Static variable in class `com.realvnc.mirrorlink.VNCFramebufferBlockingNotification`

The server application has not followed the content rules that were communicated to the server via UPnP.

REASON_GLOBALLY_MUTED - Static variable in class `com.realvnc.mirrorlink.VNCAudioBlockingNotification`

The user has muted all audio.

REASON_STREAM_MUTED - Static variable in class `com.realvnc.mirrorlink.VNCAudioBlockingNotification`

The user has muted a particular audio stream.

REASON_UI_LAYOUT_NOT_SUPPORTED - Static variable in class `com.realvnc.mirrorlink.VNCFramebufferBlockingNotification`

UI layout not supported.

REASON_UI_NOT_IN_FOCUS - Static variable in class `com.realvnc.mirrorlink.VNCFramebufferBlockingNotification`

The viewer application is not in focus.

REASON_UI_NOT_VISIBLE - Static variable in class `com.realvnc.mirrorlink.VNCFramebufferBlockingNotification`

The UI of the viewer application is not visible to the user.

REASON_UNBLOCK - Static variable in class `com.realvnc.mirrorlink.VNCAudioBlockingNotification`

The application's blocked audio stream should be resumed.

registerExtension(String, VncExtensionListener) - Method in class `com.realvnc.vncserver.android.VncServer`

Registers a new extension with the SDK.

remoteControlAvailableCb(VncServer, int) - Method in interface `com.realvnc.vncserver.android.VncServerListener`

Called in response to the application calling `checkRemoteControlAvailable()`.

RemoteControlServiceCodes - Class in `com.realvnc.vncserver.android`

Return or error codes that may be reported by Remote Control Service implementations provided by RealVNC for Android platforms.

RemoteControlServiceCodes() - Constructor for class `com.realvnc.vncserver.android.RemoteControlServiceCodes`

remoteFeatureCheckFailed(VncServer, int) - Method in interface `com.realvnc.vncserver.android.VncRemoteFeatureCheckListener`

Called when the VNC viewer has failed to pass a remote feature check.

remoteFeatureCheckSucceeded(VncServer, int, int) - Method in interface `com.realvnc.vncserver.android.VncRemoteFeatureCheckListener`

Called when the VNC viewer has successfully passed a remote feature check.

remoteKeyCb(VncServer, byte[], byte[]) - Method in interface `com.realvnc.vncserver.android.VncServerListener`

Remote Key notification callback.

removeAccessibilityServiceProvider(VncContextInformationManager.AccessibilityServiceProvider) - Method in class `com.realvnc.vncserver.android.VncContextInformationManager`

Removes a previously added accessibility service provider.

removeListener(VncContextInformationManager.Listener) - Method in class `com.realvnc.vncserver.android.VncContextInformationManager`

Removes a previously added listener, preventing it from receiving notification of changes to context information.

removeListener(VncDisplayInformationManager.Listener) - Method in class `com.realvnc.vncserver.android.VncDisplayInformationManager`

Removes a previously added listener, preventing it from receiving notification of changes to display information.

reset() - Method in class `com.realvnc.vncserver.android.VncServer`

Reset the server core back to a disconnected state.

reset(boolean) - Method in class `com.realvnc.vncserver.android.VncServer`

Reset the server core back to a disconnected state.

RESIZE_FACTOR_1_1 - Static variable in class `com.realvnc.mirrorlink.DisplayConfiguration`

Resize factor of 1/1

RESIZE_FACTOR_1_10 - Static variable in class `com.realvnc.mirrorlink.DisplayConfiguration`

Resize factor of 1/10

RESIZE_FACTOR_1_16 - Static variable in class `com.realvnc.mirrorlink.DisplayConfiguration`

Resize factor of 1/16

RESIZE_FACTOR_1_2 - Static variable in class `com.realvnc.mirrorlink.DisplayConfiguration`

Resize factor of 1/2

RESIZE_FACTOR_1_3 - Static variable in class `com.realvnc.mirrorlink.DisplayConfiguration`

Resize factor of 1/3

RESIZE_FACTOR_1_32 - Static variable in class `com.realvnc.mirrorlink.DisplayConfiguration`

Resize factor of 1/32

RESIZE_FACTOR_1_4 - Static variable in class `com.realvnc.mirrorlink.DisplayConfiguration`

Resize factor of 1/4

RESIZE_FACTOR_1_5 - Static variable in class `com.realvnc.mirrorlink.DisplayConfiguration`

Resize factor of 1/5

RESIZE_FACTOR_1_6 - Static variable in class `com.realvnc.mirrorlink.DisplayConfiguration`

Resize factor of 1/6

RESIZE_FACTOR_1_8 - Static variable in class `com.realvnc.mirrorlink.DisplayConfiguration`

Resize factor of 1/8

RESIZE_FACTOR_2_3 - Static variable in class `com.realvnc.mirrorlink.DisplayConfiguration`

Resize factor of 2_3

RESIZE_FACTOR_3_4 - Static variable in class `com.realvnc.mirrorlink.DisplayConfiguration`

Resize factor of 3_4

RESIZE_FACTOR_NONE - Static variable in class `com.realvnc.mirrorlink.DisplayConfiguration`

No supported resize factors.

restoreOrientationLock(int) - Method in class `com.realvnc.vncserver.android.VncOrientationManager`

Deprecated.

Use `VncOrientationManager.restoreOrientationLockEx(int)` instead. Requests that the screen orientation lock is restored to the requested value once the VNC connection is over.

This method can be called at any point of the VNC connection, but it is recommended to be used as soon as the connection has started, so that when the VNC connection terminates the screen orientation lock holds the same value it did before it started.

The orientation provided should be one of the constants defined in this class:

`VncOrientationManager.ORIENTATION_DISABLE_LOCK`, `VncOrientationManager.ORIENTATION_LANDSCAPE_LOCK` or `VncOrientationManager.ORIENTATION_PORTRAIT_LOCK`.

This feature was introduced in version 3.4. Older versions of the RCS default to disabling screen orientation lock at the end of a session. For compatibility, if this method is not called, that behaviour is reproduced (i.e. the orientation lock will always be restored to `VncOrientationManager.ORIENTATION_DISABLE_LOCK`).

If restoring the orientation lock is supported for the current set-up, this method will return `true`. Otherwise it will return `false`.

This method is deprecated because it will fail if invoked too early,

restoreOrientationLockCb(boolean, int) - Method in class `com.realvnc.vncserver.android.VncServerCallbackHandler`

Called to notify whether a previous `VncOrientationManager.restoreOrientationLockEx(int)` request will be honoured or not.

restoreOrientationLockEx(int) - Method in class `com.realvnc.vncserver.android.VncOrientationManager`

Requests that the screen orientation lock is restored to the requested value once the VNC connection is over.

RGB_332 - Static variable in class `com.realvnc.vncserver.core.VncPixelFormat`

A `VncPixelFormat` instance representing RGB332.

RGB_565 - Static variable in class `com.realvnc.vncserver.core.VncPixelFormat`

A `VncPixelFormat` instance representing RGB565.

RGB_888 - Static variable in class `com.realvnc.vncserver.core.VncPixelFormat`

A `VncPixelFormat` instance representing RGB888.

runningCb(VncServer) - Method in interface `com.realvnc.vncserver.android.VncServerListener`

Called when the VNC viewer and server are connected together and the VNC session has started.

S

screenSizeChangedCb(Rect) - Method in interface `com.realvnc.vncserver.android.VncDisplayInformationManager.Listener`

The screen size has changed.

sectionNames - Variable in class `com.realvnc.util.JniFile`

sections - Variable in class `com.realvnc.util.JniFile`

sendExtensionMessage(VncExtension, byte[], int, int) - Method in class `com.realvnc.vncserver.android.VncServer`

Send an extension message.

serialVersionUID - Static variable in exception `com.realvnc.util.JniFile.BadFormatException`

serialVersionUID - Static variable in exception `com.realvnc.vncserver.core.VncException`

serialVersionUID - Static variable in exception `com.realvnc.vncserver.core.VncLicenseNotValidException`

sessionPixelFormatChangedCb(VncPixelFormat) - Method in interface

`com.realvnc.vncserver.android.VncDisplayInformationManager.Listener`

The pixel format used in the RFB session has changed.

set(VncPixelFormat) - Method in class `com.realvnc.vncserver.core.VncPixelFormat`

Sets the instance to have the same values as another `VncPixelFormat` instance.

SET_STATUS_BAR_INFO - Static variable in class `com.realvnc.vncserver.android.CustomRemoteControlServiceRequests`

Disables (or re-enables) information items on the Android status bar.

SET_SYSTEM_UI_VISIBILITY - Static variable in class `com.realvnc.vncserver.android.CustomRemoteControlServiceRequests`

Requests that the visibility of the status or navigation bars be changed, for all windows across all applications.

setAuthentication(int) - Method in class `com.realvnc.vncserver.android.VncServer`

Set the type of authentication to use.

setBigEndianFlag(boolean) - Method in class `com.realvnc.vncserver.core.VncPixelFormat`

Sets the big endian flag for this pixel format.

setBitsPerPixel(int) - Method in class `com.realvnc.vncserver.core.VncPixelFormat`

Sets the bits per pixel for this pixel format.

setBlueMax(int) - Method in class `com.realvnc.vncserver.core.VncPixelFormat`

Sets the maximum value of the blue component of each pixel for this pixel format.

setBlueShift(int) - Method in class `com.realvnc.vncserver.core.VncPixelFormat`

Sets the shift value of the blue component of each pixel for this pixel format.

setClientDisplayHeightMillimeters(int) - Method in class `com.realvnc.mirrorlink.VNCCClientDisplayConfiguration`

Sets the display height size.

setClientDisplayHeightPixels(int) - Method in class `com.realvnc.mirrorlink.VNCCClientDisplayConfiguration`

Sets the display height.

setClientDisplayWidthMillimeters(int) - Method in class `com.realvnc.mirrorlink.VNCCClientDisplayConfiguration`

Sets the display width size.

setClientDisplayWidthPixels(int) - Method in class `com.realvnc.mirrorlink.VNCCClientDisplayConfiguration`

Sets the display pixel width.

setClientDistanceFromUserMillimeters(int) - Method in class `com.realvnc.mirrorlink.VNCCClientDisplayConfiguration`

Sets the expected distance from user.

setClientMajorVersion(int) - Method in class `com.realvnc.mirrorlink.VNCCClientDisplayConfiguration`

Sets the major client version.

setClientMinorVersion(int) - Method in class `com.realvnc.mirrorlink.VNCCClientDisplayConfiguration`

Sets the minor client version.

setDepth(int) - Method in class `com.realvnc.vncserver.core.VncPixelFormat`

Sets the depth for this pixel format.

setDesktopName(String) - Method in class `com.realvnc.vncserver.android.VncServer`

Sets the VNC desktop name to the given string.

setDeviceKeySupport(int) - Method in class `com.realvnc.mirrorlink.VNCCClientEventConfiguration`

Sets the device key support to the provided bitmask.

setEncryption(int) - Method in class `com.realvnc.vncserver.android.VncServer`
Set the type of encryption to use.

setFeatures(int) - Method in class `com.realvnc.mirrorlink.VNCDeviceStatus`
Sets a bitmask of device status features.

setFramebufferConfiguration(int) - Method in class `com.realvnc.mirrorlink.VNCClientDisplayConfiguration`
Sets the framebuffer configuration.

setGreenMax(int) - Method in class `com.realvnc.vncserver.core.VncPixelFormat`
Sets the maximum value of the green component of each pixel for this pixel format.

setGreenShift(int) - Method in class `com.realvnc.vncserver.core.VncPixelFormat`
Sets the shift value of the green component of each pixel for this pixel format.

setH264Encoder(VncH264Encoder, boolean) - Method in class `com.realvnc.vncserver.android.VncServer`
Register an H.264 encoder with the SDK.

setKey(byte[]) - Method in class `com.realvnc.vncserver.android.VncServer`
Set key pair to be used for authentication and encryption.

setKeyboardCountry(String) - Method in class `com.realvnc.mirrorlink.VNCClientEventConfiguration`
Sets the keyboard layout country code.

setKeyboardLanguage(String) - Method in class `com.realvnc.mirrorlink.VNCClientEventConfiguration`
Sets the keyboard layout language code.

setKnobKeySupport(int) - Method in class `com.realvnc.mirrorlink.VNCClientEventConfiguration`
Sets the knob key support to the provided bitmask.

setMiscKeySupport(int) - Method in class `com.realvnc.mirrorlink.VNCClientEventConfiguration`
Sets the miscellaneous key support to the provided bitmask.

setMultimediaKeySupport(int) - Method in class `com.realvnc.mirrorlink.VNCClientEventConfiguration`
Sets the multimedia key support to the provided bitmask.

setPixelFormatSupport(int) - Method in class `com.realvnc.mirrorlink.VNCClientDisplayConfiguration`
Sets the pixel formats supported by the client.

setPointerSupport(int) - Method in class `com.realvnc.mirrorlink.VNCClientEventConfiguration`
Sets the pointer / touchscreen support to the provided bitmap.

setRedMax(int) - Method in class `com.realvnc.vncserver.core.VncPixelFormat`
Sets the maximum value of the red component of each pixel for this pixel format.

setRedShift(int) - Method in class `com.realvnc.vncserver.core.VncPixelFormat`
Sets the shift value of the red component of each pixel for this pixel format.

setRemoteControlServicePackage(String) - Method in class `com.realvnc.vncserver.android.VncServer`
Sets the package name to use to obtain the remote control service.

setResizeFactors(int) - Method in class `com.realvnc.mirrorlink.VNCClientDisplayConfiguration`
Sets the resize factors supported by the client for use with the Transform encoding.

setTestingMode(boolean) - Static method in class `com.realvnc.util.VncLog`

setTrueColorFlag(boolean) - Method in class `com.realvnc.vncserver.core.VncPixelFormat`
Sets the true color flag for this pixel format.

setUiCountry(String) - Method in class `com.realvnc.mirrorlink.VNCClientEventConfiguration`
Sets the user interface country code.

setUiLanguage(String) - Method in class `com.realvnc.mirrorlink.VNCClientEventConfiguration`
Sets the user interface language code.

setUserData(Object) - Method in class `com.realvnc.vncserver.android.VncServer`
Set some user defined data to be associated with this server instance.

startEncoder(int, int, int, int) - Method in class `com.realvnc.vncserver.android.VncH264Encoder`
Instructs the encoder to start, with the specified resolution and H.264 level.

stopEncoder() - Method in class `com.realvnc.vncserver.android.VncH264Encoder`
Instructs the encoder to stop, and free any allocated resources.

T

toString() - Method in class `com.realvnc.mirrorlink.VNCAudioBlockingNotification`
Returns a string based representation of this object.

toString() - Method in class `com.realvnc.mirrorlink.VNCClientDisplayConfiguration`
Returns a string based representation of this object.

toString() - Method in class `com.realvnc.mirrorlink.VNCContextInformation`
Return a textual representation of this object.

toString() - Method in class `com.realvnc.mirrorlink.VNCDeviceStatus`

Retrives a textual representation of this object.

toString() - Method in class `com.realvnc.mirrorlink.VNCFramebufferBlockingNotification`

Retrieves a textual representation of this object.

toString() - Method in class `com.realvnc.mirrorlink.VNCServerDisplayConfiguration`

Retrives a textual representation of this object.

toString() - Method in class `com.realvnc.mirrorlink.VNCServerEventConfiguration`

Returns a textual representation of this object.

toString() - Method in class `com.realvnc.vncserver.android.VncSizeInt`

TRUST_LEVEL_APPLICATION_CERTIFICATE - Static variable in class `com.realvnc.mirrorlink.VNCContextInformation`

The provided data is under sole control of the VNC and UPnP server.

TRUST_LEVEL_REGISTERED_APPLICATION - Static variable in class `com.realvnc.mirrorlink.VNCContextInformation`

The provided data is under sole control of the VNC and UPnP server.

TRUST_LEVEL_SELF_REGISTERED_APPLICATION - Static variable in class `com.realvnc.mirrorlink.VNCContextInformation`

The provided data is under the control of the application.

TRUST_LEVEL_UNKNOWN - Static variable in class `com.realvnc.mirrorlink.VNCContextInformation`

The server has no trust in the reported information.

TRUST_LEVEL_USER_CONFIGURATION - Static variable in class `com.realvnc.mirrorlink.VNCContextInformation`

The provided data is under the control of the user.

TYPE - Static variable in class `com.realvnc.vncserver.core.VncCommandStringBase`

Names for known command string key/value pairs

U

uiCountry - Variable in class `com.realvnc.mirrorlink.VNCServerEventConfiguration`

Contains the user interface country code

uiLanguage - Variable in class `com.realvnc.mirrorlink.VNCServerEventConfiguration`

Contains the user interface language code.

V

valueOf(String) - Static method in enum `com.realvnc.vncserver.android.VncContextInformationManager.ListenerPriority`

Returns the enum constant of this type with the specified name.

valueOf(String) - Static method in enum `com.realvnc.vncserver.android.VncH264Encoder.FrameType`

Returns the enum constant of this type with the specified name.

values() - Static method in enum `com.realvnc.vncserver.android.VncContextInformationManager.ListenerPriority`

Returns an array containing the constants of this enum type, in the order they are declared.

values() - Static method in enum `com.realvnc.vncserver.android.VncH264Encoder.FrameType`

Returns an array containing the constants of this enum type, in the order they are declared.

VERSION - Static variable in class `com.realvnc.vncserver.core.VncCommandStringBase`

VISUAL_CONTENT_CATEGORY_CAR_MODE - Static variable in class `com.realvnc.mirrorlink.VNCContextInformation`

Content category representing car mode.

VISUAL_CONTENT_CATEGORY_GRAPHICS_3D - Static variable in class `com.realvnc.mirrorlink.VNCContextInformation`

Content category representing 3D graphics.

VISUAL_CONTENT_CATEGORY_GRAPHICS_VECTOR - Static variable in class `com.realvnc.mirrorlink.VNCContextInformation`

Content category representing vector graphics.

VISUAL_CONTENT_CATEGORY_IMAGE - Static variable in class `com.realvnc.mirrorlink.VNCContextInformation`

Content category representing images.

VISUAL_CONTENT_CATEGORY_MISC - Static variable in class `com.realvnc.mirrorlink.VNCContextInformation`

Content category representing miscellaneous content.

VISUAL_CONTENT_CATEGORY_TEXT - Static variable in class `com.realvnc.mirrorlink.VNCContextInformation`

Content category representing text.

VISUAL_CONTENT_CATEGORY_UI - Static variable in class `com.realvnc.mirrorlink.VNCContextInformation`

Content category representing user interface (e.g.

VISUAL_CONTENT_CATEGORY_UNKNOWN - Static variable in class `com.realvnc.mirrorlink.VNCContextInformation`

Content category representing unknown content.

VISUAL_CONTENT_CATEGORY_VIDEO - Static variable in class `com.realvnc.mirrorlink.VNCContextInformation`

Content category representing video.

VNC_AUTH_NONE - Static variable in class `com.realvnc.vncserver.core.VncAuthType`

No authentication is to be used.

VNC_AUTH_PASS - Static variable in class `com.realvnc.vncserver.core.VncAuthType`

Password authentication - the viewer must provide a password to be authenticated by the server.

VNC_AUTH_REV - Static variable in class `com.realvnc.vncserver.core.VncAuthType`

Reverse Authentication - the VNC server must provide either a password or a username and password to be authenticated by the viewer.

VNC_AUTH_USER_PASS - Static variable in class `com.realvnc.vncserver.core.VncAuthType`

Username and password authentication - the viewer must provide a username and password to be authenticated by the server.

VNC_ENCRYPTION_AES_128 - Static variable in class `com.realvnc.vncserver.core.VncEncryptionType`

Use 128 bit AES encryption.

VNC_ENCRYPTION_NONE - Static variable in class `com.realvnc.vncserver.core.VncEncryptionType`

Don't use encryption.

VNC_STATE_ACCEPT_REMOTE_KEY - Static variable in class `com.realvnc.vncserver.core.VncServerState`

Server is waiting for a remote key to be accepted

VNC_STATE_ACCEPTING - Static variable in class `com.realvnc.vncserver.core.VncServerState`

Server is waiting for a connection to be accepted

VNC_STATE_AUTH - Static variable in class `com.realvnc.vncserver.core.VncServerState`

Server is waiting for viewer credentials to be authenticated by the application.

VNC_STATE_AWAITING_KEY - Static variable in class `com.realvnc.vncserver.core.VncServerState`

Server is waiting for an encryption key to be set

VNC_STATE_CONNECTING - Static variable in class `com.realvnc.vncserver.core.VncServerState`

Server is initiating an outbound connection

VNC_STATE_CONNECTING_RELAY - Static variable in class `com.realvnc.vncserver.core.VncServerState`

Server is performing a data relay handshake

VNC_STATE_DISCONNECTED - Static variable in class `com.realvnc.vncserver.core.VncServerState`

Server is idle

VNC_STATE_EXITING - Static variable in class `com.realvnc.vncserver.core.VncServerState`

Server is in the process of exiting

VNC_STATE_GENERATING_KEY - Static variable in class `com.realvnc.vncserver.core.VncServerState`

Server is generating an encryption key

VNC_STATE_HANDSHAKING - Static variable in class `com.realvnc.vncserver.core.VncServerState`

Server is processing the RFB handshaking phase

VNC_STATE_LISTENING - Static variable in class `com.realvnc.vncserver.core.VncServerState`

Server is listening for an incoming connection

VNC_STATE_ML_AWAITING_CLIENT_DISPLAY_CONFIGURATION - Static variable in class `com.realvnc.vncserver.core.VncServerState`

Server is waiting for a MirrorLink 'client display configuration' message from the viewer.

VNC_STATE_ML_AWAITING_CLIENT_EVENT_CONFIGURATION - Static variable in class `com.realvnc.vncserver.core.VncServerState`

Server is waiting for a MirrorLink 'client event configuration' message from the viewer.

VNC_STATE_ML_AWAITING_SERVER_DISPLAY_CONFIGURATION - Static variable in class `com.realvnc.vncserver.core.VncServerState`

Server is waiting for a MirrorLink 'server display configuration' message from the application.

VNC_STATE_ML_AWAITING_SERVER_EVENT_CONFIGURATION - Static variable in class `com.realvnc.vncserver.core.VncServerState`

Server is waiting for a MirrorLink 'server event configuration' message from the application.

VNC_STATE_REVERSE_AUTH - Static variable in class `com.realvnc.vncserver.core.VncServerState`

Server is waiting for a reverse authentication password from the application.

VNC_STATE_RUNNING - Static variable in class `com.realvnc.vncserver.core.VncServerState`

Server is connected to a viewer

VNC_STATE_SETUP - Static variable in class `com.realvnc.vncserver.core.VncServerState`

Server is setting the parameters for the RFB session

VNCAudioBlockingNotification - Class in `com.realvnc.mirrorlink`

Class holding an AudioBlockingNotification MirrorLink extension message to be sent to the server.

VNCAudioBlockingNotification(int, int) - Constructor for class `com.realvnc.mirrorlink.VNCAudioBlockingNotification`

Construct a VNCAudioBlockingNotification object.

VNCAudioInfo - Class in `com.realvnc.mirrorlink`

Class containing constants to be used when defining audio information for an application.

VNCAudioInfo() - Constructor for class `com.realvnc.mirrorlink.VNCAudioInfo`

VncAuthType - Class in `com.realvnc.vncserver.core`

Type of authentication to be used by the VNC server.

VncBearer - Interface in `com.realvnc.vncserver.core`

Objects implementing this interface are used by the SDK to provide data transport facilities between the server and viewer.

VncBearerCallbacks - Interface in `com.realvnc.vncserver.core`

Objects implementing this interface are used by the SDK to provide a way for the bearers to call SDK provided functionality.

VncBearerInfo - Interface in `com.realvnc.vncserver.core`

Objects implementing this interface are used to provided detailed information on a pluggable bearer, and can be obtained through a call to the `getBearerInfo` method of the `VncServer` object.

VNCClientDisplayConfiguration - Class in `com.realvnc.mirrorlink`

Class holding a ClientDisplayConfiguration MirrorLink extension message to be sent to the server.

VNCClientDisplayConfiguration(int, int, int, int, int, int, int, int, int, int) - Constructor for class `com.realvnc.mirrorlink.VNCClientDisplayConfiguration`

Constructs a new client display configuration object.

VNCClientEventConfiguration - Class in `com.realvnc.mirrorlink`

Class holding a ClientEventConfiguration MirrorLink extension message to be sent to the server.

VNCClientEventConfiguration(String, String, String, String, int, int, int, int, int) - Constructor for class `com.realvnc.mirrorlink.VNCClientEventConfiguration`

Creates a new client event configuration object.

VncCommandString - Class in `com.realvnc.vncserver.android`

Android implementation encapsulating a VNC command string.

VncCommandString() - Constructor for class `com.realvnc.vncserver.android.VncCommandString`

Create a new object representing an initially empty command string.

VncCommandStringBase - Class in `com.realvnc.vncserver.core`

A abstract class for encapsulating a VNC command string.

VncCommandStringBase() - Constructor for class `com.realvnc.vncserver.core.VncCommandStringBase`

Create a new object representing an initially empty command string.

VncConnection - Interface in `com.realvnc.vncserver.core`

An object representing a connection across which the server will talk to a VNC viewer.

VNCContextInformation - Class in `com.realvnc.mirrorlink`

Class holding a decoded ContextInformation rectangle that has been received from the server.

VNCContextInformation(int, int, int, int, int, int) - Constructor for class `com.realvnc.mirrorlink.VNCContextInformation`

Constructs a new context information object.

VncContextInformationManager - Class in `com.realvnc.vncserver.android`

This class provides management of the context information for the applications, view and windows present on the display.

VncContextInformationManager() - Constructor for class `com.realvnc.vncserver.android.VncContextInformationManager`

VncContextInformationManager.AccessibilityServiceProvider - Interface in `com.realvnc.vncserver.android`

This interface represents a class that facilitates the usage of an accessibility service for context information gathering.

VncContextInformationManager.CapturedContextInformation - Interface in `com.realvnc.vncserver.android`

Interface describing the context information for an area of the device screen.

VncContextInformationManager.Listener - Interface in `com.realvnc.vncserver.android`

This interface allows objects to be notified of changes to the context information for the visual elements of the screen.

VncContextInformationManager.ListenerPriority - Enum in `com.realvnc.vncserver.android`

Enum to indicate priority of listeners.

VNCDeviceStatus - Class in `com.realvnc.mirrorlink`

Class holding a decoded DeviceStatus MirrorLink extension message that has been received from, or will be sent to, the server.

VNCDeviceStatus(int) - Constructor for class `com.realvnc.mirrorlink.VNCDeviceStatus`

Construct a DeviceStatus object.

VncDisplayInformationManager - Class in `com.realvnc.vncserver.android`

This class provides management of the display information related to the VNC session.

VncDisplayInformationManager() - Constructor for class `com.realvnc.vncserver.android.VncDisplayInformationManager`

VncDisplayInformationManager.Listener - Interface in `com.realvnc.vncserver.android`

Listener interface used by the VNC Server Display Information Manager to notify the application that some of the display information has changed.

VncEncryptionType - Class in `com.realvnc.vncserver.core`

Type of authentication to be used by the VNC server.

VncException - Exception in `com.realvnc.vncserver.core`

An exception class to describe errors using standard VNC error codes.

VncException(int) - Constructor for exception `com.realvnc.vncserver.core.VncException`

VncException(int, String) - Constructor for exception `com.realvnc.vncserver.core.VncException`

VncException(int, Exception) - Constructor for exception `com.realvnc.vncserver.core.VncException`

VncException(int, Throwable) - Constructor for exception `com.realvnc.vncserver.core.VncException`

VncException(int, String, Exception) - Constructor for exception `com.realvnc.vncserver.core.VncException`

VncException(int, String, Throwable) - Constructor for exception `com.realvnc.vncserver.core.VncException`

VncExtension - Interface in `com.realvnc.vncserver.android`

This is an opaque object used as a unique handle for an externally registered protocol extension.

VncExtensionListener - Interface in `com.realvnc.vncserver.android`

This interface is used for receiving externally defined protocol extension messages.

VncFramebufferBlockingNotification - Class in `com.realvnc.mirrorlink`

Class holding a FramebufferBlockingNotification MirrorLink extension message to be sent to the server.

VncFramebufferBlockingNotification(int, int) - Constructor for class `com.realvnc.mirrorlink.VncFramebufferBlockingNotification`

Construct a VncFramebufferBlockingNotification object.

VncFramebufferBlockingNotification(int, int, int) - Constructor for class `com.realvnc.mirrorlink.VncFramebufferBlockingNotification`

Construct a VncFramebufferBlockingNotification object.

VncH264Encoder - Class in `com.realvnc.vncserver.android`

This class can be extended to implement an H.264 encoder.

VncH264Encoder() - Constructor for class `com.realvnc.vncserver.android.VncH264Encoder`

VncH264Encoder.BufferOwner - Interface in `com.realvnc.vncserver.android`

VncH264Encoder.FrameType - Enum in `com.realvnc.vncserver.android`

VncH264Encoder.ScreenGrabHelper - Interface in `com.realvnc.vncserver.android`

VncLicenseNotValidException - Exception in `com.realvnc.vncserver.core`

An exception class to describe license not valid errors.

VncLicenseNotValidException() - Constructor for exception `com.realvnc.vncserver.core.VncLicenseNotValidException`

Constructs a license not valid exception for license errors where the serial number of the license couldn't be determined.

VncLicenseNotValidException(byte[]) - Constructor for exception `com.realvnc.vncserver.core.VncLicenseNotValidException`

Constructs a license not valid exception for license errors where the serial number of the license could be determined.

VncLog - Class in `com.realvnc.util`

VncLog() - Constructor for class `com.realvnc.util.VncLog`

VncMirrorLinkKeyEventListener - Interface in `com.realvnc.vncserver.android`

Listener interface allowing server SDK users to implement custom handling of MirrorLink key events.

VncMirrorLinkKeys - Class in `com.realvnc.mirrorlink`

VncMirrorLinkKeys

VncMirrorLinkKeys() - Constructor for class `com.realvnc.mirrorlink.VncMirrorLinkKeys`

VncOrientationManager - Class in `com.realvnc.vncserver.android`

This class provides management of the orientation of the device display.

VncOrientationManager() - Constructor for class `com.realvnc.vncserver.android.VncOrientationManager`

VncPixelFormat - Class in `com.realvnc.vncserver.core`

Defines the format of the pixels in a framebuffer.

VncPixelFormat(int, int, boolean, boolean, int, int, int, int, int) - Constructor for class `com.realvnc.vncserver.core.VncPixelFormat`

Constructs and populates a VncPixelFormat instance.

VncPixelFormat() - Constructor for class `com.realvnc.vncserver.core.VncPixelFormat`

Constructs an empty VncPixelFormat instance.

VncPixelFormat(VncPixelFormat) - Constructor for class `com.realvnc.vncserver.core.VncPixelFormat`

Copies a VncPixelFormat instance.

VncRemoteControlInfo - Interface in `com.realvnc.vncserver.android`

Objects implementing this interface are used to provide detailed information about the forms of remote control, and can be obtained through a call to the `VncServer.getRemoteControlInfo()` method.

VncRemoteFeatureCheckListener - Interface in `com.realvnc.vncserver.android`

This interface is used for receiving externally defined remote feature checks.

VncServer - Class in `com.realvnc.vncserver.android`

This class provides the API for a VNC server.

VncServer() - Constructor for class `com.realvnc.vncserver.android.VncServer`

Do not use the constructor.

VNC_SERVER_ERR_ALREADY_EXISTS - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

A custom extension with the same name has already been registered.

VNC_SERVER_ERR_BAD_CHALLENGE - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

VNC Data Relay could not authenticate the server.

VNC_SERVER_ERR_BAD_CRYPT - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

RFB protocol or AES checksum is corrupt, or VNC Mobile Viewer did not have a matching private key.

VNC_SERVER_ERR_BAD_MESSAGE - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

VNC Data Relay received an invalid message from the server.

VNC_SERVER_ERR_BAD_PIXEL_FORMAT - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

VNC Mobile Viewer specified an unsupported pixel color depth.

VNC_SERVER_ERR_BAD_PORT - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

Invalid port number.

VNC_SERVER_ERR_BAD_SESSION_ID - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

Either the command string contained an invalid VNC Data Relay session ID, or the communication channel to which it refers is no longer reserved.

VNC_SERVER_ERR_BEARER_NOT_FOUND - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

Transport mechanism specified in command string missing or corrupt.

VNC_SERVER_ERR_CAPTURE_FRAME_BUFFER_NOT_IMPLEMENTED - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

Screen capture is not implemented in this platform.

VNC_SERVER_ERR_COMMAND_FETCH_FAILED - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

HTTP or HTTPS request to command string web service failed.

VNC_SERVER_ERR_COMMAND_SUPERSEDED - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

A command string for a different remote control session is received before the device user accepts the prompt authorizing the original session.

VNC_SERVER_ERR_CONNECTION_CLOSED - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

VNC Mobile Viewer terminated the remote control session.

VNC_SERVER_ERR_CONNECTION_REFUSED - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

Port could not be contacted.

VNC_SERVER_ERR_CRITICAL_CAPABILITY_UNSUPPORTED - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

The connection has failed because the VNC Viewer does not support a capability which is critical to the operation of this server.

VNC_SERVER_ERR_DEPRECATED_FIELD_USED - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

A deprecated field has been set.

VNC_SERVER_ERR_ENVIRONMENT - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

The application environment is unsupported.

VNC_SERVER_ERR_FEATURE_NOT_LICENSED - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

The requested operation could not be completed due to the feature not being licensed.

VNC_SERVER_ERR_HOST_UNREACHABLE - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

IP address could not be contacted.

VNC_SERVER_ERR_INSUFFICIENT_BUFFER_SPACE - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

The requested operation could not be completed due to insufficient buffer space.

VNC_SERVER_ERR_INTERNAL_ERROR - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

General error.

VNC_SERVER_ERR_INVALID_COMMAND_STRING - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

Invalid command string.

VNC_SERVER_ERR_INVALID_PARAMETER - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

An invalid parameter was passed to an API call.

VNC_SERVER_ERR_KEY_GENERATION - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

The RSA key generation algorithm failed.

VNC_SERVER_ERR_KEY_TOO_BIG - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

The RSA key is too large.

VNC_SERVER_ERR_LICENSE_NOT_VALID - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

The requested operation could not be completed due to the provided license not being valid.

VNC_SERVER_ERR_LOGIN_REJECTED - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

User rejected authentication credentials.

VNCSERVER_ERR_NAME_LOOKUP_FAILED - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

Domain name could not be resolved.

VNCSERVER_ERR_NETWORK - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

General network error.

VNCSERVER_ERR_NETWORK_LOST - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

No network connection.

VNCSERVER_ERR_NO_ENCODINGS - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

VNC Mobile Viewer specified an unsupported encoding.

VNCSERVER_ERR_NO_SUITABLE_RCS - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

No suitable remote control service found.

VNCSERVER_ERR_NONE - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

No error.

VNCSERVER_ERR_NOT_LICENSED_FOR_VIEWER - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

License incompatible with that of VNC Mobile Viewer.

VNCSERVER_ERR_PEER_TIMEOUT - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

VNC Mobile Viewer did not connect to the other end of the reserved VNC Data Relay communication channel in time.

VNCSERVER_ERR_PERMISSIONS - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

Insufficient device permissions.

VNCSERVER_ERR_PORT_IN_USE - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

Port is in use.

VNCSERVER_ERR_PROTOCOL_MISMATCH - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

Protocol incompatible with that of VNC Mobile Viewer.

VNCSERVER_ERR_RCS_EXITED - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

Remote control service exited.

VNCSERVER_ERR_RCS_LACKS_PERMISSIONS - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

Remote control service does not have the required permissions.

VNCSERVER_ERR_RCS_LIBRARY_NOT_FOUND - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

Required remote control support not present.

VNCSERVER_ERR_RCS_NOT_ENABLED - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

Remote control service has not been enabled yet.

VNCSERVER_ERR_RESET - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

The server has been reset.

VNCSERVER_ERR_RESOURCES - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

Insufficient system resources.

VNCSERVER_ERR_SIGNATURE_REJECTED - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

VNC Mobile Viewer signature specified in command string not the same as that of the actual VNC Mobile Viewer that connects.

VNCSERVER_ERR_STATE - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

An invalid API call was made.

VNCSERVER_ERR_TIMED_OUT - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

A general network time-out occurred.

VNCSERVER_ERR_TOO_LOW_ANDROID_VERSION - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

The Android version is too low.

VNCSERVER_ERR_TOO_LOW_OPENGL_ES_VERSION - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

The OpenGL ES version is too low.

VNCSERVER_ERR_TOO_MANY_EXTENSIONS - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

The maximum number of custom extensions (8) have already been registered.

VNCSERVER_ERR_TOO_MANY_EXTERNAL_ENCODERS - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

Add external encoder had failed because the VNC Server limit on the number of external encoders is exceeded.

VNCSERVER_ERR_UNABLE_TO_START_SERVICE - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

The underlying VNC Mobile Server service could not be started.

VNCSERVER_ERR_UNDERLYING_LIBRARY_NOT_FOUND - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

Underlying Library Not Found.

VNCSERVER_ERR_UNSUPPORTED_AUTH - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

Invalid authentication type.

VNCSERVER_ERR_USB_NOT_CONNECTED - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

USB Not Connected.

VNCSERVER_ERR_USER_REFUSED_CONNECTION - Static variable in class `com.realvnc.vncserver.core.VncServerCoreErrors`

Device user rejected prompt authorizing remote control.

VncServerCallbackHandler - Class in [com.realvnc.vncserver.android](#)

Base class for callbacks received for a VNC server.

VncServerCallbackHandler() - Constructor for class [com.realvnc.vncserver.android.VncServerCallbackHandler](#)

VncServerCoreErrors - Class in [com.realvnc.vncserver.core](#)

VNC specific error codes to be returned from the VNC server.

VncServerCoreErrors() - Constructor for class [com.realvnc.vncserver.core.VncServerCoreErrors](#)

VNCServerDisplayConfiguration - Class in [com.realvnc.mirrorlink](#)

Class holding a decoded ServerDisplayConfiguration MirrorLink extension message that has been received from the server.

VNCServerDisplayConfiguration(int, int, int, int, int, int) - Constructor for class [com.realvnc.mirrorlink.VNCServerDisplayConfiguration](#)

Construct a VNCServerDisplayConfiguration object.

VNCServerEventConfiguration - Class in [com.realvnc.mirrorlink](#)

Class holding a decoded ServerEventConfiguration MirrorLink extension message that has been received from the server.

VNCServerEventConfiguration(String, String, String, String, int, int, int, int, int) - Constructor for class [com.realvnc.mirrorlink.VNCServerEventConfiguration](#)

Constructs a new server event configuration object.

VncServerListener - Interface in [com.realvnc.vncserver.android](#)

Listener interface used by the VNC server to notify the application that certain events have occurred.

VncServerMirrorLinkListener - Interface in [com.realvnc.vncserver.android](#)

Extension to the [VncServerListener](#) class to provide extra callbacks in relation to events using the MirrorLink protocol.

VncServerOrientationListener - Interface in [com.realvnc.vncserver.android](#)

A type of listener which can be informed of orientation changes detected by the Android VNC server SDK.

VncServerState - Class in [com.realvnc.vncserver.core](#)

Constants representing the various states that the VNC server can be in.

VncSizeInt - Class in [com.realvnc.vncserver.android](#)

Represents a width and height.

VncSizeInt(int, int) - Constructor for class [com.realvnc.vncserver.android.VncSizeInt](#)

Constructs a [VncSizeInt](#) object with the specified width and height.

VNCViewerEventConfiguration - Class in [com.realvnc.mirrorlink](#)

Deprecated.

VNCViewerEventConfiguration() - Constructor for class [com.realvnc.mirrorlink.VNCViewerEventConfiguration](#)

Deprecated.

X

XK_DEVICE_APPLICATION - Static variable in class [com.realvnc.mirrorlink.VNCMirrorLinkKeys](#)

XK_DEVICE_BACKWARD - Static variable in class [com.realvnc.mirrorlink.VNCMirrorLinkKeys](#)

XK_DEVICE_CLEAR - Static variable in class [com.realvnc.mirrorlink.VNCMirrorLinkKeys](#)

XK_DEVICE_DELETE - Static variable in class [com.realvnc.mirrorlink.VNCMirrorLinkKeys](#)

XK_DEVICE_FORWARD - Static variable in class [com.realvnc.mirrorlink.VNCMirrorLinkKeys](#)

XK_DEVICE_HOME - Static variable in class [com.realvnc.mirrorlink.VNCMirrorLinkKeys](#)

XK_DEVICE_MENU - Static variable in class [com.realvnc.mirrorlink.VNCMirrorLinkKeys](#)

XK_DEVICE_OK - Static variable in class [com.realvnc.mirrorlink.VNCMirrorLinkKeys](#)

XK_DEVICE_PHONE_CALL - Static variable in class [com.realvnc.mirrorlink.VNCMirrorLinkKeys](#)

XK_DEVICE_PHONE_END - Static variable in class [com.realvnc.mirrorlink.VNCMirrorLinkKeys](#)

XK_DEVICE_SEARCH - Static variable in class [com.realvnc.mirrorlink.VNCMirrorLinkKeys](#)

XK_DEVICE_SOFT_LEFT - Static variable in class [com.realvnc.mirrorlink.VNCMirrorLinkKeys](#)

XK_DEVICE_SOFT_MIDDLE - Static variable in class [com.realvnc.mirrorlink.VNCMirrorLinkKeys](#)

XK_DEVICE_SOFT_RIGHT - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_DEVICE_ZOOM_IN - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_DEVICE_ZOOM_OUT - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_FUNCTION_KEY_0 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_FUNCTION_KEY_1 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_FUNCTION_KEY_10 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_FUNCTION_KEY_11 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_FUNCTION_KEY_12 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_FUNCTION_KEY_2 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_FUNCTION_KEY_3 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_FUNCTION_KEY_4 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_FUNCTION_KEY_5 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_FUNCTION_KEY_6 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_FUNCTION_KEY_7 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_FUNCTION_KEY_8 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_FUNCTION_KEY_9 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_ITU_KEY_0 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_ITU_KEY_1 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_ITU_KEY_2 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_ITU_KEY_3 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_ITU_KEY_4 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_ITU_KEY_5 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_ITU_KEY_6 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_ITU_KEY_7 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_ITU_KEY_8 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_ITU_KEY_9 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_ITU_KEY_ASTERIX - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_ITU_KEY_POUND - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_KNOB_2D_ROTATE_x_0 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_KNOB_2D_ROTATE_X_0 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_KNOB_2D_ROTATE_x_1 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_KNOB_2D_ROTATE_X_1 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

[illegible]

XK_KNOB_2D_SHIFT_LEFT_0 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_KNOB_2D_SHIFT_LEFT_1 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_KNOB_2D_SHIFT_LEFT_2 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_KNOB_2D_SHIFT_LEFT_3 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_KNOB_2D_SHIFT_PULL_0 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_KNOB_2D_SHIFT_PULL_1 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_KNOB_2D_SHIFT_PULL_2 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_KNOB_2D_SHIFT_PULL_3 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_KNOB_2D_SHIFT_PUSH_0 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_KNOB_2D_SHIFT_PUSH_1 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_KNOB_2D_SHIFT_PUSH_2 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_KNOB_2D_SHIFT_PUSH_3 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_KNOB_2D_SHIFT_RIGHT_0 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_KNOB_2D_SHIFT_RIGHT_1 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_KNOB_2D_SHIFT_RIGHT_2 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_KNOB_2D_SHIFT_RIGHT_3 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_KNOB_2D_SHIFT_UP_0 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_KNOB_2D_SHIFT_UP_1 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_KNOB_2D_SHIFT_UP_2 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_KNOB_2D_SHIFT_UP_3 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_KNOB_2D_SHIFT_UP_LEFT_0 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_KNOB_2D_SHIFT_UP_LEFT_1 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_KNOB_2D_SHIFT_UP_LEFT_2 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_KNOB_2D_SHIFT_UP_LEFT_3 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_KNOB_2D_SHIFT_UP_RIGHT_0 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_KNOB_2D_SHIFT_UP_RIGHT_1 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_KNOB_2D_SHIFT_UP_RIGHT_2 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_KNOB_2D_SHIFT_UP_RIGHT_3 - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_MULTIMEDIA_FORWARD - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_MULTIMEDIA_MUTE - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_MULTIMEDIA_NEXT - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_MULTIMEDIA_PAUSE - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_MULTIMEDIA_PHOTO - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_MULTIMEDIA_PLAY - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_MULTIMEDIA_PREVIOUS - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_MULTIMEDIA_REWIND - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_MULTIMEDIA_STOP - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

XK_MULTIMEDIA_UNMUTE - Static variable in class com.realvnc.mirrorlink.VNCMirrorLinkKeys

A B C D E F G H I K L M O P Q R S T U V X