CHAPTER 6

Cisco Unified JTAPI 拡張

Cisco Unified JTAPI 拡張は、JTAPI 1.2 仕様では提供されていないものの、Cisco Unified Communications Manager の機能を利用するためのクラスとインターフェイスのセットで構成されます。開発者はこの拡張を使用して、新しいアプリケーションを作成することも、既存の拡張を変更して新しいメソッドを作成することもできます。

この章では、Cisco Unified Communications Manager で利用できる拡張 (インターフェイスとクラス) について説明し、次のセクションで構成されています。

クラスの階層

次のクラス階層は com.cisco.jtapi.extensions パッケージに含まれています。

hierarchy.java.lang.Object

- com.cisco.jtapi.extensions.CiscoAddressCallInfo
- com.cisco.jtapi.extensions.CiscoJtapiVersion
- com.cisco.jtapi.extensions.CiscoMediaCapability
  - com.cisco.jtapi.extensions.CiscoG711MediaCapability
  - com.cisco.jtapi.extensions.CiscoG723MediaCapability
  - com.cisco.jtapi.extensions.CiscoG729MediaCapability
  - com.cisco.jtapi.extensions.CiscoGSMMediaCapability
  - com.cisco.jtapi.extensions.CiscoWideBandMediaCapability
- com.cisco.jtapi.extensions.CiscoRTPParams
- java.lang.Throwable（java.io.Serializable を実装）
  - java.lang.Exception
    - com.cisco.jtapi.extensions.CiscoRegistrationException
    - com.cisco.jtapi.extensions.CiscoUnregistrationException

CiscoAddressCallInfo

クラスの履歴

Cisco Unified Communications Manager リリース	説明
7.1 (2)	変更を記録するために履歴表に追加されました。

宣言

public class CiscoAddressCallInfo extends java.lang.Object

java.lang.Object

　　com.cisco.jtapi.extensions.CiscoAddressCallInfo

コンストラクタ

CiscoAddressCallInfo (int inumActiveCalls, int imaxActiveCalls, int inumCallsOnHold, int imaxCallsOnHold)

CiscoAddressCallInfo (int inumActiveCalls, int imaxActiveCalls, int inumCallsOnHold, int imaxCallsOnHold, CiscoCall[] icalls)

フィールド

なし

メソッド

表 6-1　CiscoAddressCallInfo のメソッド

インターフェイス	メソッド	説明
CiscoCall[]	getCalls()	CiscoAddress に対する Cisco コールの配列を返します。
int	getMaxActiveCalls()	CiscoAddress でサポートされるアクティブ コールの最大数を整数で返します。
int	getMaxCallsOnHold()	CiscoAddress で保留状態にできるコールの最大数を整数で返します。
int	getNumActiveCalls()	CiscoAddress のアクティブ コールの数を整数で返します。
int	getNumCallsOnHold()	CiscoAddress で保留中のコールの数を整数で返します。

継承したメソッド

クラス java.lang.Object から
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

関連資料

なし

CiscoG711MediaCapability

CiscoG711MediaCapability オブジェクトは G.711 で符号化された RTP ストリームのプロパティを指定します。G.711 メディア終端をサポートしているアプリケーションは、CiscoMediaTerminal を登録する際に、このオブジェクトを使用して適切なパケット サイズを指定します。デフォルトのパケット サイズは 30 ミリ秒です。

クラスの履歴

Cisco Unified Communications Manager リリース	説明
7.1(x)	変更を記録するために履歴表に追加されました。

宣言

public class CiscoG711MediaCapability extends CiscoMediaCapability

java.lang.Object

　　com.cisco.jtapi.extensions.CiscoMediaCapability

　　　　com.cisco.jtapi.extensions.CiscoG711MediaCapability

コンストラクタ

表 6-2　CiscoG711MediaCapability のコンストラクタ

インターフェイス	コンストラクタ	説明
public	CiscoG711MediaCapability(int rtpPacketFrameSize)	CiscoG711MediaCapability を構築します。
public	CiscoG711MediaCapability()	CiscoG711MediaCapability を構築します。

フィールド

表 6-3　CiscoG711MediaCapability のフィールド

インターフェイス	フィールド	説明
public static final int	FRAMESIZE_TWENTY_MILLISECOND_PACKET	RTP パケットのフレームサイズ：20 ミリ秒の RTP パケット。
public static final int	FRAMESIZE_THIRTY_MILLISECOND_PACKET	RTP パケットのフレームサイズ：30 ミリ秒の RTP パケット。
public static final int	FRAMESIZE_SIXTY_MILLISECOND_PACKET	RTP パケットのフレームサイズ：60 ミリ秒の RTP パケット。

継承したフィールド

クラス com.cisco.jtapi.extensions.CiscoMediaCapability から
G711_64K_30_MILLISECONDS, G723_6K_30_MILLISECONDS, G729_30_MILLISECONDS, GSM_80_MILLISECONDS, WIDEBAND_256K_10_MILLISECONDS

メソッド

なし

継承したメソッド

クラス com.cisco.jtapi.extensions.CiscoMediaCapability から
getMaxFramesPerPacket, getPayloadType, isSupported, toString

クラス java.lang.Object から
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

関連資料

「定数フィールド値」（P.F-1）を参照してください。

CiscoG723MediaCapability

CiscoG723MediaCapability オブジェクトは G.723 で符号化された RTP ストリームのプロパティを指定します。G.723 メディア終端をサポートしているアプリケーションは、CiscoMediaTerminal を登録する際に、このオブジェクトを使用して適切なパケット サイズおよびビット レートを指定します。デフォルトのパケット サイズは 30 ミリ秒で、デフォルトのビット レートは 6.4k です。

クラスの履歴

Cisco Unified Communications Manager リリース	説明
7.1x	変更を記録するために履歴表に追加されました。

宣言

public class CiscoG723MediaCapability extends CiscoMediaCapability

java.lang.Object

　　com.cisco.jtapi.extensions.CiscoMediaCapability

　　　　com.cisco.jtapi.extensions.CiscoG723MediaCapability

コンストラクタ

表 6-4　CiscoG723MediaCapability のコンストラクタ

インターフェイス	コンストラクタ	説明
public	CiscoG723MediaCapability(int rtpPacketFrameSize, int bitRate)	CiscoG723MediaCapability を構築します

フィールド

表 6-5　CiscoG723MediaCapability のフィールド

インターフェイス	フィールド	説明
public static final int	FRAMESIZE_TWENTY_MILLISECOND_PACKET	RTP パケットのフレームサイズ：20 ミリ秒の RTP パケット。
public static final int	FRAMESIZE_THIRTY_MILLISECOND_PACKET	RTP パケットのフレームサイズ：30 ミリ秒の RTP パケット。
public static final int	FRAMESIZE_SIXTY_MILLISECOND_PACKET	RTP パケットのフレームサイズ：60 ミリ秒の RTP パケット。

継承したフィールド

クラス com.cisco.jtapi.extensions.CiscoMediaCapability から
G711_64K_30_MILLISECONDS, G723_6K_30_MILLISECONDS, G729_30_MILLISECONDS, GSM_80_MILLISECONDS, WIDEBAND_256K_10_MILLISECONDS

メソッド

表 6-6　CiscoG723MediaCapability のメソッド

インターフェイス	メソッド	説明
public int	getBitRate()	この機能オブジェクトで指定されたビット レートを返します。戻り値：RTPBitRate インターフェイスからのビット レート。
public java.lang.String	toString()	Object.toString() メソッドをオーバーライドします。オーバーライド：クラス CiscoMediaCapability の toString。

継承したメソッド

クラス com.cisco.jtapi.extensions.CiscoMediaCapability から
getMaxFramesPerPacket, getPayloadType, isSupported

クラス java.lang.Object から
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

関連資料

「定数フィールド値」（P.F-1）を参照してください。

CiscoG729MediaCapability

CiscoG729MediaCapability オブジェクトは G.729 で符号化された RTP ストリームのプロパティを指定します。G.729 メディア終端をサポートしているアプリケーションは、CiscoMediaTerminal を登録する際に、このオブジェクトを使用して適切なパケット サイズを指定します。デフォルトのパケット サイズは 30 ミリ秒です。

クラスの履歴

Cisco Unified Communications Manager リリース	説明
7.1x	変更を記録するために履歴表に追加されました。

宣言

public class CiscoG729MediaCapability extends CiscoMediaCapability

java.lang.Object

　　com.cisco.jtapi.extensions.CiscoMediaCapability

　　　　com.cisco.jtapi.extensions.CiscoG729MediaCapability

コンストラクタ

表 6-7　G729MediaCapability のコンストラクタ

コンストラクタ	説明
CiscoG729MediaCapability(int payload, int rtpPacketFrameSize)	CiscoG729MediaCapability を構築します。

フィールド

表 6-8　CiscoG729MediaCapability のフィールド

インターフェイス	フィールド	説明
static int	FRAMESIZE_SIXTY_MILLISECOND_PACKET	RTP パケットのフレームサイズ：60 ミリ秒の RTP パケット。
static int	FRAMESIZE_THIRTY_MILLISECOND_PACKET	RTP パケットのフレームサイズ：30 ミリ秒の RTP パケット。
static int	FRAMESIZE_TWENTY_MILLISECOND_PACKET	RTP パケットのフレームサイズ：20 ミリ秒の RTP パケット。
static int	FRAMESIZE_TWENTY_MILLISECOND_PACKET	RTP パケットのフレームサイズ：20 ミリ秒の RTP パケット。

継承したフィールド

クラス com.cisco.jtapi.extensions.CiscoMediaCapability から

G711_64K_30_MILLISECONDS, G723_6K_30_MILLISECONDS, G729_30_MILLISECONDS, GSM_80_MILLISECONDS, WIDEBAND_256K_10_MILLISECONDS

メソッド

なし

継承したメソッド

クラス com.cisco.jtapi.extensions.CiscoMediaCapability から

getMaxFramesPerPacket, getPayloadType, isSupported, toString

クラス java.lang.Object から

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

関連資料

「定数フィールド値」（P.F-1）を参照してください。

CiscoGSMMediaCapability

CiscoGSMMediaCapability オブジェクトは GSM で符号化された RTP ストリームのプロパティを指定します。GSM メディア終端をサポートしているアプリケーションは、CiscoMediaTerminal を登録する際に、このオブジェクトを使用して適切なパケット サイズを指定します。デフォルトのパケット サイズは 30 ミリ秒です。

クラスの履歴

Cisco Unified Communications Manager リリース	説明
7.1x	変更を記録するために履歴表に追加されました。

宣言

public class CiscoGSMMediaCapability extends CiscoMediaCapability

java.lang.Object

　　com.cisco.jtapi.extensions.CiscoMediaCapability

　　　　com.cisco.jtapi.extensions.CiscoGSMMediaCapability

コンストラクタ

表 6-9　CiscoGSMMediaCapability のコンストラクタ

インターフェイス	コンストラクタ	説明
public	CiscoGSMMediaCapability()	CiscoGSMMediaCapability を構築します。
public	CiscoGSMMediaCapability(int rtpPacketFrameSize)	CiscoGSMMediaCapability を構築します。

フィールド

表 6-10　CiscoGSMMediaCapability のフィールド

インターフェイス	フィールド	説明
static int	FRAMESIZE_EIGHTY_MILLISECOND_PACKET	RTP パケットのフレームサイズ：80 ミリ秒の RTP パケット。

継承したフィールド

クラス com.cisco.jtapi.extensions.CiscoMediaCapability から
G711_64K_30_MILLISECONDS, G723_6K_30_MILLISECONDS, G729_30_MILLISECONDS, GSM_80_MILLISECONDS, WIDEBAND_256K_10_MILLISECONDS

メソッド

なし

継承したメソッド

クラス com.cisco.jtapi.extensions.CiscoMediaCapability から
getMaxFramesPerPacket, getPayloadType, isSupported, toString

クラス java.lang.Object から
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

関連資料

なし

CiscoJtapiVersion

このクラスは、インストールされている Cisco JTAPI のバージョン情報を提供します。プログラムはアクセス用メソッドを使用してバージョン番号を取得できます。Cisco Jtapi Version は a.b(x.y) の形式で提供されます（a はメジャー バージョン、b はマイナー バージョン、x はリビジョン番号、y はビルド番号を表します）。

クラスの履歴

Cisco Unified Communications Manager リリース	説明
7.1x	変更を記録するために履歴表に追加されました。

宣言

public class CiscoJtapiVersion extends java.lang.Object

java.lang.Object

　　com.cisco.jtapi.extensions.CiscoJtapiVersion

コンストラクタ

publicCiscoJtapiVersion()None

フィールド

なし

メソッド

表 6-11　CiscoJtapiVersion のメソッド

インターフェイス	メソッド	説明
java.lang.String	getBuildDescription()	リリース バージョンの場合は「release」を返し、リリース バージョンでない場合は「debug」を返します。
int	getBuildNumber()	バージョンのビルド番号を返します。
int	getExtendedBuildNumber()	バージョンの拡張ビルド番号を返します。
int	getMajorVersion()	メジャー バージョン番号を返します。
int	getMinorVersion()	マイナー バージョン番号を返します。
int	getRevisionNumber()	バージョンのリビジョン番号を返します。
public java.lang.String	getVersion()	名前を使用せず、a.b(x.y)-z の形式でバージョン情報を返します。
public java.lang.String	toString()	a.b(x.y)-z の形式でバージョン情報を返します。クラス java.lang.Object の toString をオーバーライドします。

継承したメソッド

クラス java.lang.Object から

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

関連資料

なし

CiscoMediaCapability

CiscoMediaCapability オブジェクトは、登録された CiscoMediaTerminals に対してアプリケーションがサポートできるメディア形式のプロパティを指定します。CiscoMediaCapability は抽象クラスなので、アプリケーションはそのサブクラスだけを直接的に構築できます。

クラスの履歴

Cisco Unified Communications Manager リリース	説明
7.1x	変更を記録するために履歴表に追加されました。

宣言

public class CiscoMediaCapability extends java.lang.Object

java.lang.Object

　　com.cisco.jtapi.extensions.CiscoMediaCapability

サブクラス

CiscoG711MediaCapability, CiscoG723MediaCapability, CiscoG729MediaCapability, CiscoGSMMediaCapability, CiscoWideBandMediaCapability

コンストラクタ

表 6-12　CiscoMediaCapability のコンストラクタ

インターフェイス	コンストラクタ	説明
public	CiscoMediaCapability(int payloadType, int maxFramesPerPacket)	ペイロード タイプとパケットサイズを指定して CiscoMediaCapability オブジェクトを構築します（ミリ秒単位）。

フィールド

表 6-13　CiscoMediaCapability のフィールド

インターフェイス	フィールド	説明
static	G711_64K_30_MILLISECONDS	デフォルト パラメータを使用する G.711 機能。
static	G723_6K_30_MILLISECONDS	デフォルト パラメータを使用する G.723 機能。
static	G729_30_MILLISECONDS	デフォルト パラメータを使用する G.729 機能。
static	GSM_80_MILLISECONDS	デフォルト パラメータを使用する GSM 機能。
static	WIDEBAND_256K_10_MILLISECONDS	デフォルト パラメータを使用するワイドバンド機能。

メソッド

表 6-14　CiscoMediaCapability のメソッド

インターフェイス	メソッド	説明
int	getMaxFramesPerPacket(	このオブジェクトで指定されているパケット サイズ（ミリ秒単位）を返します。maxFramesPerPacket パラメータは、H.245 プロトコル定義を受け継いだものです。 Cisco Unified Communications Manager ではこのフィールドを、RTP パケットあたりのフレーム数ではなく、デバイスが受信可能な RTP パケットあたりのオーディオ（ミリ秒単位）として使用します。 Cisco Unified IP Phone での発着信ではこのレートを超えることができませんが、サード パーティ製の IP フォンでは、他の（より高い）レートを使用できる場合があります。
int	getPayloadType()	このオブジェクトで指定されている RTPPayload インターフェイスのペイロード タイプを返します。
boolean	isSupported()	このオブジェクトのペイロードがサポートされているかどうかを返します。payloadType がサポートされている場合に true、そうでない場合に false を返します。
java.lang.String	toString()	クラス java.lang.Object の toString をオーバーライドします。

継承したメソッド

クラス java.lang.Object から

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

関連資料

CiscoG711MediaCapability、CiscoG723MediaCapability、CiscoG729MediaCapability、CiscoGSMMediaCapability、CiscoWideBandMediaCapability、CiscoRTPBitRate、および CiscoRTPPayload を参照してください。

CiscoRegistrationException

CiscoMediaTerminal.register メソッドは、登録プロセスが失敗すると、その理由にかかわらず、この例外をスローします。たとえば、プロバイダーが OUT_OF_SERVICE である場合、またはデバイスが登録済みである場合、登録は失敗します。

クラスの履歴

Cisco Unified Communications Manager リリース	説明
7.1x	変更を記録するために履歴表に追加されました。

宣言

public class CiscoRegistrationException extends java.lang.Exception

java.lang.Object

　　java.lang.Throwable

　　　　java.lang.Exception

　　　　　　com.cisco.jtapi.extensions.CiscoRegistrationException

実装インターフェイス

java.io.Serializable

コンストラクタ

表 6-15　CiscoRegistrationException のコンストラクタ

インターフェイス	コンストラクタ	説明
public	CiscoRegistrationException(java.lang.String description)	例外の説明をパラメータとして使用します。

メソッド

なし

継承したメソッド

クラス java.lang.Throwable から

fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

クラス java.lang.Object から

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

関連資料

CiscoMediaTerminal.register(java.net.InetAddress, int, com.cisco.jtapi.extensions.CiscoMediaCapability[]) を参照してください。

CiscoRTPParams

CiscoRTPParams クラスを使用して、コールごとにメディア端末の動的 RTP アドレスとポート番号を指定できます。アプリケーションは、CiscoMediaTerminal の setRTPParams() でこのオブジェクトを渡すことができます。これらのパラメータは特定のコールだけで有効です。

クラスの履歴

Cisco Unified Communications Manager リリース	説明
7.1x	変更を記録するために履歴表に追加されました。

宣言

public class CiscoRTPParams extends java.lang.Object

java.lang.Object

コンストラクタ

CiscoRTPParams (java.net.InetAddress, rtpAddress, int rtpPort)

フィールド

なし

メソッド

表 6-16　CiscoRTPParams のメソッド

インターフェイス	メソッド	説明
java.net.InetAddress	getRTPAddress()	関連付けられたコールの RTP ストリームの受信に使用されるインターネット アドレスを返します。
java.lang.String	getRTPAddressHostName()	関連付けられたコールの RTP ストリームの受信に使用される IP ホスト名を返します。
byte[]	getRTPByteAddress()	RTP ストリームの受信に使用されるインターネット アドレスをバイト形式で返します。
int	getRTPPort()	RTP ストリームの受信に使用される UDP ポートを返します。
java.lang.String	toString()	「IP アドレス / ポート番号」の形式の文字列を返します。クラス java.lang.Object の toString をオーバーライドします。

継承したメソッド

クラス java.lang.Object から
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

関連資料

CiscoTerminal と CiscoMediaTerminal を参照してください。

CiscoUnregistrationException

CiscoMediaTerminal.unregister メソッドは、登録解除プロセスが失敗すると、この例外をスローします。たとえば、プロバイダーが OUT_OF_SERVICE である場合、または端末がすでに登録解除されている場合、登録は失敗します。

クラスの履歴

Cisco Unified Communications Manager リリース	説明
7.1x	変更を記録するために履歴表に追加されました。

宣言

public class CiscoUnregistrationException extends java.lang.Exception

java.lang.Object

java.lang.Throwable

java.lang.Exception

com.cisco.jtapi.extensions.CiscoUnregistrationException

実装インターフェイス

java.io.Serializable

コンストラクタ

表 6-17　CiscoUnregistrationException のコンストラクタ

インターフェイス	コンストラクタ	説明
public	CiscoUnregistrationException()(java.lang.String description)	なし

フィールド

なし

メソッド

なし

継承したメソッド

クラス java.lang.Throwable から

fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

クラス java.lang.Object から

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

関連資料

CiscoMediaTerminal.unregister(), Serialized Form を参照してください。

CiscoWideBandMediaCapability

CiscoWideBandMediaCapability オブジェクトは、ワイド バンドで符号化された RTP ストリームのプロパティを指定します。ワイド バンド メディア終端をサポートしているアプリケーションは、CiscoMediaTerminal を登録する際に、このオブジェクトを使用して適切なパケット サイズを指定します。デフォルトのパケット サイズは 10 ミリ秒です。

クラスの履歴

Cisco Unified Communications Manager リリース	説明
7.1x	変更を記録するために履歴表に追加されました。

宣言

public class CiscoWideBandMediaCapability extends CiscoMediaCapability

java.lang.Object

　　com.cisco.jtapi.extensions.CiscoMediaCapability

　　　　com.cisco.jtapi.extensions.CiscoWideBandMediaCapability

コンストラクタ

表 6-18　CiscoWideBandMediaCapability のコンストラクタ

インターフェイス	コンストラクタ	説明
public	CiscoWideBandMediaCapability(int packetsize)	指定されたパケット サイズの CiscoWideBandMediaCapability オブジェクトを構築します。デフォルトのパケット サイズは 10 ミリ秒です。 **パラメータ** • packetsize：RTP パケットのフレームサイズです。

フィールド

表 6-19　CiscoWideBandMedicaCapability のフィールド

インターフェイス	フィールド	説明
static int	FRAMESIZE_TEN_MILLISECOND_PACKET	RTP パケットのフレームサイズ：10 ミリ秒の RTP パケット。

継承したフィールド

クラス com.cisco.jtapi.extensions.CiscoMediaCapability から
G711_64K_30_MILLISECONDS, G723_6K_30_MILLISECONDS, G729_30_MILLISECONDS, GSM_80_MILLISECONDS, WIDEBAND_256K_10_MILLISECONDS

メソッド

なし

継承したメソッド

クラス com.cisco.jtapi.extensions.CiscoMediaCapability から
getMaxFramesPerPacket, getPayloadType, isSupported, toString

クラス java.lang.Object から
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

関連資料

「定数フィールド値」（P.F-1）を参照してください。

インターフェイスの階層

次のインターフェイスの階層は com.cisco.jtapi.extensions パッケージ階層に含まれています。

javax.telephony.Address

- com.cisco.jtapi.extensions.CiscoAddress (com.cisco.jtapi.extensions.CiscoObjectContainer も拡張)
  - com.cisco.jtapi.extensions.CiscoIntercomAddress
- javax.telephony.callcenter.RouteAddress
  - com.cisco.jtapi.extensions.CiscoRouteAddress

javax.telephony.AddressObserver

- com.cisco.jtapi.extensions.CiscoAddressObserver

javax.telephony.Call

- javax.telephony.callcontrol.CallControlCall
  - com.cisco.jtapi.extensions.CiscoCall (com.cisco.jtapi.extensions.CiscoObjectContainer も拡張)
    - com.cisco.jtapi.extensions.CiscoConsultCall

com.cisco.jtapi.extensions.CiscoCallCtlTermConnHeldReversionEv

com.cisco.jtapi.extensions.CiscoConferenceChain

com.cisco.jtapi.extensions.CiscoFeatureReason

com.cisco.jtapi.extensions.CiscoJtapiException

com.cisco.jtapi.extensions.CiscoJtapiProperties

com.cisco.jtapi.extensions.CiscoLocales

com.cisco.jtapi.extensions.CiscoMediaSecurityIndicator

com.cisco.jtapi.extensions.CiscoMediaConnectionMode

com.cisco.jtapi.extensions.CiscoMediaEncryptionAlgorithmType

com.cisco.jtapi.extensions.CiscoMediaEncryptionKeyInfo

com.cisco.jtapi.extensions.CiscoMediaSecurityIndicator

com.cisco.jtapi.extensions.CiscoMonitorInitiatorInfo

com.cisco.jtapi.extensions.CiscoMonitorTargetInfo

com.cisco.jtapi.extensions.CiscoObjectContainer

- com.cisco.jtapi.extensions.CiscoAddress (javax.telephony.Address も拡張)
  - com.cisco.jtapi.extensions.CiscoIntercomAddress
- com.cisco.jtapi.extensions.CiscoCall (javax.telephony.callcontrol.CallControlCall も拡張)
- com.cisco.jtapi.extensions.CiscoConsultCall
- com.cisco.jtapi.extensions.CiscoCallID
- com.cisco.jtapi.extensions.CiscoConnection (javax.telephony.callcontrol.CallControlConnection も拡張)
- com.cisco.jtapi.extensions.CiscoConnectionID
- com.cisco.jtapi.extensions.CiscoConsultCall
- com.cisco.jtapi.extensions.CiscoIntercomAddress
- com.cisco.jtapi.extensions.CiscoJtapiPeer (javax.telephony.JtapiPeer、com.cisco.services.tracing.TraceModule も拡張)
- com.cisco.jtapi.extensions.CiscoMediaTerminal
- com.cisco.jtapi.extensions.CiscoProvider
- com.cisco.jtapi.extensions.CiscoRouteTerminal
- com.cisco.jtapi.extensions.CiscoTerminal (javax.telephony.Terminal も拡張)
  - com.cisco.jtapi.extensions.CiscoMediaTerminal
  - com.cisco.jtapi.extensions.CiscoRouteTerminal
- com.cisco.jtapi.extensions.CiscoTerminalConnection (javax.telephony.callcontrol.CallControlTerminalConnection も拡張)

com.cisco.jtapi.extensions.CiscoPartyInfo

com.cisco.jtapi.extensions.CiscoProvFeatureID

com.cisco.jtapi.extensions.CiscoProviderCapabilityChangedEv

com.cisco.jtapi.extensions.CiscoRecorderInfo

com.cisco.jtapi.extensions.CiscoRTPBitRate

com.cisco.jtapi.extensions.CiscoRTPHandle

com.cisco.jtapi.extensions.CiscoRTPInputProperties

com.cisco.jtapi.extensions.CiscoRTPOutputProperties

com.cisco.jtapi.extensions.CiscoRTPPayload

com.cisco.jtapi.extensions.CiscoSynchronousObserver

com.cisco.jtapi.extensions.CiscoTermConnPrivacyChangedEv

com.cisco.jtapi.extensions.CiscoTermEvFilter

com.cisco.jtapi.extensions.CiscoTerminalProtocol

com.cisco.jtapi.extensions.CiscoTone

com.cisco.jtapi.extensions.CiscoUrlInfo

javax.telephony.Connection

javax.telephony.callcontrol.CallControlConnection

com.cisco.jtapi.extensions.CiscoConnection (com.cisco.jtapi.extensions.CiscoObjectContainer も拡張)

javax.telephony.events.Ev

javax.telephony.events.AddrEv

com.cisco.jtapi.extensions.CiscoAddrEv (com.cisco.jtapi.extensions.CiscoEv も拡張)

com.cisco.jtapi.extensions.CiscoAddrAutoAcceptStatusChangedEv

com.cisco.jtapi.extensions.CiscoAddrInServiceEv

com.cisco.jtapi.extensions.CiscoAddrIntercomInfoChangedEv

com.cisco.jtapi.extensions.CiscoAddrIntercomInfoRestorationFailedEv

com.cisco.jtapi.extensions.CiscoAddrOutOfServiceEv (com.cisco.jtapi.extensions.CiscoOutOfServiceEv も拡張)

com.cisco.jtapi.extensions.CiscoAddressRecordingConfigChangedEv

javax.telephony.callcontrol.events.CallCtlEv

javax.telephony.callcontrol.events.CallCtlCallEv（javax.telephony.events.CallEv も拡張）

javax.telephony.callcontrol.events.CallCtlConnEv（javax.telephony.events.ConnEv も拡張）

javax.telephony.callcontrol.events.CallCtlConnOfferedEv

com.cisco.jtapi.extensions.CiscoCallCtlConnOfferedEv

javax.telephony.events.CallEv

javax.telephony.events.CallActiveEv

com.cisco.jtapi.extensions.CiscoConsultCallActiveEv (com.cisco.jtapi.extensions.CiscoCallEv も拡張)

javax.telephony.callcontrol.events.CallCtlCallEv（javax.telephony.callcontrol.events.CallCtlEv も拡張）

javax.telephony.callcontrol.events.CallCtlConnEv（javax.telephony.events.ConnEv も拡張）

javax.telephony.callcontrol.events.CallCtlConnOfferedEv

com.cisco.jtapi.extensions.CiscoCallCtlConnOfferedEv

com.cisco.jtapi.extensions.CiscoCallEv (com.cisco.jtapi.extensions.CiscoEv も拡張)

com.cisco.jtapi.extensions.CiscoCallChangedEv
com.cisco.jtapi.extensions.CiscoCallSecurityStatusChangedEv
com.cisco.jtapi.extensions.CiscoConferenceChainAddedEv
com.cisco.jtapi.extensions.CiscoConferenceChainRemovedEv
com.cisco.jtapi.extensions.CiscoConferenceEndEv
com.cisco.jtapi.extensions.CiscoConferenceStartEv
com.cisco.jtapi.extensions.CiscoConsultCallActiveEv
(javax.telephony.events.CallActiveEv も拡張)
com.cisco.jtapi.extensions.CiscoToneChangedEv
com.cisco.jtapi.extensions.CiscoTransferEndEv
com.cisco.jtapi.extensions.CiscoTransferStartEv
javax.telephony.events.ConnEv
javax.telephony.callcontrol.events.CallCtlConnEv
（javax.telephony.callcontrol.events.CallCtlCallEv も拡張）
javax.telephony.callcontrol.events.CallCtlConnOfferedEv
com.cisco.jtapi.extensions.CiscoCallCtlConnOfferedEv
javax.telephony.events.TermConnEv
com.cisco.jtapi.extensions.CiscoTermConnMonitoringEndEv
com.cisco.jtapi.extensions.CiscoTermConnMonitoringStartEv
com.cisco.jtapi.extensions.CiscoTermConnMonitorInitiatorInfoEv
com.cisco.jtapi.extensions.CiscoTermConnMonitorTargetInfoEv
com.cisco.jtapi.extensions.CiscoTermConnRecordingEndEv
com.cisco.jtapi.extensions.CiscoTermConnRecordingStartEv
com.cisco.jtapi.extensions.CiscoTermConnRecordingTargetInfoEv
com.cisco.jtapi.extensions.CiscoTermConnSelectChangedEv
com.cisco.jtapi.extensions.CiscoEv
com.cisco.jtapi.extensions.CiscoAddrActivatedEv
com.cisco.jtapi.extensions.CiscoAddrActivatedOnTerminalEv
com.cisco.jtapi.extensions.CiscoAddrAddedToTerminalEv
com.cisco.jtapi.extensions.CiscoAddrAutoAcceptStatusChangedEv
com.cisco.jtapi.extensions.CiscoAddrCreatedEv
com.cisco.jtapi.extensions.CiscoAddrEv (javax.telephony.events.AddrEv も拡張)
com.cisco.jtapi.extensions.CiscoAddrAutoAcceptStatusChangedEv
com.cisco.jtapi.extensions.CiscoAddrInServiceEv
com.cisco.jtapi.extensions.CiscoAddrIntercomInfoChangedEv
com.cisco.jtapi.extensions.CiscoAddrIntercomInfoRestorationFailedEv
com.cisco.jtapi.extensions.CiscoAddrOutOfServiceEv
(com.cisco.jtapi.extensions.CiscoAddrEv、com.cisco.jtapi.extensions.CiscoOutOfServiceEv も拡張)

com.cisco.jtapi.extensions.CiscoAddressRecordingConfigChangedEv

com.cisco.jtapi.extensions.CiscoAddrInServiceEv

com.cisco.jtapi.extensions.CiscoAddrIntercomInfoChangedEv

com.cisco.jtapi.extensions.CiscoAddrIntercomInfoRestorationFailedEv

com.cisco.jtapi.extensions.CiscoAddrOutOfServiceEv
(com.cisco.jtapi.extensions.CiscoAddrEv も拡張)

com.cisco.jtapi.extensions.CiscoAddressRecordingConfigChangedEv

com.cisco.jtapi.extensions.CiscoAddrRemovedEv

com.cisco.jtapi.extensions.CiscoAddrRemovedFromTerminalEv

com.cisco.jtapi.extensions.CiscoAddrRestrictedEv

com.cisco.jtapi.extensions.CiscoAddrRestrictedOnTerminalEv

com.cisco.jtapi.extensions.CiscoCallChangedEv

com.cisco.jtapi.extensions.CiscoCallEv (javax.telephony.events.CallEv も拡張)

com.cisco.jtapi.extensions.CiscoCallChangedEv

com.cisco.jtapi.extensions.CiscoCallSecurityStatusChangedEv

com.cisco.jtapi.extensions.CiscoConferenceChainAddedEv

com.cisco.jtapi.extensions.CiscoConferenceChainRemovedEv

com.cisco.jtapi.extensions.CiscoConferenceEndEv

com.cisco.jtapi.extensions.CiscoConferenceStartEv

com.cisco.jtapi.extensions.CiscoConsultCallActiveEv (javax.telephony.events.CiscoCallEv も拡張)

com.cisco.jtapi.extensions.CiscoToneChangedEv

com.cisco.jtapi.extensions.CiscoTransferEndEv

com.cisco.jtapi.extensions.CiscoTransferStartEv

com.cisco.jtapi.extensions.CiscoCallSecurityStatusChangedEv

com.cisco.jtapi.extensions.CiscoConferenceChainAddedEv

com.cisco.jtapi.extensions.CiscoConferenceChainRemovedEv

com.cisco.jtapi.extensions.CiscoConferenceEndEv

com.cisco.jtapi.extensions.CiscoConferenceStartEv

com.cisco.jtapi.extensions.CiscoConsultCallActiveEv (javax.telephony.events.CallActiveEv、com.cisco.jtapi.extensions.CiscoCallEv も拡張)

com.cisco.jtapi.extensions.CiscoMediaOpenLogicalChannelEv

com.cisco.jtapi.extensions.CiscoOutOfServiceEv

com.cisco.jtapi.extensions.CiscoAddrOutOfServiceEv (com.cisco.jtapi.extensions.CiscoAddrEv も拡張)

com.cisco.jtapi.extensions.CiscoTermOutOfServiceEv (com.cisco.jtapi.extensions.CiscoTermEv も拡張)

com.cisco.jtapi.extensions.CiscoProvCallParkEv

com.cisco.jtapi.extensions.CiscoProvFeatureEv (javax.telephony.events.ProvEv も拡張)

com.cisco.jtapi.extensions.CiscoAddrActivatedEv
com.cisco.jtapi.extensions.CiscoAddrActivatedOnTerminalEv
com.cisco.jtapi.extensions.CiscoAddrAddedToTerminalEv
com.cisco.jtapi.extensions.CiscoAddrCreatedEv
com.cisco.jtapi.extensions.CiscoAddrRemovedEv
com.cisco.jtapi.extensions.CiscoAddrRemovedFromTerminalEv
com.cisco.jtapi.extensions.CiscoAddrRestrictedEv
com.cisco.jtapi.extensions.CiscoAddrRestrictedOnTerminalEv
com.cisco.jtapi.extensions.CiscoProvCallParkEv
com.cisco.jtapi.extensions.CiscoProvFeatureEv
    com.cisco.jtapi.extensions.CiscoProvCallParkEv
com.cisco.jtapi.extensions.CiscoRestrictedEv
    com.cisco.jtapi.extensions.CiscoAddrRestrictedEv
    com.cisco.jtapi.extensions.CiscoAddrRestrictedOnTerminalEv
    com.cisco.jtapi.extensions.CiscoTermActivatedEv
    com.cisco.jtapi.extensions.CiscoTermCreatedEv
    com.cisco.jtapi.extensions.CiscoTermRemovedEv
    com.cisco.jtapi.extensions.CiscoTermRestrictedEv
    com.cisco.jtapi.extensions.CiscoProvFeatureEv
    com.cisco.jtapi.extensions.CiscoProvCallParkEv
com.cisco.jtapi.extensions.CiscoRestrictedEv
    com.cisco.jtapi.extensions.CiscoAddrRestrictedEv
    com.cisco.jtapi.extensions.CiscoAddrRestrictedOnTerminalEv
com.cisco.jtapi.extensions.CiscoRTPInputKeyEv
com.cisco.jtapi.extensions.CiscoRTPInputStartedEv
com.cisco.jtapi.extensions.CiscoRTPInputStoppedEv
com.cisco.jtapi.extensions.CiscoRTPOutputKeyEv
com.cisco.jtapi.extensions.CiscoRTPOutputStartedEv
com.cisco.jtapi.extensions.CiscoRTPOutputStoppedEv
com.cisco.jtapi.extensions.CiscoTermActivatedEv
com.cisco.jtapi.extensions.CiscoTermButtonPressedEv
com.cisco.jtapi.extensions.CiscoTermCreatedEv
com.cisco.jtapi.extensions.CiscoTermDataEv
com.cisco.jtapi.extensions.CiscoTermDeviceStateActiveEv
com.cisco.jtapi.extensions.CiscoTermDeviceStateAlertingEv
com.cisco.jtapi.extensions.CiscoTermDeviceStateHeldEv
com.cisco.jtapi.extensions.CiscoTermDeviceStateWhisperEv
com.cisco.jtapi.extensions.CiscoTermDeviceStateWhisperEv

com.cisco.jtapi.extensions.CiscoTermDNDStatusChangedEv

com.cisco.jtapi.extensions.CiscoTermEvFilter (javax.telephony.events.TermEv も拡張)

com.cisco.jtapi.extensions.CiscoMediaOpenLogicalChannelEv

com.cisco.jtapi.extensions.CiscoRTPInputKeyEv

com.cisco.jtapi.extensions.CiscoRTPInputStartedEv

com.cisco.jtapi.extensions.CiscoRTPInputStoppedEv

com.cisco.jtapi.extensions.CiscoRTPOutputKeyEv

com.cisco.jtapi.extensions.CiscoRTPOutputStartedEv

com.cisco.jtapi.extensions.CiscoRTPOutputStoppedEv

com.cisco.jtapi.extensions.CiscoTermButtonPressedEv

com.cisco.jtapi.extensions.CiscoTermDataEv

com.cisco.jtapi.extensions.CiscoTermDeviceStateActiveEv

com.cisco.jtapi.extensions.CiscoTermDeviceStateAlertingEv

com.cisco.jtapi.extensions.CiscoTermDeviceStateHeldEv

com.cisco.jtapi.extensions.CiscoTermDeviceStateIdleEv

com.cisco.jtapi.extensions.CiscoTermDeviceStateWhisperEv

com.cisco.jtapi.extensions.CiscoTermDNDStatusChangedEv

com.cisco.jtapi.extensions.CiscoTermInServiceEv

com.cisco.jtapi.extensions.CiscoTermOutOfServiceEv(com.cisco.jtapi.extensions.CiscoOutOfServiceEv も拡張)

com.cisco.jtapi.extensions.CiscoTermRegistrationFailedEv

com.cisco.jtapi.extensions.CiscoTermSnapshotCompletedEv

com.cisco.jtapi.extensions.CiscoTermSnapshotEv

com.cisco.jtapi.extensions.CiscoTermInServiceEv

com.cisco.jtapi.extensions.CiscoTermOutOfServiceEv
(com.cisco.jtapi.extensions.CiscoOutOfServiceEv、com.cisco.jtapi.extensions.CiscoTermEv も拡張)

com.cisco.jtapi.extensions.CiscoTermRegistrationFailedEv

com.cisco.jtapi.extensions.CiscoTermRemovedEv

com.cisco.jtapi.extensions.CiscoTermRestrictedEv

com.cisco.jtapi.extensions.CiscoTermSnapshotCompletedEv

com.cisco.jtapi.extensions.CiscoTermSnapshotEv

com.cisco.jtapi.extensions.CiscoToneChangedEv

com.cisco.jtapi.extensions.CiscoTransferEndEv

com.cisco.jtapi.extensions.CiscoTransferStartEv

javax.telephony.events.ProvEv

com.cisco.jtapi.extensions.CiscoProvEv （com.cisco.jtapi.extensions.CiscoEv も拡張）

com.cisco.jtapi.extensions.CiscoAddrActivatedEv

com.cisco.jtapi.extensions.CiscoAddrActivatedOnTerminalEv
com.cisco.jtapi.extensions.CiscoAddrAutoAcceptStatusChangedEv
com.cisco.jtapi.extensions.CiscoAddrCreatedEv
com.cisco.jtapi.extensions.CiscoAddrRemovedEv
com.cisco.jtapi.extensions.CiscoAddrRemovedFromTerminalEv
com.cisco.jtapi.extensions.CiscoAddrRestrictedEv
com.cisco.jtapi.extensions.CiscoAddrRestrictedOnTerminalEv
com.cisco.jtapi.extensions.CiscoProvCallParkEv
com.cisco.jtapi.extensions.CiscoProvFeatureEv
com.cisco.jtapi.extensions.CiscoProvCallParkEv
com.cisco.jtapi.extensions.CiscoRestrictedEv
com.cisco.jtapi.extensions.CiscoAddrRestrictedEv
com.cisco.jtapi.extensions.CiscoAddrRestrictedOnTerminalEv
com.cisco.jtapi.extensions.CiscoTermActivatedEv
com.cisco.jtapi.extensions.CiscoTermCreatedEv
com.cisco.jtapi.extensions.CiscoTermRemovedEv
com.cisco.jtapi.extensions.CiscoTermRestrictedEv

javax.telephony.events.TermEv

com.cisco.jtapi.extensions.CiscoTermEv (com.cisco.jtapi.extensions.CiscoEv も拡張)
com.cisco.jtapi.extensions.CiscoMediaOpenLogicalChannelEv
com.cisco.jtapi.extensions.CiscoRTPInputKeyEv
com.cisco.jtapi.extensions.CiscoRTPInputStartedEv
com.cisco.jtapi.extensions.CiscoRTPInputStoppedEv
com.cisco.jtapi.extensions.CiscoRTPOutputKeyEv
com.cisco.jtapi.extensions.CiscoRTPOutputStartedEv
com.cisco.jtapi.extensions.CiscoRTPOutputStoppedEv
com.cisco.jtapi.extensions.CiscoTermButtonPressedEv
com.cisco.jtapi.extensions.CiscoTermDataEv
com.cisco.jtapi.extensions.CiscoTermDeviceStateActiveEv
com.cisco.jtapi.extensions.CiscoTermDeviceStateAlertingEv
com.cisco.jtapi.extensions.CiscoTermDeviceStateHeldEv
com.cisco.jtapi.extensions.CiscoTermDeviceStateIdleEv
com.cisco.jtapi.extensions.CiscoTermDeviceStateWhisperEv
com.cisco.jtapi.extensions.CiscoTermDNDStatusChangedEv
com.cisco.jtapi.extensions.CiscoTermInServiceEv
com.cisco.jtapi.extensions.CiscoTermOutOfServiceEv
(com.cisco.jtapi.extensions.CiscoOutOfServiceEv も拡張)
com.cisco.jtapi.extensions.CiscoTermRegistrationFailedEv

com.cisco.jtapi.extensions.CiscoTermSnapshotCompletedEv

com.cisco.jtapi.extensions.CiscoTermSnapshotEv

javax.telephony.JtapiPeer

com.cisco.jtapi.extensions.CiscoJtapiPeer (com.cisco.jtapi.extensions.CiscoObjectContainer, com.cisco.services.tracing.TraceModule も拡張)

javax.telephony.Provider

com.cisco.jtapi.extensions.CiscoProvider (com.cisco.jtapi.extensions.CiscoObjectContainer も拡張)

javax.telephony.capabilities.ProviderCapabilities

com.cisco.jtapi.extensions.CiscoProviderCapabilities

javax.telephony.ProviderObserver

com.cisco.jtapi.extensions.CiscoProviderObserver

javax.telephony.callcenter.RouteSession

com.cisco.jtapi.extensions.CiscoRouteSession

javax.telephony.callcenter.events.RouteSessionEvent

javax.telephony.callcenter.events.RouteEvent

com.cisco.jtapi.extensions.CiscoRouteEvent

javax.telephony.callcenter.events.RouteUsedEvent

com.cisco.jtapi.extensions.CiscoRouteUsedEvent

javax.telephony.Terminal

com.cisco.jtapi.extensions.CiscoTerminal (com.cisco.jtapi.extensions.CiscoObjectContainer も拡張)

com.cisco.jtapi.extensions.CiscoMediaTerminal

com.cisco.jtapi.extensions.CiscoRouteTerminal

javax.telephony.TerminalConnection

javax.telephony.callcontrol.CallControlTerminalConnection

com.cisco.jtapi.extensions.CiscoTerminalConnection (com.cisco.jtapi.extensions.CiscoObjectContainer も拡張)

javax.telephony.TerminalObserver

com.cisco.jtapi.extensions.CiscoTerminalObserver

com.cisco.services.tracing.TraceModule

com.cisco.jtapi.extensions.CiscoJtapiPeer (com.cisco.jtapi.extensions.CiscoObjectContainer, javax.telephony.JtapiPeer も拡張)

CiscoAddrActivatedEv

アドレスが制御され、「restriction（制限）」状態が active に変更された場合、CiscoAddrActivatedEv イベントがアプリケーションに送信されます。アプリケーションは、アドレスまたは関連付けられた端末が制御リストに含まれている場合に、このイベントを受信します。該当アドレスにオブザーバが存在

する場合、アプリケーションは CiscoAddrInServiceEv を受信します。オブザーバが存在しない場合、アプリケーションはオブザーバの追加を試みることが可能で、アドレスはイン サービス状態になります。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

CiscoEv, CiscoProvEv, javax.telephony.events.Ev, javax.telephony.events.ProvEv

宣言

public interface CiscoAddrActivatedEv extends CiscoProvEv

フィールド

表 6-20　CiscoAddrActivatedEv のフィールド

インターフェイス	フィールド
static int	ID

継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

表 6-21　CiscoAddrActivatedEv のメソッド

インターフェイス	メソッド	説明
javax.telephony.Address	getAddress()	有効化されたアドレスを返します。

継承したメソッド

インターフェイス javax.telephony.events.ProvEv から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

詳細については、定数フィールド値を参照してください。

CiscoCallCtlConnOfferedEv インターフェイスは CallCtlConnOfferedEv インターフェイスを拡張して、アプリケーションが発信側端末の IP アドレスを取得できるようにします。すべての発信側デバイスの IP アドレス情報が参照可能とは限りません。戻り値が 0（または null）の場合、情報が取得可能でないことを示します。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

javax.telephony.callcontrol.events.CallCtlCallEv, javax.telephony.callcontrol.events.CallCtlConnEv, javax.telephony.callcontrol.events.CallCtlConnOfferedEv, javax.telephony.callcontrol.events.CallCtlEv, javax.telephony.events.CallEv, javax.telephony.events.ConnEv, javax.telephony.events.Ev

宣言

public interface CiscoCallCtlConnOfferedEv extends
javax.telephony.callcontrol.events.CallCtlConnOfferedEv

フィールド

なし

継承したフィールド

インターフェイス javax.telephony.callcontrol.events.CallCtlConnOfferedEv から
なし

インターフェイス javax.telephony.callcontrol.events.CallCtlEv から
CAUSE_ALTERNATE, CAUSE_BUSY, CAUSE_CALL_BACK, CAUSE_CALL_NOT_ANSWERED, CAUSE_CALL_PICKUP, CAUSE_CONFERENCE, CAUSE_DO_NOT_DISTURB, CAUSE_PARK, CAUSE_REDIRECTED, CAUSE_REORDER_TONE, CAUSE_TRANSFER, CAUSE_TRUNKS_BUSY, CAUSE_UNHOLD

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

表 6-22　CiscoCallCtlConnOfferedEv のメソッド

インターフェイス	メソッド	説明
java.net.InetAddress	getCallingPartyIpAddr()	発信側の IP アドレスか、IP アドレスが取得できない場合は 0（または null）を返します。

継承したメソッド

インターフェイス javax.telephony.callcontrol.events.CallCtlCallEv から
getCalledAddress, getCallingAddress, getCallingTerminal, getLastRedirectedAddress

インターフェイス javax.telephony.callcontrol.events.CallCtlEv から
getCallControlCause

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.CallEv から
getCall

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.ConnEv から
getConnection

インターフェイス javax.telephony.events.CallEv から
getCall

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

なし

CiscoAddrActivatedOnTerminalEv

CiscoAddrActivatedOnTerminalEv イベントは、共用回線が有効になるか、または共用回線を持つ端末が有効になると送信されます。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

CiscoEv, CiscoProvEv, javax.telephony.events.Ev, javax.telephony.events.ProvEv

宣言

public interface CiscoAddrActivatedOnTerminalEv extends CiscoProvEv

フィールド

なし

継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

表 6-23　CiscoAddrActivatedOnTerminalEv のメソッド

インターフェイス	メソッド	説明
javax.telephony.Address	getAddress()	端末で制限解除とマークされているアドレスを返します。
javax.telephony.Terminal	getTerminal()	アドレスが有効になっている（制限解除とマークされている）端末を返します。

継承したメソッド

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.ProvEv から
getProvider

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

詳細については、定数フィールド値を参照してください。

CiscoAddrAddedToTerminalEv

次の場合に、CiscoAddrAddedToTerminalEv が送信されます。

- 共用回線が含まれている制御リストにユーザが端末を追加すると、このイベントがアプリケーションに送信されます。ユーザが制御リストにアドレスを持っている場合、制御リスト内にあるのと同じアドレスを持つ新しい端末を追加すると、このイベントが送信されます。
- エクステンション モビリティ（EM）ユーザが共用回線を持つプロファイルで端末にログインすると、このイベントによって新しい端末が既存のアドレスに追加されたことが通知されます。
- 新しい共用回線がユーザの制御リストに追加されると、このイベントがアプリケーションに送信されます。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

CiscoEv, CiscoProvEv, javax.telephony.events.Ev, javax.telephony.events.ProvEv

宣言

public interface CiscoAddrAddedToTerminalEv extends CiscoProvEv

フィールド

表 6-24　CiscoAddrAddedToTerminalEv のフィールド

インターフェイス	フィールド
static int	ID

継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

表 6-25　CiscoAddrAddedToTerminalEv のメソッド

インターフェイス	メソッド	説明
javax.telephony.Address	getAddress()	新しい端末が追加されるアドレスを返します。
javax.telephony.Terminal	getTerminal()	アドレスに追加される端末を返します。

継承したメソッド

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.ProvEv から
getProvider

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

詳細については、定数フィールド値を参照してください。

CiscoAddrAutoAcceptStatusChangedEv

CiscoAddrAutoAcceptStatusChangedEv は、端末のアドレスの AutoAccept ステータスが変更されるたびにアプリケーションに送信されます。アドレスに複数の端末がある場合、このイベントは個々の端末のアドレスの AutoAccept ステータスごとに送信されます。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

javax.telephony.events.AddrEv, CiscoAddrEv, CiscoEv, javax.telephony.events.Ev

宣言

public interface CiscoAddrAutoAcceptStatusChangedEv extends CiscoAddrEv

フィールド

表 6-26　CiscoAddrAutoAcceptStatusChangedEv のフィールド

インターフェイス	フィールド
static int	ID

継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUTUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_R_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

表 6-27　CiscoAddrAutoAcceptStatusChangedEv のメソッド

インターフェイス	メソッド	説明
int	getAutoAcceptStatus()	端末上のアドレスの AutoAccept ステータスを返します。CiscoAddress.AUTOACCEPT_OFF または CiscoAddress.AUTOACCEPT_ON を返します。
CiscoTerminal	getTerminal()	このアドレスの AutoAccept ステータスが変更される端末を返します。

継承したメソッド

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.AddrEv から
getAddress

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

getAutoAcceptStatus および CiscoAddress.getAutoAcceptStatus(Terminal terminal) を参照してください。

CiscoAddrCreatedEv

CiscoAddrCreatedEv イベントは、アドレスがプロバイダー ドメインに追加されるときに送信されます。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

CiscoEv, CiscoProvEv, javax.telephony.events.Ev, javax.telephony.events.ProvEv

宣言

public interface CiscoAddrCreatedEv extends CiscoProvEv

フィールド

表 6-28　CiscoAddrCreatedEv のフィールド

インターフェイス	フィールド
ID	static final int ID

継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

表 6-29　CiscoAddrCreatedEv のメソッド

インターフェイス	メソッド	説明
getAddress	javax.telephony.Address getAddress()	プロバイダー ドメインに追加されたアドレスを返します。プロバイダー ドメインに追加されるアドレスを返します。

継承したメソッド

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.ProvEv から
getProvider

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

「定数フィールド値」（P.F-1）を参照してください。

CiscoAddress

CiscoAddress インターフェイスは、Cisco Unified Communications Manager の機能を追加することによって、アドレス インターフェイスを拡張します。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	電話だけをサポートする Message Waiting Indication（MWI；メッセージ受信通知）機能が拡張され、音声とファックスのメッセージ数が追加されました。

スーパーインターフェイス

javax.telephony.Address, CiscoObjectContainer

サブインターフェイス

CiscoIntercomAddress

フィールド

表 6-30　CiscoAddress のフィールド

インターフェイス	フィールド	説明
Static int	APPLICATION_CONTROLLED_RECORDING	アプリケーションによって制御される録音は、アドレスに設定されます。
Static int	AUTO_RECORDING	自動録音は、アドレスに設定されます。
Static int	AUTOACCEPT_OFF	AutoAccept はオフです。
Static int	AUTOACCEPT_ON	AutoAccept はオンです。
Static int	AUTOANSWER_OFF	AutoAnswer はオフです。
Static int	AUTOANSWER_UNKNOWN	AutoAnswer のステータスは不明です。
Static int	AUTOANSWER_WITHHEADSET	ヘッドセットを使用した AutoAnswer は許可されます。
static int	AUTOANSWER_WITHSPEAKERSET	スピーカー セットを使用した AutoAnswer は許可されます。
static int	EXTERNAL	これは有効な名前の外部アドレスを表します。
static int	EXTERNAL_UNKNOWN	これは不明な名前の外部アドレスを表します。
static int	IN_SERVICE	アドレスがイン サービスです。
static int	INTERNAL	これは内部アドレスです。
static int	MONITORING_TARGET	これはモニタリングのターゲットまたはエージェントのあるアドレスを表します。
static int	NO_RECORDING	アドレスに対して録音がオフになっています。
static int	OUT_OF_SERVICE	アドレスがアウトオブサービスです。
static int	RINGER_DEFAULT	呼び出し音のステータスを設定された値に変更します。

表 6-30　CiscoAddress のフィールド

インターフェイス	フィールド	説明
static int	RINGER_DISABLE	アドレスの呼び出し音を無効にします。
static int	RINGER_ENABLE	アドレスの呼び出し音を有効にします。
static int	UNKNOWN	これは不明な名前のアドレスを表します。

メソッド

表 6-31　CiscoAddress のメソッド

インターフェイス	メソッド	説明
void	clearCallConnections()	このインターフェイスを使用して、アドレス上からファントム コールをクリアします。 **例外** javax.telephony.PrivilegeViolationException— このインターフェイスを使用して、アドレス上からファントム コールをクリアします。
CiscoAddressCallInfo	getAddressCallInfo (javax.telephony.Terminal terminal)	このインターフェイスを使用して、端末に存在するコールに関する情報を取得します。
int	getAutoAcceptStatus(javax.telephony.Terminal terminal)	端末上のアドレスの AutoAccept ステータスを返します。 **例外** javax.telephony.PlatformException, javax.telephony.InvalidStateException, javax.telephony.MethodNotSupportedException 端末 繧Ãアドレスの AutoAccept ステータスを返します。次のいずれかの定数を返します。 • CiscoAddress.AUTOACCEPT_OFF • CiscoAddress.AUTOACCEPT_ON **事前条件** (this.getProvider()).getState() == Provider.IN_SERVICE getState() == IN_SERVICE **パラメータ** • terminal：AutoAccepts の端末

表 6-31　CiscoAddress のメソッド （続き）

インターフェイス	メソッド	説明
int	getAutoAnswerStatus(javax.telephony.Terminal term)	このインターフェイスは、指定された端末のこのアドレスの AutoAnswer ステータスを返します。 **例外** javax.telephony.PlatformException, javax.telephony.InvalidStateException, javax.telephony.MethodNotSupportedException 返される値が AUTOANSWER_OFF の場合、AutoAnswer が無効であることを意味します。返される値が AUTOANSWER_WITHHEADSET の場合、HEADSET で AutoAnswer が有効であることを意味します。返される値が AUTOANSWER_WITHSPEAKERSET の場合、SPEAKERSET で AutoAnswer が有効であることを意味します。返される値は AUTOANSWER_UNKNOWN の場合、AutoAnswer ステータスが UNKNOWN であることを意味します。 **事前条件** (this.getProvider()).getState() == Provider.IN_SERVICE getState() == IN_SERVICE **パラメータ** • term：AutoAnswer がオンになっている端末。 次のいずれかの値を返します。 • CiscoAddress.AUTOANSWER_OFF • CiscoAddress.AUTOANSWER_ WITHHEADSET • CiscoAddress.AUTOANSWER_ WITHSPEAKERSET • CiscoAddress.AUTOANSWER_UNKNOWN **例外** javax.telephony.InvalidStateException：プロバイダーまたはアドレスが「イン サービス」ではありません。 javax.telephony.PlatformException：アドレスが Terminal term にない場合 javax.telephony.MethodNotSupportedException：アドレスが外部アドレスの場合
javax.telephony.Terminal[]	getInServiceAddrTerminals()	このインターフェイスを使用して、イン サービスである共用回線を検出します。共用回線では、複数の端末で同じアドレスが現れます。 戻り値：Terminal[] — このアドレスがイン サービス状態にある端末の配列。
java.lang.String	getPartition()	アドレスに関連付けられたパーティションを返します。

表 6-31　CiscoAddress のメソッド （続き）

インターフェイス	メソッド	説明
int	getRecordingConfig(javax.telephony.Terminal term)	このアドレスに設定された録音タイプを返します。 **例外** javax.telephony.PlatformException,javax.telephony.InvalidStateException, javax.telephony.MethodNotSupportedException **戻り値** • int：このアドレスに設定された録音タイプを返します。 • CiscoAddess.NO_RECORDING：コールを録音できません。 • CiscoAddress.AUTO_RECORDING：Unified CM はこのアドレスで応答したすべてのコールを録音します。 • CiscoAddress.APPLICATION_CONTROLLED_RECORDING：アプリケーションが録音を開始したときだけコールが録音されます。 **例外** javax.telephony.InvalidStateException：プロバイダーまたはアドレスが「イン サービス」ではありません。 javax.telephony.PlatformException：アドレスが Terminal term にない場合 javax.telephony.MethodNotSupportedException：アドレスが外部アドレスの場合
int	getRegistrationState()	**推奨されません。** このメソッドは getState() メソッドに置き換えられました。次の定数のいずれかの可能性があるこのアドレスの状態を返します。 • CiscoAddress.OUT_OF_SERVICE • CiscoAddress.IN_SERVICE
javax.telephony.Terminal[]	getRestrictedAddrTerminals()	このアドレスが制限されている端末の配列を返します。共用回線の場合、端末上のいくつかの回線が制限されている可能性があります。 アプリケーションは、制限されたアドレスのコール イベントは受信できません。制限されたアドレスが他の制御端末とのコールに関わっている場合、制限されたアドレス用の接続が作成されますが、制限されたアドレス用の TerminalConnection はありません。 戻り値：Terminal[] — このアドレスが制限されている端末の配列。制限されている端末がない場合、このメソッドは null を返します。

表 6-31　CiscoAddress のメソッド （続き）

インターフェイス	メソッド	説明
int	getState()	このアドレスの状態を返します。状態を表す定数は次のとおりです。 • CiscoAddress.OUT_OF_SERVICE • CiscoAddress.IN_SERVICE
int	getType()	次のアドレス定数を返します。 • CiscoAddress.INTERNAL • CiscoAddress.EXTERNAL • CiscoAddress.EXTERNAL_UNKNOWN • CiscoAddress.UNKNOWN • CiscoAddress.MONITORING_TARGET
boolean	isRestricted(javax.telephony.Terminal terminal)	このメソッドは、端末上のこのアドレスが制限されている場合、true を返し、制限されていない場合、false を返します。

表 6-31　CiscoAddress のメソッド （続き）

インターフェイス	メソッド	説明
void	setAutoAcceptStatus(int autoAcceptStatus, javax.telephony.Terminal terminal)	このメソッドでは、アプリケーションが CiscoMediaTerminal や CiscoRouteTerminal 上のアドレスに対して AutoAccept を有効にできるようになります。 **例外** javax.telephony.PlatformException, javax.telephony.InvalidStateException, javax.telephony.MethodNotSupportedException このメソッドでは、アプリケーションが CiscoMediaTerminal や CiscoRouteTerminal 上のアドレスに対して AutoAccept を有効にできるようになります。 CiscoMediaTerminal または CiscoRouteTerminal 以外の CiscoTerminal 上のアドレスの AutoAccept は常にオンになります。パラメータとして渡された端末が CiscoMediaTerminal や CiscoRouteTerminal でない場合、このメソッドは例外をスローします。 CiscoTerminal とアドレスを共用している CiscoMediaTerminal の場合は、CiscoMediaTerminal で AutoAccept を有効にすることを推奨します。 **事前条件** (this.getProvider()).getState() == Provider.IN_SERVICE getState() == IN_SERVICE **事後条件** 自動応答のステータスを有効または無効に設定します。 **パラメータ** • autoAcceptStatus：CiscoAddress.AUTOACCEPT_OFF または CiscoAddress.AUTOACCEPT_ON のいずれかにできます。autoAcceptStatus が AUTOACCEPT_ON の場合、端末のアドレスに対して AutoAccept が有効にされます。autoAcceptStatus が AUTOACCEPT_OFF の場合、端末のアドレスに対して AutoAccept が無効にされます。 • terminal：AutoAccept が有効にされる端末。 **例外** javax.telephony.InvalidStateException：プロバイダーまたはアドレスが「イン サービス」ではありません。 javax.telephony.PlatformException：この端末にはこのアドレスがありません。 javax.telephony.MethodNotSupportedException：端末が CiscoMediaTerminal または CiscoRouteTerminal でない場合。

表 6-31　CiscoAddress のメソッド （続き）

インターフェイス	メソッド	説明
void	setMessageWaiting(java.lang.String destination, boolean enable)	destination で指定されたアドレスのメッセージ受信インジケータを有効にするか無効にするかを指定します。enable が true の場合は、メッセージ受信が有効になります（すでに有効になっている場合はそのまま）。enable が false の場合は、メッセージ受信が無効になります（すでに無効になっている場合はそのまま）。 **例外** javax.telephony.MethodNotSupportedException, javax.telephony.InvalidStateException,javax.telephony.PrivilegeViolationException **事前条件** (this.getProvider()).getState() == Provider.IN_SERVICE **事後条件** 有効/無効のステータスに応じて、メッセージ受信インジケータを有効または無効にします。 **（注）** この実装は現在、CallControlAddress で次のように指定されている事後条件を実現しません。 this.getMessageWaiting() == enable このアドレスに CallCtlAddrMessageWaitingEv が配信されます。 **パラメータ** • destination：メッセージ受信インジケータを有効または無効にする DN/アドレス。 • enable：メッセージ受信を有効にする場合は true、無効にする場合は false。 **例外** • javax.telephony.MethodNotSupportedException：このメソッドは指定された実装でサポートされていません。 javax.telephony.InvalidStateException **（注）** プロバイダーが「イン サービス」ではありません。 javax.telephony.PrivilegeViolationException **（注）** プロバイダーのユーザに、この宛先のメッセージ待ちインジケータを起動する権限がありません。

表 6-31　CiscoAddress のメソッド（続き）

インターフェイス	メソッド	説明
void	setRingerStatus(int status)	このアドレスの呼び出し音ステータスを変更します。 **例外** javax.telephony.MethodNotSupportedException, javax.telephony.InvalidStateException, javax.telephony.InvalidArgumentException 次に示す定数のうちの 1 つを受け入れます。 • CiscoAddress.RINGER_DEFAULT • CiscoAddress.RINGER_DISABLE • CiscoAddress.RINGER_ENABLE
void	setMessageSummary (boolean enable, boolean voiceCounts, int totalNewVoiceMsgs, int totalOldVoiceMsgs, boolean highPriorityVoiceCounts, int newHighPriorityVoiceMsgs, int oldHighPriorityVoiceMsgs, boolean faxCounts, int totalNewFaxMsgs, int totalOldFaxMsgs, boolean highPriorityFaxCounts, int newHighPriorityFaxMsgs, int oldHighPriorityFaxMsgs)	このインターフェイスを使用して、メッセージ受信インジケータと音声 / ファックスのメッセージ受信数を設定します。enable が true の場合は、メッセージ受信が有効になります（すでに有効になっている場合はそのまま）。enable が false の場合は、メッセージ受信が無効になります（すでに無効になっている場合はそのまま）。 **事前条件** (this.getProvider()).getState() == Provider.IN_SERVICE **事後条件** メッセージ受信インジケータを有効または無効にして、メッセージ受信数を設定します。

表 6-31　CiscoAddress のメソッド （続き）

インターフェイス	メソッド	説明
		パラメータ • enable：メッセージ受信を有効にする場合は true、無効にする場合は false。 • voiceCounts：音声メッセージ数が指定されているかどうかを示します。 • totalNewVoiceMsgs：新しい音声メッセージ受信の合計数を指定します。 • totalOldVoiceMsgs：古い音声メッセージ受信の合計数を指定します。 • highPriorityVoiceCounts：優先度の高い音声メッセージ数 ™ 指定されているかどうかを示します。 • newHighPriorityVoiceMsgs：優先度の高い新しい音声メッセージ受信の合計数を指定します。 • oldHighPriorityVoiceMsgs：優先度の高い古い音声メッセージ受信の合計数を指定します。 • faxCounts：ファックス メッセージ数が指定されているかどうかを示します。 • totalNewFaxMsgs：新しいファックス メッセージ受信の合計数を指定します。 • totalOldFaxMsgs：古いファックス メッセージ受信の合計数を指定します。 • highPriorityFaxCounts：優先度の高いファックス メッセージ数が指定されているかどうかを示します。 • newHighPriorityFaxMsgs：優先度の高い新しいファックス メッセージ受信の合計数を指定します。 • oldHighPriorityFaxMsgs：優先度の高い古いファックス メッセージ受信の合計数を指定します。 例外 javax.telephony.MethodNotSupportedException：このメソッドは指定された実装でサポートされていません。 javax.telephony.InvalidStateException：プロバイダーが「インサービス」ではありません。 javax.telephony.PrivilegeViolationException：ユーザに、この宛先のメッセージ受信インジケータを設定する権限がありません。

表 6-31　CiscoAddress のメソッド（続き）

インターフェイス	メソッド	説明
void	setMessageSummary(java.lang.String destination, boolean enable, boolean voiceCounts, int totalNewVoiceMsgs, int totalOldVoiceMsgs, boolean highPriorityVoiceCounts, int newHighPriorityVoiceMsgs, int oldHighPriorityVoiceMsgs, boolean faxCounts, int totalNewFaxMsgs, int totalOldFaxMsgs, boolean highPriorityFaxCounts, int newHighPriorityFaxMsgs, int oldHighPriorityFaxMsgs)	このインターフェイスを使用して、宛先によって指定されたアドレスに対するメッセージ受信インジケータと音声 / ファックスのメッセージ受信数を設定します。 **事前条件** (this.getProvider()).getState() == Provider.IN_SERVICE **事後条件** メッセージ受信インジケータを有効または無効にして、メッセージ受信数を設定します。 **パラメータ** • destination：メッセージ待機インジケータを有効または無効にする DN/ アドレス。 • enable：メッセージ受信を有効にする場合は true、無効にする場合は false。 • voiceCounts：音声メッセージ数が指定されているかどうかを示します。 • totalNewVoiceMsgs：新しい音声メッセージ受信の合計数を指定します。 • totalOldVoiceMsgs：古い音声メッセージ受信の合計数を指定します。 • highPriorityVoiceCounts：優先度の高い音声メッセージ数が指定されているかどうかを示します。 • newHighPriorityVoiceMsgs：優先度の高い新しい音声メッセージ受信の合計数を指定します。 • oldHighPriorityVoiceMsgs：優先度の高い古い音声メッセージ受信の合計数を指定します。 • faxCounts：ファックス メッセージ数が指定されているかどうかを示します。 • totalNewFaxMsgs：新しいファックス メッセージ受信の合計数を指定します。 • totalOldFaxMsgs：古いファックス メッセージ受信の合計数を指定します。 • highPriorityFaxCounts：優先度の高いファックス メッセージ数が指定されているかどうかを示します。 • newHighPriorityFaxMsgs：優先度の高い新しいファックス メッセージ受信の合計数を指定します。 • oldHighPriorityFaxMsgs：優先度の高い古いファックス メッセージ受信の合計数を指定します。

継承したメソッド

インターフェイス javax.telephony.Address から
addCallObserver, addObserver, getAddressCapabilities, getCallObservers, getCapabilities, getConnections, getName, getObservers, getProvider, getTerminals, removeCallObserver, removeObserver

インターフェイス com.cisco.jtapi.extensions.CiscoObjectContainer から
getObject, setObject

関連資料

詳細については、「定数フィールド値」（P.F-1）を参照してください。

CiscoAddressObserver

アプリケーションは、Address.addObserver メソッドを使用してアドレスを監視する際に、このインターフェイスを実装して CiscoAddrInServiceEv や CiscoAddrOutOfServiceEv などの CiscoAddrEv イベントを受信します。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

javax.telephony.AddressObserver

宣言

public interface CiscoAddressObserver extends javax.telephony.AddressObserver

フィールド

なし

メソッド

なし

継承したメソッド

インターフェイス javax.telephony.AddressObserver から
addressChangedEvent

関連資料

詳細については、CiscoAddrInServiceEv と CiscoAddrOutOfServiceEv を参照してください。

CiscoAddrEv

JTAPI のコアである javax.telephony.events.AddrEv インターフェイスを拡張する CiscoAddrEv インターフェイスは、Cisco によって拡張されたすべての JTAPI アドレス イベントの基本インターフェイスになります。このパッケージのアドレス関連イベントはすべて、直接的または間接的にこのインターフェイスを拡張します。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

javax.telephony.events.AddrEv, CiscoEv, javax.telephony.events.Ev

サブインターフェイス

CiscoAddrAutoAcceptStatusChangedEv, CiscoAddrInServiceEv, CiscoAddrIntercomInfoChangedEv, CiscoAddrIntercomInfoRestorationFailedEv, CiscoAddrOutOfServiceEv, CiscoAddrRecordingConfigChangedEv

宣言

public interface CiscoAddrEv extends CiscoEv, javax.telephony.events.AddrEv

フィールド

なし

継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

なし

継承したメソッド

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.AddrEv から
getAddress

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

詳細については、javax.telephony.events.AddrEv を参照してください。

CiscoAddrCreatedEv

アドレス イベントのフィルタを設定するためにアプリケーションに提供された CiscoAddrEvFilter。アプリケーションは次の API を使用して、アドレスに対するイベント通知を受信するためのフィルタを有効または無効に設定したり、またはフィルタに設定された値を確認したりできます。アプリケーションはこのフィルタを有効にすることができます。新しいイベントを受信する場合（CiscoAddrParkStatusEv）、デフォルトではその他のイベントに対するフィルタ値が true です。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1)	パーク モニタリングおよび Assisted DPark サポート機能に対してこのイベントが追加されました。

フィールド

なし

メソッド

表 6-32　CiscoAddrEvFilter のメソッド

インターフェイス	メソッド	説明
boolean	getCiscoAddrParkStatusEvFilter()	アプリケーションはこの API を起動して、CiscoAddrParkStatusEv に対するフィルタのステータスを確認できます。返されるデフォルト値は false です。
Void	setCiscoAddrParkStatusEvFilter (Boolean filterValue)	アプリケーションはこの API を起動して、CiscoAddrParkStatusEv に対するフィルタのステータスを設定できます。
boolean	getCiscoAddrIntercomInfo ChangedEvFilter()	アプリケーションはこの API を起動して、CiscoAddrIntercomInfoChangedEv に対するフィルタのステータスを確認できます。デフォルト値は true です。
void	setCiscoAddrIntercomInfo ChangedEvFilter(boolean filter value)	アプリケーションはこの API を起動して、CiscoAddrIntercomInfoChangedEv に対するフィルタのステータスを設定できます。
boolean	getCiscoAddrIntercomInfo RestorationFailedEvFilter()	アプリケーションはこの API を起動して、CiscoAddrIntercomInfoRestorationFailedEv に対するフィルタのステータスを確認できます。デフォルト値は true です。
void	setCiscoAddrIntercomInfoRestoration FailedEvFilter(boolean filter value)	アプリケーションはこの API を起動して、CiscoAddrIntercomInfoRestorationFailedEv に対するフィルタのステータスを設定できます。

表 6-32　CiscoAddrEvFilter のメソッド （続き）

インターフェイス	メソッド	説明
boolean	getCiscoAddrRecordingConfig ChangedEvFilter()	アプリケーションはこの API を起動して、CiscoAddrRecordingConfigChangedEv に対するフィルタのステータスを取得できます。デフォルト値は true です。
void	setCiscoAddrRecordingConfig ChangedEvFilter(boolean filter value)	アプリケーションはこの API を起動して、CiscoAddrRecordingConfigChangedEv に対するフィルタの値を設定できます。

継承したメソッド

なし

パラメータ

set メソッドはブール値をパラメータとして使用します。

値の範囲

get メソッドはブール値（true または false）を返します。

関連資料

「定数フィールド値」（P.F-1）を参照してください。

CiscoAddrInServiceEv

CiscoAddrInServiceEv はアドレスが現在、IN_SERVICE であることを示します。共用回線（複数の端末で同じアドレスが表示される）では、アプリケーションがすべての端末に対して複数の CiscoAddressInService イベントを受け取ることがあります。アプリケーションはこのインターフェイスを使用して、アドレス（または共用回線）が IN_SERVICE になる端末を検出できます。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

javax.telephony.events.AddrEv, CiscoAddrEv, CiscoEv, javax.telephony.events.Ev

宣言

public interface CiscoAddrInServiceEv extends CiscoAddrEv

フィールド

表 6-33　CiscoAddrInService のフィールド

インターフェイス	フィールド
Static int	ID

継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

表 6-34　CiscoAddrInService のメソッド

インターフェイス	メソッド	説明
getTerminal	CiscoTerminal getTerminal()	このアドレスが IN_SERVICE になる端末を返します。

継承したメソッド

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.AddrEv から
getAddress

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

詳細については、CiscoAddress.getInServiceAddrTerminals() と「定数フィールド値」（P.F-1）を参照してください。

CiscoAddrIntercomInfoChangedEv

CiscoIntercomAddress のターゲットの DN またはインターコム ターゲット ラベルが変更されると、CiscoAddrIntercomInfoChangedEv イベントがアプリケーションに送信されます。このイベントは、CiscoIntercomAddress に追加されたすべてのアプリケーション オブザーバに送信されます。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

javax.telephony.events.AddrEv, CiscoAddrEv, CiscoEv, javax.telephony.events.Ev

宣言

public interface CiscoAddrIntercomInfoChangedEv extends CiscoAddrEv

フィールド

表 6-35　CiscoAddrIntercomInfoChangedEv のフィールド

インターフェイス	フィールド
Static Int	ID

継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

表 6-36　CiscoAddrIntercomInfoChangedEv のメソッド

インターフェイス	メソッド	説明
getIntercomAddress	getIntercomAddress()	情報が変更されたインターコムのアドレスを返します。

継承したメソッド

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.AddrEv から
getAddress

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

詳細については、CiscoAddrEv と「定数フィールド値」（P.F-1）を参照してください。

CiscoAddrIntercomInfoRestorationFailedEv

フェールオーバーまたはフェールバックの発生時に、アプリケーションが設定したインターコム アドレスのインターコム ターゲット DN または CiscoIntercomAddress のインターコム ターゲット ラベルを JTAPI が復元できなかった場合に、CiscoAddrIntercomInfoRestorationFailedEv イベントがアプリケーションに送信されます。このイベントは、インターコム ターゲット DN またはインターコム ターゲット ラベルを設定したアプリケーションのためのアプリケーション オブザーバで提供されます。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

javax.telephony.events.AddrEv, CiscoAddrEv, CiscoEv, javax.telephony.events.Ev

宣言

public interface CiscoAddrIntercomInfoRestorationFailedEv extends CiscoAddrEv

フィールド

表 6-37　CiscoAddrIntercomInfoRestorationFailedEv のフィールド

インターフェイス	フィールド
Static int	ID

継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE,

CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

表 6-38　CiscoAddrIntercomInfoRestorationFailedEv のメソッド

インターフェイス	メソッド	説明
CiscoIntercomAddress	getIntercomAddress()	このインターフェイスは、情報の復元が失敗したインターコムの Cisco IntercomAddress を返します。

継承したメソッド

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.AddrEv から
getAddress

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

詳細については、「定数フィールド値」（P.F-1）と CiscoAddrEv を参照してください。

CiscoAddrOutOfServiceEv

CiscoAddrOutOfServiceEv イベントは、アドレスが OUT_OF_SERVICE になったことをアプリケーションに通知します。共用回線（複数の端末で同じアドレスが表示される）では、アプリケーションがすべての端末に対して複数の CiscoAddrOutOfServiceEv イベントを受け取ることがあります。アプリケーションはこのインターフェイスを使用して、アドレス（または共用回線）が OUT_OF_SERVICE になる端末を検出できます。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

javax.telephony.events.AddrEv, CiscoAddrEv, CiscoEv, CiscoOutOfServiceEv, javax.telephony.events.Ev

宣言

public interface CiscoAddrOutOfServiceEv extends CiscoAddrEv, CiscoOutOfServiceEv

フィールド

表 6-39　CiscoAddrOutOfServiceEv のフィールド

インターフェイス	フィールド
Static int	ID

継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス com.cisco.jtapi.extensions.CiscoOutOfServiceEv から
CAUSE_CALLMANAGER_FAILURE, CAUSE_CTIMANAGER_FAILURE, CAUSE_DEVICE_FAILURE, CAUSE_DEVICE_RESTRICTED, CAUSE_DEVICE_UNREGISTERED, CAUSE_LINE_RESTRICTED, CAUSE_NOCALLMANAGER_AVAILABLE, CAUSE_REHOME_TO_HIGHER_PRIORITY_CM, CAUSE_REHOMING_FAILURE

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

表 6-40　CiscoAddrOutOfServiceEv のメソッド

インターフェイス	メソッド	説明
CiscoTerminal	getTerminal()	このアドレスが OUT_OF_SERVICE になる端末を返します。

継承したメソッド

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.AddrEv から
getAddress

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

詳細については、「定数フィールド値」（P.F-1）と CiscoAddress.getInServiceAddrTerminals() を参照してください。

CiscoAddrParkStatusEv

Cisco Unified IP Phone を使用してコールをパークする場合、JTAPI はこのイベントを使用してパークの状態を報告します。これは、パークが起動されたアドレスにアドレスのオブザーバが追加されたすべてのアプリケーションに報告されます。このイベントは、Cisco Unified IP Phone からパークが起動された場合にだけ配信されます。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	パーク モニタリングおよび Assisted DPark 機能のためのインターフェイスが追加されました。

宣言

public interface CiscoAddrParkStatusEv extends CiscoAddrEv

フィールド

表 6-41　CiscoAddrParkStatusEv のフィールド

インターフェイス	フィールド	説明
static int	PARKED	コールがパークされるときのパークのステータス。
static int	REMINDER	パーク モニタリング復帰タイマーの期限が切れたときのパークのステータス。
static int	RETRIEVED	パークされたコールがパーク元または第三者によって取得されるときのパークのステータス。
static int	FORWARDED	Park Monitoring Forward-No-retrieve タイマーの期限が切れて、パークされたコールが転送されるときのパークのステータス。
static int	ABANDONED	パークされたコールが接続解除されるときのパークのステータス。

継承したフィールド

なし

メソッド

表 6-42　CiscoAddrParkStatusEv のメソッド

インターフェイス	メソッド	説明
int	getParkState()	パークされたコールの現在のパーク状態を返します。
int	getTransactionID()	パークされた特定のコールに固有の ID を返します。パークされた同じコールに対する複数のパーク状態のトランザクション ID は同じです。

表 6-42　CiscoAddrParkStatusEv のメソッド

インターフェイス	メソッド	説明
CiscoCallID	getCiscoCallID()	CiscoCallID を返します。
String	getParkDN()	コールがパークされている DN を返します。
String	getParkDNPartition()	パーク DN のパーティションを返します。
String	getParkedParty()	パークされた通話者の DN を返します。
String	getParkedPartyPartition()	パークされた通話者のパーティションを返します。
Terminal	getTerminal()	このイベントが配信されたアドレスの端末を返します。

値の範囲

フィールドの値は次のとおりです。

- PARKED: 2
- REMINDER: 3
- RETRIEVED: 4
- ABANDONED: 5
- FORWARDED: 6

関連資料

詳細については、定数フィールド値を参照してください。

CiscoAddrRecordingConfigChangedEv

CiscoAddrRecordingConfigChangedEv イベントは、アドレスの録音設定が変更されると、アドレスのオブザーバに送信されます。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

javax.telephony.events.AddrEv, CiscoAddrEv, CiscoEv, javax.telephony.events.Ev

宣言

public interface CiscoAddrRecordingConfigChangedEv extends CiscoAddrEv

フィールド

表 6-43　CiscoAddrRecordingConfigChangedEv のフィールド

インターフェイス	フィールド
static int	ID

継承したフィールド

インターフェイス javax.telephony.events.Ev から

CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から

CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

表 6-44　CiscoAddrRecordingConfigChangedEv のメソッド

インターフェイス	メソッド	説明
Int	getRecordingConfig()	このアドレスの新しい録音設定を返します。値は次のいずれかになります。 • CiscoAddress.NO_RECORDING • CiscoAddress.AUTO_RECORDING • CiscoAddress.APPLICATION_CONTROLLED_RECORDING
javax.telephony.Terminal	getTerminal()	録音設定が変更された端末を返します。

継承したメソッド

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.AddrEv から
getAddress

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

詳細については、「定数フィールド値」（P.F-1）と CiscoAddrEv を参照してください。

CiscoAddrRemovedEv

JTAPI は、アドレスがプロバイダー ドメインから削除されるときに CiscoAddrRemovedEv イベントを送信します。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

CiscoEv, CiscoProvEv, javax.telephony.events.Ev, javax.telephony.events.ProvEv

宣言

public interface CiscoAddrRemovedEv extends CiscoProvEv

フィールド

表 6-45　CiscoAddrRemovedEv のフィールド

インターフェイス	フィールド
static int	ID

継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

表 6-46　CiscoAddrRemovedEv のメソッド

フィールド	メソッド	説明
javax.telephony.Address	getAddress()	プロバイダー ドメインから削除されるアドレスと、ユーザ制御リストから削除されるアドレスを返します。

継承したメソッド

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.ProvEv から
getProvider

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

詳細については、「定数フィールド値」（P.F-1）を参照してください。

CiscoAddrRemovedFromTerminalEv

CiscoAddrRemovedFromTerminalEv イベントは、次の状況で送信されます。

- 管理者が共用回線の含まれるユーザ制御リストから端末を削除した場合、このイベントがアプリケーションに送信されます。
- エクステンション モビリティ（EM; Extension Mobility）ユーザが共用回線の含まれているプロファイルを持つ端末からログアウトすると、このイベントにより、端末の 1 つが既存のアドレスから削除されることが通知されます。
- ユーザ制御リスト内の端末から共用回線が削除された場合。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

CiscoEv, CiscoProvEv, javax.telephony.events.Ev, javax.telephony.events.ProvEv

宣言

public interface CiscoAddrRemovedFromTerminalEv extends CiscoProvEv

フィールド

表 6-47　CiscoAddrRemovedFromTerminalEv のフィールド

インターフェイス	フィールド
Static int	ID

継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

表 6-48　CiscoAddrRemovedFromTerminalEv のメソッド

インターフェイス	メソッド	説明
javax.telephony.Address	getAddress()	端末から削除されたアドレスを返します。
javax.telephony.Terminal	getTerminal()	アドレスから削除された端末を返します。

継承したメソッド

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.ProvEv から
getProvider

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

詳細については、「定数フィールド値」（P.F-1）を参照してください。

CiscoAddrRestrictedEv

アドレスが監視され、Restriction 状態が Restricted に変更された場合、このイベントがアプリケーションに送信されます。

アプリケーションは、アドレスまたは関連付けられた端末が Cisco Unified Communications Manager Administration で制限されている場合、このイベントを受信します。制限された回線では、アドレスが OUT_OF_SERVICE になり、再び有効になるまで IN_SERVICE に戻りません。アドレスが制限されている場合は、addCallObserver および addObserver によって例外がスローされます。

共用回線では、いくつかの回線だけが制限されて残りは制限されなかった場合、例外はスローされませんが、制限された共用回線はイベントを受け取りません。すべての共用回線が制限された場合は、オブザーバを追加すると、アプリケーションは例外のスローを試行します。オブザーバを追加した後にアドレスが制限された場合、アプリケーションは CiscoAddrOutOfServiceEv を受信し、アドレスが有効にされると IN_SERVICE になります。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

CiscoEv, CiscoProvEv, CiscoRestrictedEv, javax.telephony.events.Ev, javax.telephony.events.ProvEv

宣言

public interface CiscoAddrRestrictedEv extends CiscoRestrictedEv

フィールド

なし

継承したフィールド

インターフェイス com.cisco.jtapi.extensions.CiscoRestrictedEv から
CAUSE_UNKNOWN, CAUSE_UNSUPPORTED_DEVICE_CONFIGURATION, CAUSE_UNSUPPORTED_PROTOCOL, CAUSE_USER_RESTRICTED

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

表 6-49　CiscoAddrRestrictedEv のメソッド

インターフェイス	メソッド	説明
javax.telephony.Address	getAddress()	Cisco Unified Communications Manager で Restricted 状態に変更されたアドレスを返します。

継承したメソッド

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.ProvEv から
getProvider

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

詳細については、「定数フィールド値」（P.F-1）を参照してください。

CiscoAddrRestrictedOnTerminalEv

ユーザが制御リストに共用回線を持っており、これらの回線のいずれかが Cisco Unified CM で制限付きとしてマークされている場合、このイベントが送信されます。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

CiscoEv, CiscoProvEv, CiscoRestrictedEv, javax.telephony.events.Ev, javax.telephony.events.ProvEv

宣言

public interface CiscoAddrRestrictedOnTerminalEv extends CiscoRestrictedEv

フィールド

表 6-50　CiscoAddrRestrictedOnTerminalEv のフィールド

インターフェイス	フィールド
Static int	ID

継承したフィールド

インターフェイス com.cisco.jtapi.extensions.CiscoRestrictedEv から
CAUSE_UNKNOWN, CAUSE_UNSUPPORTED_DEVICE_CONFIGURATION, CAUSE_UNSUPPORTED_PROTOCOL, CAUSE_USER_RESTRICTED

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING,

META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

表 6-51　CiscoAddrRestricedOnTerminalEv のメソッド

インターフェイス	メソッド	説明
javax.telephony.Address	getAddress()	制限されるアドレスを返します。
javax.telephony.Terminal	getTerminal()	アドレスが制限される端末を返します。

継承したメソッド

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.ProvEv から
getProvider

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

詳細については、「定数フィールド値」（P.F-1）を参照してください。

CiscoCall

CiscoCall インターフェイスは、Cisco Unified Communications Manager に固有の機能を追加することによって、CallControlCall インターフェイスを拡張します。

Cisco Unified Communications Manager では、すべての Call オブジェクトは、グローバル コール ハンドルという共通識別子を共有する一連のコール レッグで構成されています。JTAPI では、Connection オブジェクトがコール レッグを表し、一連の接続を関連付ける Call オブジェクトに、それらのコール レッグに共通するグローバル コール ハンドルが含まれます。

CiscoCall 内のグローバル コール ハンドルは、CallManagerID プロパティと CallID プロパティを使用してアクセスできます。CallManagerID と CallID が組み合わされてグローバル コール ハンドルが形成され、Cisco Unified Communications Manager によって管理されます。このプロパティのペアは、すべての ACTIVE Call オブジェクトを通じて、常に一意になると見なしますが、ACTIVE だったコールが INACTIVE になると、新たに作成された Call オブジェクトを識別するために、その CallManagerID と CallID が再利用される場合があります。したがって、現在 INACTIVE の Call オブジェクトと、現在 ACTIVE の Call オブジェクトが、同じ CallManagerID プロパティと CallID プロパティを持つ場合があります。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1)	任意の通話者のドロップ（Drop Any Party）機能に新しいメソッド isConference() が追加されました。

スーパーインターフェイス

javax.telephony.Call, javax.telephony.callcontrol.CallControlCall, CiscoObjectContainer

サブインターフェイス

CiscoConsultCall

宣言

public interface CiscoCall extends javax.telephony.callcontrol.CallControlCall, CiscoObjectContainer

フィールド

表 6-52　CiscoCall のフィールド

インターフェイス	フィールド	説明
Static int	CALLSECURITY_AUTHENTICATED	コールのセキュリティ ステータスが認証されています。
Static int	CALLSECURITY_ENCRYPTED	コールのセキュリティ ステータスが暗号化されています。

表 6-52　CiscoCall のフィールド （続き）

インターフェイス	フィールド	説明
Static int	CALLSECURITY_NOTAUTHENTICATED	コールのセキュリティ ステータスが認証されていません。
Static int	CALLSECURITY_UNKNOWN	コールのセキュリティ ステータスが不明です。
Static int	FEATUREPRIORITY_EMERGENCY	機能プライオリティが緊急状態（emergency）です。
Static int	FEATUREPRIORITY_NORMAL	機能プライオリティが通常（normal）です。
Static int	FEATUREPRIORITY_URGENT	機能プライオリティが緊急（urgent）です。
Static int	PLAYTONE_BOTHLOCALANDREMOTE	このオプションが使用される場合、発信者とモニタリング ターゲット（エージェント）の両方に対してトーンが再生されます。
Static int	PLAYTONE_LOCALONLY	このオプションが使用される場合、モニタリング ターゲット（エージェント）だけに対してトーンが再生されます。
Static int	PLAYTONE_NOLOCAL_OR_REMOTE	このオプションが使用される場合、モニタリング ターゲット（エージェント）または発信者に対してトーンが再生されます。
Static int	PLAYTONE_REMOTEONLY	このオプションが使用される場合、発信者だけに対してトーンが再生されます。
Static int	SILENT_MONITOR	このオプションは、サイレント モニタリングが要求されたことを示します。

継承したフィールド

インターフェイス javax.telephony.Call から
ACTIVE, IDLE, INVALID

メソッド

表 6-53　CiscoCall のメソッド

インターフェイス	メソッド	説明
Void	conference(javax.telephony.Call[]otherCalls)	このインターフェイスは、複数のコールをまとめた結果の、1 つのコールに対して発呼されるすべてのコールの参加者の共有体になります。
java.lang.boolean	isConference()	電話会議のコールである場合は true を返します。そうでない場合は false を返します。

表 6-53　CiscoCall のメソッド （続き）

インターフェイス	メソッド	説明
javax.telephony.Connection[]	connect (javax.telephony.Terminal origterm, javax.telephony.Address origaddr java.lang.String.dialedDigits int featurePriority)	このメソッドは Call.connect() をオーバーロードします。新しいパラメータ featurePriority を受け付けます。featurePriority パラメータは、次のいずれかになります。 • CiscoCall.FEATUREPRIORITY_NORMAL • CiscoCall.FEATUREPRIORITY_URGENT • CiscoCall.FEATUREPRIORITY_ EMERGENCY 例外： javax.telephony.ResourceUnavailableException, javax.telephony.PrivilegeViolationException, javax.telephony.InvalidPartyException, javax.telephony.InvalidArgumentException, javax.telephony.InvalidStateException, javax.telephony.MethodNotSupportedException
boolean	getCalledAddressPI()	getCalledAddressPI に関連付けられた Presentation Indicator（PI; プレゼテーション インジケータ）を返します。
CiscoPartyInfo	getCalledPartyInfo()	コールの着側の PartyInfo を返します。
CiscoCallID	getCallID()	CallID は、同じ CallManagerID を持つすべての ACTIVE コールを通じて、一意の識別子です。
boolean	getCallingAddressPI()	getCallingAddressPI に関連付けられた Presentation Indicator（PI）を返します。
int	getCallSecurityStatus()	このインターフェイスはコールの SecurityStatus を返します。
CiscoConference Chain	getConferenceChain()	このコールがチェーニングされた電話会議の場合、CiscoConferenceChain オブジェクトを返します。
javax.telephony.Address	getCurrentCalledAddress()	コールの現在の着側のアドレスを返します。
boolean	getCurrentCalledAddressPI()	CurrentCalledAddress に関連付けられた Presentation Indicator（PI）を返します。
boolean	getCurrentCalledDisplayNamePI()	getCurredCalledDisplayNamePI に関連付けられた Presentation Indicator（PI）を返します。
java.lang.String	getCurrentCalledPartyDisplayName()	このインターフェイスはコールの着側の表示名を返します。
CiscoPartyInfo	getCurrentCalledPartyInfo()	コールの現在の着側の PartyInfo を返します。

表 6-53　CiscoCall のメソッド （続き）

インターフェイス	メソッド	説明
java.lang.String	getCurrentCalledPartyUnicodeDisplayName()	コールの着側の Unicode 表示名を返します。
int	getCurrentCalledPartyUnicodeDisplayNamelocale()	現在のコールの着側の Unicode 表示名のロケールを返します。
javax.telephony.Address	getCurrentCallingAddress()	コールの現在の着側のアドレスを返します。
boolean	getCurrentCallingAddressPI()	getCurrentCallingAddressPI に関連付けられた Presentation Indicator（PI）を返します。
boolean	getCurrentCallingDisplayNamePI()	getCurrentCalledDisplayNamePI に関連付けられた Presentation Indicator（PI）を返します。
java.lang.String	getCurrentCallingPartyDisplayName()	コールの発側の表示名を返します。
CiscoPartyInfo	getCurrentCallingPartyInfo()	現在のコールの発側の PartyInfo を返します。
java.lang.String	getCurrentCallingPartyUnicodeDisplayName()	コールの発側の Unicode 表示名を返します。
int	getCurrentCallingPartyUnicodeDisplayNamelocale()	現在のコールの着側の Unicode 表示名のロケールを返します。
java.lang.String	getGlobalizedCallingParty()	これは globalizedCallingParty を返します。
CiscoPartyInfo	getLastRedirectedPartyInfo()	コールを最後にリダイレクトした通話者の PartyInfo を返します。
boolean	getLastRedirectingAddressPI()	getLastRedirectingAddressPI に関連付けられた Presentation Indicator（PI）を返します。
CiscoPartyInfo	getLastRedirectingPartyInfo()	推奨されません。getLastRedirectedPartyInfo() を使用してください。
javax.telephony.Address	getModifiedCalledAddress()	このインターフェイスは、着信側トランスフォーメーション パターンまたはその他の方法を使用してコールの着側が変更された場合、変更された着信側アドレスを返します。
javax.telephony.Address	getModifiedCallingAddress()	アプリケーションが selectRoute API またはその他の方法を使用して発信者を変更した場合、このインターフェイスは変更された発信側アドレスを返します。
javax.telephony.Connection[]	startMonitor(javax.telephony.Terminal MonitorInitiatorterminal, javax.telephony.Address MonitorInitiatoraddress, int monitorTargetcallid, java.lang.String monitorTargetDN, java.lang.String monitorTargetTerminalName, int monitorType, int playToneDirection)	アプリケーションがモニタリングのターゲットのコールについての情報を持っている場合、このアプリケーションはこのインターフェイスを使用してコールをモニタリングできます。

表 6-53　CiscoCall のメソッド （続き）

インターフェイス	メソッド	説明
javax.telephony.Connection[]	startMonitor(javax.telephony.Terminal MonitorInitiatorterminal, javax.telephony.Address MonitorInitiator address, javax.telephony.TerminalConnection term ConnofMonitorTarget, int monitorType, int PlayToneDirection)	アプリケーションがモニタリング ターゲット（エージェント）のアドレスを監視している場合、このアプリケーションはモニタリング ターゲット（エージェント）の端末接続を使用して、モニタリング要求を開始できます。
javax.telephony.Connection	transfer(java.lang.String address, java.lang.String facCode, java.lang.String cmcCode)	このメソッドは、転送アドレスでこれらのコードがコールをオファーすることを要求される場合に facCode (Forced Authorization Code) と cmcCode (Client Matter Code) も受け付けることを除いて、CallControlCall.transfer(String address) インターフェイスと同じです。

（注）　Cisco Unified JTAPI 実装では、CallControlCall.getCalledAddress() がコールの最初の着側、つまりオリジナルの着側を返します。

継承したメソッド

インターフェイス javax.telephony.callcontrol.CallControlCall から

addParty, conference, consult, consult, drop, getCalledAddress, getCallingAddress, getCallingTerminal, getConferenceController, getConferenceEnable, getLastRedirectedAddress, getTransferController, getTransferEnable, offHook, setConferenceController, setConferenceEnable, setTransferController, setTransferEnable, transfer, transfer

インターフェイス interface javax.telephony.Call から

addObserver, connect, getCallCapabilities, getCapabilities, getConnections, getObservers, getProvider, getState, removeObserver

インターフェイス com.cisco.jtapi.extensions.CiscoObjectContainer から

getObject, setObject

パラメータ

- origterm -
- origaddr -
- dialedDigits -
- featurePriority -

会議コントローラ

会議機能が動作するためには、コントローラの端末上の共通の参加者の TerminalConnection として表される、共通の参加者がすべてのコールに属している必要があります。これらの TerminalConnection は、会議コントローラと呼ばれています。コントローラの端末上のコールの 1 つの TerminalConnection だけが CallControlTerminalConnection.TALKING 状態になるため、セカンダリ コール上の TerminalConnection は CallControlTerminalConnection.HELD 状態にある必要があります。このメソッドの呼び出しの結果、すべての会議コントローラ TerminalConnection が 1 つの TerminalConnection にマージされます。

アプリケーションは、CallControlCalll.setConferenceController() メソッドを呼び出して会議コントローラを設定することによって電話会議が設定されたときに、会議コントローラとして動作する端末を設定できます。CalControlCall.getConferenceController() メソッドは、現在の会議コントローラを返します。ない場合は null を返します。最初に会議コントローラが設定されていない場合、会議機能が起動されると、実装により適切な TerminalConnection が選択されます。

電話コールの引数

このメソッドに引数として渡されたセカンダリ コールの参加者は、このメソッドが起動されたコールにすべて移動されます。つまり、セカンダリ コールの参加者の新しい接続と TerminalConnections が、このコールに対して作成されます。セカンダリ コール上の接続と TerminalConnection はそのコールから削除され、コールは Call.INVALID 状態に移行します。

ほかの共用参加者

指定された会議コントローラ以外に、いくつかのコールの一部であるその他のアドレスと端末がある可能性があります。これらのインスタンスでは、両方のコールで共用される参加者は、1 つにマージされます。つまり、このコールでは接続と TerminalConnection が変更されずそのまま維持されます。セカンダリ コールの対応する接続と TerminalConnection は、そのコールから削除されます。

事前条件

1. tc1 をこの Call の会議コントローラにする
2. connection1 = tc1.getConnection() にする
3. tc2 ～ tcN を otherCalls の会議コントローラにする
4. (this.getProvider()).getState() == Provider.IN_SERVICE
5. this.getState() == Call.ACTIVE
6. tc1.getTerminal() == tc2.getTerminal()...=tcN.getTerminal
7. tc1.getCallControlState() == CallControlTerminalConnection.TALKING/HELD
8. tc2-tcN.getCallControlState() == CallControlTerminalConnection.HELD/TALKING
9. this != otherCalls

事後条件

1. (this.getProvider()).getState() == Provider.IN_SERVICE
2. this.getState() == Call.ACTIVE
3. otherCall.getState() == INVALID
4. otherCall からマージされる Connection を c[] にする

5. otherCall からマージされる TerminalConnection を tc[] にする
6. このコールに対して作成される新しい一連の接続を new(c) にする
7. このコールに対して作成される新しい一連の TerminalConnection を new(tc) にする
8. this.getConnections() の new(c) 要素
9. new(c).getCallState() == c.getCallState()
10. (this.getConnections()).getTerminalConnections() の new(tc) 要素
11. new(tc).getCallState() == tc.getCallState()
12. c[i].getCallControlState() == CallControlConnection.DISCONNECTED（すべての i に対して）
13. tc[i].getCallControlState() == CallControlTerminalConnection.DROPPED（すべての i に対して）
14. CallInvalidEv が otherCall に対して配信される
15. CallCtlConnDisconnectedEv/ConnDisconnectedEv がすべての c[i] に対して配信される
16. CallCtlTermConnDroppedEv/TermConnDroppedEv がすべての tc[i] に対して配信される
17. ConnCreatedEv がすべての new(c) に対して配信される
18. TermConnCreatedEv がすべての new(tc) に対して配信される
19. 適切なイベントがすべての new(c) および new(tc) に対して配信される

パラメータ

otherCalls：このコール オブジェクトとマージされるその他のコール。

例外

javax.telephony.InvalidArgumentException：指定されたコール オブジェクトが会議に有効ではありません。

javax.telephony.InvalidStateException：プロバイダーが「イン サービス」でないか、コールが「アクティブ」でないか、会議コントローラが正しい状態でないかのいずれかです。

javax.telephony.MethodNotSupportedException：この実装では、このメソッドをサポートしません。

javax.telephony.PrivilegeViolationException：アプリケーションに、このメソッドを起動する適切な権限がありません。

javax.telephony.ResourceUnavailableException：これは、このメソッドの正常な起動に必要な内部リソースがないことを意味します。

関連項目

ConnCreatedEv, TermConnCreatedEv, ConnDisconnectedEv, TermConnDroppedEv, CallInvalidEv, CallCtlConnDisconnectedEv, CallCtlTermConnDroppedEv

javax.telephony.Connection transfer(java.lang.String address java.lang.String facCode, java.lang.String cmcCode)

connect(Terminal, Address, String, CiscoRTPParams) の例外

javax.telephony.InvalidArgumentException, javax.telephony.InvalidStateException, javax.telephony.InvalidPartyException, javax.telephony.MethodNotSupportedException, javax.telephony.PrivilegeViolationException, javax.telephony.ResourceUnavailableExceptionこのメソッドは、転送アドレスでこれらのコードがコールをオファーすることを要求される場合に facCode (Forced Authorization Code) と cmcCode (Client Matter Code) も受け付けることを除いて、CallControlCall.transfer(String address) インターフェイスと同じです。1 つのコードだけが必要な場合、他のコードは null 値にする必要がある場合があります。

ユーザがコードを入力しない場合、または無効なコードを入力した場合、コールが提供されない可能性や、platformException に次のエラー コードが含まれる可能性があります。

CiscoJTAPIException.CTIERR_FAC_CMC_REASON_FAC_NEEDED
CiscoJTAPIException.CTIERR_FAC_CMC_REASON_CMC_NEEDED
CiscoJTAPIException.CTIERR_FAC_CMC_REASON_FAC_CMC_NEEDED
CiscoJTAPIException.CTIERR_FAC_CMC_REASON_FAC_INVALID
CiscoJTAPIException.CTIERR_FAC_CMC_REASON_CMC_INVALID

このメソッドのオーバーロードされたバージョンは、このコールの現在のすべての参加者を、転送コントローラ参加者の例外と一緒に、別のアドレスに転送します。転送機能が別のコールを保留し、転送を同時に実行するため、これは、「シングル ステップ転送」とも呼ばれます。このメソッドへのアドレス文字引数が有効であり、完全である必要があります。

転送コントローラ

このバージョンのこのメソッドの転送コントローラは、このコールの転送を実行し、転送の完了後にコールからドロップ オフする参加者を表します。転送コントローラは、CallControlTerminalConnection.TALKING 状態にする必要のある TerminalConnection です。

アプリケーションは、CallControlCall.setTransferController() メソッドを介する転送コントローラとして動作する TerminalConnection を制御できます。CallControlCall.getTransferController() メソッドは、現在の転送コントローラを返します。ない場合は null を返します。最初に転送コントローラが設定されていない場合、転送機能が起動されると、実装により適切な TerminalConnection が選択されます。

転送機能が起動されると、転送コントローラが CallControlTerminalConnection.DROPPED 状態になります。これが接続に関連付けられた唯一の TerminalConnection の場合、その接続も CallControlConnection.DISCONNECTED 状態になります。

新しい接続

このメソッドは、コールが転送された側を示す新しい接続を作成し、返します。コールがプロバイダードメインの外に転送され、記録できなくなった可能性がある場合、この接続が null になることがあります。この接続は少なくとも CallControlConnection.IDLE 状態である必要があります。接続の状態はこのメソッドが返す前に「アイドル状態」の次へと進行している可能性があり、イベントに反映されます。この新しい接続は、コールの通常の宛先側の接続として進行します。接続の一般的なシナリオについて Call.connect() メソッドで説明します。

事前条件

1. tc をこのコールの転送コントローラにする
2. (this.getProvider()).getState() == Provider.IN_SERVICE
3. this.getState() == Call.ACTIVE
4. tc.getCallControlState() == CallControlTerminalConnection.TALKING

事後条件

1. newconnection を作成され、返される接続にする
2. Let connection == tc.getConnection()
3. (this.getProvider()).getState() == Provider.IN_SERVICE
4. this.getState() == Call.ACTIVE
5. tc.getCallControlState() == CallControlTerminalConnection.DROPPED

6. connection.getTerminalConnections().length == 1 の場合、connection.getCallControlState() == CallControlConnection.DISCONNECTED
7. null でない場合、newconnection は this.getConnections() の要素
8. null でない場合、newconnection.getCallControlState() で、少なくとも CallControlConnection.IDLE
9. ConnCreatedEv が newconnection に対して配信される
10. CallCtlTermConnDroppedEv/TermConnDroppedEv が tc に対して配信される
11. 他に TerminalConnections が存在しない場合、CallCtlConnDisconnectedEv/ConnDisconnectedEv が接続に対して配信される

パラメータ

- address：コールが転送される着信先アドレス文字列（dialedDigits）。
- facCode：Force Authorization Code。
- cmcCode：Client Matter Code。

戻り値

宛先に関連付けられた新しい接続、または null。

例外

javax.telephony.InvalidArgumentException：転送を制御するように指定された TerminalConnection が有効でないか、このコールの一部ではありません。

javax.telephony.InvalidStateException：プロバイダーが「イン サービス」でないか、コールが「アクティブ」でないか、転送コントローラが「通話中」でないかのいずれかです。

javax.telephony.InvalidPartyException：着信先アドレスが無効であるか、不完全です。

javax.telephony.MethodNotSupportedException：この実装では、このメソッドをサポートしません。

javax.telephony.PrivilegeViolationException：アプリケーションに、このメソッドを起動する適切な権限がありません。

javax.telephony.ResourceUnavailableException：このメソッドの正常な起動に必要な内部リソースがありません。

関連項目

ConnCreatedEv, ConnDisconnectedEv, TermConnDroppedEv, CallCtlConnDisconnectedEv, CallCtlTermConnDroppedEv

getCurrentCalledAddressPIboolean getCurrentCalledAddressPI()CurrentCalledAddress に関連付けられた Presentation Indicator（PI）を返します。true が返された場合、アプリケーションはこのアドレス名をエンド ユーザに表示できます。false が返された場合、アプリケーションはこのアドレス名をエンド ユーザに表示しません。

getCurrentCalledDisplayNamePIboolean getCurrentCalledDisplayNamePI()
getCurredCalledDisplayNamePI に関連付けられた Presentation Indicator（PI）を返します。true が返された場合、アプリケーションはこの DisplayName をエンド ユーザに表示できます。false が返された場合、アプリケーションはこの DisplayName をエンド ユーザに表示しません。

getCurrentCallingAddressPIboolean getCurrentCallingAddressPI()getCurrentCallingAddressPI に関連付けられた Presentation Indicator（PI）を返します。true が返された場合、アプリケーションはこのアドレス名をエンド ユーザに表示できます。false が返された場合、アプリケーションはこのアドレス名をエンド ユーザに表示しません。

getCurrentCallingDisplayNamePIboolean getCurrentCallingDisplayNamePI()
getCurrentCalledDisplayNamePI に関連付けられた Presentation Indicator（PI）を返します。true が返された場合、アプリケーションはこの DisplayName をエンド ユーザに表示できます。false が返された場合、アプリケーションはこの DisplayName をエンド ユーザに表示しません。

getLastRedirectingAddressPIboolean getLastRedirectingAddressPI()getLastRedirectingAddressPI に関連付けられた Presentation Indicator（PI）を返します。true が返された場合、アプリケーションはこのアドレス名をエンド ユーザに表示できます。false が返された場合、アプリケーションはこのアドレス名をエンド ユーザに表示しません。

getCalledAddressPIboolean getCalledAddressPI()getCalledAddressPI に関連付けられた Presentation Indicator（PI）を返します。true が返された場合、アプリケーションはこのアドレス名をエンド ユーザに表示できます。false が返された場合、アプリケーションはこのアドレス名をエンド ユーザに表示しません。

getCallingAddressPIboolean getCallingAddressPI()getCallingAddressPI に関連付けられた Presentation Indicator（PI）を返します。true が返された場合、アプリケーションはこのアドレス名をエンド ユーザに表示できます。false が返された場合、アプリケーションはこのアドレス名をエンド ユーザに表示しません。

getCurrentCalledPartyUnicodeDisplayNamejava.lang.String
getCurrentCalledPartyUnicodeDisplayName()現在のコールの着側の Unicode 表示名を返します。表示名が不明な場合は null を返します。

getCurrentCalledPartyUnicodeDisplayNamelocaleint
getCurrentCalledPartyUnicodeDisplayNamelocale()現在のコールの着側の Unicode 表示名のロケールを返します。CiscoLocale インターフェイスは、サポートされているロケールを列挙します。

getCurrentCallingPartyUnicodeDisplayNamejava.lang.String
getCurrentCallingPartyUnicodeDisplayName()コールの発側の Unicode 表示名を返します。表示名が不明な場合は null を返します。

getCurrentCallingPartyUnicodeDisplayNamelocaleint
getCurrentCallingPartyUnicodeDisplayNamelocale()現在のコールの着側の Unicode 表示名のロケールを返します。

getCurrentCallingPartyInfoCiscoPartyInfo getCurrentCallingPartyInfo()現在のコールの発側の PartyInfo を返します。

getCurrentCalledPartyInfoCiscoPartyInfo getCurrentCalledPartyInfo()コールの現在の着側の PartyInfo を返します。

getLastRedirectingPartyInfoCiscoPartyInfo getLastRedirectingPartyInfo()推奨されません。getLastRedirectedPartyInfo() を使用してください。

コールを最後にリダイレクトした通話者の PartyInfo を返します。

getLastRedirectedPartyInfoCiscoPartyInfo getLastRedirectedPartyInfo()コールを最後にリダイレクトした通話者の PartyInfo を返します。

getCalledPartyInfoCiscoPartyInfo getCalledPartyInfo()コールの着側の PartyInfo を返します。

javax.telephony.Connection[]startMonitor(javax.telephony.Terminal MonitorInitiatorterminal,
javax.telephony.Address MonitorInitiatoraddress,
javax.telephony.TerminalConnection termConnofMonitorTarget, int monitorType,
int PlayToneDirection)

throws

javax.telephony.ResourceUnavailableException, javax.telephony.PrivilegeViolationException,
javax.telephony.InvalidPartyException, javax.telephony.InvalidArgumentException,
javax.telephony.InvalidStateException, javax.telephony.MethodNotSupportedException

アプリケーションがモニタリング ターゲット（エージェント）のアドレスを監視している場合、このアプリケーションはモニタリング ターゲット（エージェント）の端末接続を使用して、モニタリング要求を開始できます。このインターフェイスは発側のエンドポイントからコールを発信し、モニタリング ターゲットのコールをモニタリングします。

事前条件

1. (this.getProvider()).getState() == Provider.IN_SERVICE
2. this.getState() == Call.IDLE
3. ((CiscoProviderCapabilities)(this.getTerminal().getProvider().getProviderCapabilities()).canMonitor( ) == TRUE
4. TerminalConnection.getProvider() == this.getProvider()

パラメータ

- MonitorInitiatorterminal：発呼側の端末。
- MonitorInitiatoraddress：発呼側の端末アドレス。
- termConnofMonitorTarget：ターゲットの TerminalConnection。
- monitorType：モニタリングのタイプ。CiscoCall.SILENT_MONITOR を使用します。
- PlayToneDirection：トーンをターゲット、発信側、または両方で再生するかを示します。使用可能な値は、CiscoCall.PLAYTONE_NOLOCAL_OR_REMOTE、CiscoCall.PLAYTONE_LOCALONLY、CiscoCall.PLAYTONE_REMOTEONLY、または CiscoCall.PLAYTONE_BOTHLOCALANDREMOTE です。

例外

javax.telephony.ResourceUnavailableException

javax.telephony.PrivilegeViolationException

javax.telephony.InvalidPartyException

javax.telephony.InvalidArgumentException

javax.telephony.InvalidStateException

javax.telephony.MethodNotSupportedException

関連資料

詳細については、CallControlCall を参照してください。

CiscoCallChangedEv

CiscoCallChangedEv イベントは、すべてのサポートされる機能でコールの Global Call ID（GCID）が変更されるたびに、コール オブザーバに配信されます。CiscoCallChangedEv は、パス交換（(QSIG_PR）およびその他の機能（転送、会議、割り込み、パーク解除など）のためにコールの GCID が変更された場合に配信されます。共用回線の場合、複数の CiscoCallChangedEv イベントが配信されます。

このイベントは、2 つ以上のコールを 1 つのコールにマージしたときに配信されます。転送、会議、パーク解除、割り込み、および C 割り込みによってこのイベントが開始されます。アプリケーションは CiscoCallEv.getCiscoFeatureReason() を起動して、このイベントが発生した原因である機能コードを検出できます。

このイベントは CallControlCallObserver インターフェイスによって報告されます。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

javax.telephony.events.CallEv, CiscoCallEv, CiscoEv, javax.telephony.events.Ev

宣言

public interface CiscoCallChangedEv extends CiscoCallEv

フィールド

表 6-54　CiscoCallChangedEv のフィールド

インターフェイス	フィールド
static int	ID

継承したフィールド

インターフェイス com.cisco.jtapi.extensions.CiscoCallEv から
CAUSE_ACCESSINFORMATIONDISCARDED, CAUSE_BARGE,
CAUSE_BCBPRESENTLYAVAIL, CAUSE_BCNAUTHORIZED, CAUSE_BEARERCAPNIMPL,
CAUSE_CALLBEINGDELIVERED, CAUSE_CALLIDINUSE,
CAUSE_CALLMANAGER_FAILURE, CAUSE_CALLREJECTED, CAUSE_CALLSPLIT,
CAUSE_CHANTYPENIMPL, CAUSE_CHANUNACCEPTABLE,
CAUSE_CTICCMSIP400BADREQUEST, CAUSE_CTICCMSIP401UNAUTHORIZED,
CAUSE_CTICCMSIP402PAYMENTREQUIRED, CAUSE_CTICCMSIP403FORBIDDEN,
CAUSE_CTICCMSIP404NOTFOUND, CAUSE_CTICCMSIP405METHODNOTALLOWED,
CAUSE_CTICCMSIP406NOTACCEPTABLE,
CAUSE_CTICCMSIP407PROXYAUTHENTICATIONREQUIRED,
CAUSE_CTICCMSIP408REQUESTTIMEOUT, CAUSE_CTICCMSIP410GONE,
CAUSE_CTICCMSIP411LENGTHREQUIRED,
CAUSE_CTICCMSIP413REQUESTENTITYTOOLONG,
CAUSE_CTICCMSIP414REQUESTURITOOLONG,
CAUSE_CTICCMSIP415UNSUPPORTEDMEDIATYPE,

CAUSE_CTICCMSIP416UNSUPPORTEDURISCHEME,
CAUSE_CTICCMSIP420BADEXTENSION, CAUSE_CTICCMSIP421EXTENSTIONREQUIRED,
CAUSE_CTICCMSIP423INTERVALTOOBRIEF,
CAUSE_CTICCMSIP480TEMPORARILYUNAVAILABLE,
CAUSE_CTICCMSIP481CALLLEGDOESNOTEXIST, CAUSE_CTICCMSIP482LOOPDETECTED,
CAUSE_CTICCMSIP483TOOMANYHOOPS, CAUSE_CTICCMSIP484ADDRESSINCOMPLETE,
CAUSE_CTICCMSIP485AMBIGUOUS, CAUSE_CTICCMSIP486BUSYHERE,
CAUSE_CTICCMSIP487REQUESTTERMINATED,
CAUSE_CTICCMSIP488NOTACCEPTABLEHERE, CAUSE_CTICCMSIP491REQUESTPENDING,
CAUSE_CTICCMSIP493UNDECIPHERABLE,
CAUSE_CTICCMSIP500SERVERINTERNALERROR,
CAUSE_CTICCMSIP501NOTIMPLEMENTED, CAUSE_CTICCMSIP502BADGATEWAY,
CAUSE_CTICCMSIP503SERVICEUNAVAILABLE, CAUSE_CTICCMSIP504SERVERTIMEOUT,
CAUSE_CTICCMSIP505SIPVERSIONNOTSUPPORTED,
CAUSE_CTICCMSIP513MESSAGETOOLARGE, CAUSE_CTICCMSIP600BUSYEVERYWHERE,
CAUSE_CTICCMSIP603DECLINE, CAUSE_CTICCMSIP604DOESNOTEXISTANYWHERE,
CAUSE_CTICCMSIP606NOTACCEPTABLE, CAUSE_CTICONFERENCEFULL,
CAUSE_CTIDEVICENOTPREEMPTABLE, CAUSE_CTIDROPCONFEREE,
CAUSE_CTIMANAGER_FAILURE, CAUSE_CTIPRECEDENCECALLBLOCKED,
CAUSE_CTIPRECEDENCELEVELEXCEEDED,
CAUSE_CTIPRECEDENCEOUTOFBANDWIDTH, CAUSE_CTIPREEMPTFORREUSE,
CAUSE_CTIPREEMPTNOREUSE, CAUSE_DESTINATIONOUTOFORDER,
CAUSE_DESTNUMMISSANDDCNOTSUB, CAUSE_DPARK, CAUSE_DPARK_REMINDER,
CAUSE_DPARK_UNPARK, CAUSE_EXCHANGEROUTINGERROR, CAUSE_FAC_CMC,
CAUSE_FACILITYREJECTED, CAUSE_IDENTIFIEDCHANDOESNOTEXIST, CAUSE_IENIMPL,
CAUSE_INBOUNDBLINDTRANSFER, CAUSE_INBOUNDCONFERENCE,
CAUSE_INBOUNDTRANSFER, CAUSE_INCOMINGCALLBARRED,
CAUSE_INCOMPATABLEDDESTINATION, CAUSE_INTERWORKINGUNSPECIFIED,
CAUSE_INVALIDCALLREFVALUE, CAUSE_INVALIDIECONTENTS,
CAUSE_INVALIDMESSAGEUNSPECIFIED, CAUSE_INVALIDNUMBERFORMAT,
CAUSE_INVALIDTRANSITNETSEL, CAUSE_MANDATORYIEMISSING,
CAUSE_MSGNCOMPATABLEWCS, CAUSE_MSGTYPENCOMPATWCS,
CAUSE_MSGTYPENIMPL, CAUSE_NETOUTOFORDER, CAUSE_NOANSWERFROMUSER,
CAUSE_NOCALLSUSPENDED, CAUSE_NOCIRCAVAIL, CAUSE_NOERROR,
CAUSE_NONSELECTEDUSERCLEARING, CAUSE_NORMALCALLCLEARING,
CAUSE_NORMALUNSPECIFIED, CAUSE_NOROUTETODDESTINATION,
CAUSE_NOROUTETOTRANSITNET, CAUSE_NOUSERRESPONDING,
CAUSE_NUMBERCHANGED, CAUSE_ONLYRDIVEARERCAPAVAIL,
CAUSE_OUTBOUNDCONFERENCE, CAUSE_OUTBOUNDTRANSFER,
CAUSE_OUTOFBANDWIDTH, CAUSE_PROTOCOLERRORUNSPECIFIED, CAUSE_QSIG_PR,
CAUSE_QUALOFSERVNAVAIL, CAUSE_QUIET_CLEAR,
CAUSE_RECOVERYONTIMEREXPIRY, CAUSE_REDIRECTED,
CAUSE_REQCALLIDHASBEENCLEARED, CAUSE_REQCIRCNAVIL,
CAUSE_REQFACILITYNIMPL, CAUSE_REQFACILITYNOTSUBSCRIBED,
CAUSE_RESOURCESNAVAIL, CAUSE_RESPONSETOSTATUSENQUIRY,
CAUSE_SERVNOTAVAILUNSPECIFIED, CAUSE_SERVOPERATIONVIOLATED,
CAUSE_SERVOROPTNAVAILORIMPL, CAUSE_SUBSCRIBERABSENT,
CAUSE_SUSPCALLBUTNOTTHISONE, CAUSE_SWITCHINGEQUIPMENTCONGESTION,
CAUSE_TEMPORARYFAILURE, CAUSE_UNALLOCATEDNUMBER, CAUSE_USERBUSY

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

表 6-55　CiscoCallChangedEv のメソッド

インターフェイス	メソッド	説明
CiscoConnection	getConnection()	変更が発生したアドレスに CiscoConnection を返します。
CiscoCall	getOriginalCall()	INVALID 状態にするコールを返します。
CiscoCall	getSurvivingCall()	callID の変更後も有効のままになっているコールを返します。
javax.telephony.TerminalConnection	getTerminalConnection()	変更が発生した TerminalConnection を返します。この値は、アドレスに対して TerminalConnection が作成される前にコール ID が変更された場合、null である可能性があります。

継承したメソッド

インターフェイス com.cisco.jtapi.extensions.CiscoCallEv から
getCiscoCause, getCiscoFeatureReason

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.CallEv から
getCall

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

詳細については、「定数フィールド値」（P.F-1）を参照してください。

CiscoCallConsultCancelledEv

このイベントはアプリケーションに、操作のキャンセルが発生したことを通知します。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	スワップ / キャンセルの新しいイベント：転送 / 会議動作の変更機能。

スーパーインターフェイス

なし

宣言

public interface CiscoCallConsultCancelledEv

フィールド

なし

継承したフィールド

なし

メソッド

表 6-56　CiscoCallConsultCancelledEv のメソッド

インターフェイス	メソッド	説明
CiscoCall	getConsultCall()	コンサルト オペレーションがキャンセルされるコンサルト コールを返します。コンサルト コールが存在しない場合、NULL を返します。 このコール イベントの getCall() API は親コールを返します。

継承したメソッド

なし

関連資料

なし。

CiscoCallCtlConnOfferedEv

CiscoCallCtlConnOfferedEv インターフェイスは CallCtlConnOfferedEv インターフェイスを拡張して、アプリケーションが発信側端末の IP アドレスを取得できるようにします。すべての発信側デバイスの IP アドレス情報が参照可能とは限りません。戻り値が 0（または null）の場合、情報が取得可能でないことを示します。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

javax.telephony.callcontrol.events.CallCtlCallEv, javax.telephony.callcontrol.events.CallCtlConnEv, javax.telephony.callcontrol.events.CallCtlConnOfferedEv, javax.telephony.callcontrol.events.CallCtlEv, javax.telephony.events.CallEv, javax.telephony.events.ConnEv, javax.telephony.events.Ev

宣言

public interface CiscoCallCtlConnOfferedEv extends javax.telephony.callcontrol.events.CallCtlConnOfferedEv

フィールド

なし

継承したフィールド

インターフェイス javax.telephony.callcontrol.events.CallCtlConnOfferedEv から
なし

インターフェイス javax.telephony.callcontrol.events.CallCtlEv から
CAUSE_ALTERNATE, CAUSE_BUSY, CAUSE_CALL_BACK, CAUSE_CALL_NOT_ANSWERED, CAUSE_CALL_PICKUP, CAUSE_CONFERENCE, CAUSE_DO_NOT_DISTURB, CAUSE_PARK, CAUSE_REDIRECTED, CAUSE_REORDER_TONE, CAUSE_TRANSFER, CAUSE_TRUNKS_BUSY, CAUSE_UNHOLD

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

表 6-57　CiscoCallCtlConnOfferedEv のメソッド

インターフェイス	メソッド	説明
java.net.InetAddress	getCallingPartyIpAddr()	発信側の IP アドレスか、IP アドレスが取得できない場合は 0（または null）を返します。

継承したメソッド

インターフェイス javax.telephony.callcontrol.events.CallCtlCallEv から
getCalledAddress, getCallingAddress, getCallingTerminal, getLastRedirectedAddress

インターフェイス javax.telephony.callcontrol.events.CallCtlEv から
getCallControlCause

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.CallEv から
getCall

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.ConnEv から
getConnection

インターフェイス javax.telephony.events.CallEv から
getCall

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

なし

CiscoCallCtlTermConnHeldReversionEv

CiscoCallCtlTermConnHeldReversionEv イベントは、Cisco Unified Communications Manager から TerminalConnection で保留復帰通知を受信したことを示します。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

javax.telephony.callcontrol.events.CallCtlCallEv, javax.telephony.callcontrol.events.CallCtlEv, javax.telephony.callcontrol.events.CallCtlTermConnEv, javax.telephony.events.CallEv, javax.telephony.events.Ev, javax.telephony.events.TermConnEv

宣言

public interface CiscoCallCtlTermConnHeldReversionEv extends javax.telephony.callcontrol.events.CallCtlTermConnEv

フィールド

表 6-58　CiscoCallCtlTermConnHeldReversionEv のフィールド

インターフェイス	フィールド
static int	ID

継承したフィールド

インターフェイス javax.telephony.callcontrol.events.CallCtlEv から
CAUSE_ALTERNATE, CAUSE_BUSY, CAUSE_CALL_BACK, CAUSE_CALL_NOT_ANSWERED, CAUSE_CALL_PICKUP, CAUSE_CONFERENCE, CAUSE_DO_NOT_DISTURB, CAUSE_PARK, CAUSE_REDIRECTED, CAUSE_REORDER_TONE, CAUSE_TRANSFER, CAUSE_TRUNKS_BUSY, CAUSE_UNHOLD

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE,

CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

なし

継承したメソッド

インターフェイス javax.telephony.callcontrol.events.CallCtlCallEv から
getCalledAddress, getCallingAddress, getCallingTerminal, getLastRedirectedAddress

インターフェイス javax.telephony.callcontrol.events.CallCtlEv から
getCallControlCause

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.CallEv から
getCall

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.TermConnEv から
getTerminalConnection

インターフェイス javax.telephony.events.CallEv から
getCall

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

詳細については、「定数フィールド値」（P.F-1）を参照してください。

CiscoCallEv

JTAPI のコアである javax.telephony.events.CallEv インターフェイスを拡張する CiscoCallEv インターフェイスは、Cisco によって拡張されたすべての JTAPI Call イベントの基本インターフェイスになります。このパッケージのコール関連イベントはすべて、直接的または間接的にこのインターフェイスを拡張します。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

javax.telephony.events.CallEv, CiscoEv, javax.telephony.events.Ev

サブインターフェイス

CiscoCallChangedEv, CiscoCallSecurityStatusChangedEv, CiscoConferenceChainAddedEv, CiscoConferenceChainRemovedEv, CiscoConferenceEndEv, CiscoConferenceStartEv, CiscoConsultCallActiveEv, CiscoToneChangedEv, CiscoTransferEndEv, CiscoTransferStartEv

宣言

public interface CiscoCallEv extends CiscoEv, javax.telephony.events.CallEv

フィールド

表 6-59　CiscoCallEv のフィールド

インターフェイス	フィールド	説明
Static int	CAUSE_ACCESSINFORMATION DISCARDED	これは、リモート ユーザが要求したときに、ネットワークがアクセス情報を配信できないことを示します。
Static int	CAUSE_BARGE	コールが割り込みコールであることを示します。

表 6-59　CiscoCallEv のフィールド （続き）

インターフェイス	フィールド	説明
staticint	CAUSE_BCBPRESENTLYAVAIL	この原因は、ユーザがベアラ機能を要求したことを示します。ベアラ機能は、この原因を生成した装置によって実装されますが、この時点では利用できません。
static int	CAUSE_BCNAUTHORIZED	この原因は、ユーザがベアラ機能を要求したことを示します。ベアラ機能は、この原因を生成した装置によって実装されますが、このユーザは使用を許可されていません。
static int	CAUSE_BEARERCAPNIMPL	この原因は、この原因を送信した装置が、要求されたベアラ機能をサポートしていないことを示します。
static int	CAUSE_CALLBEINGDELIVERED	この原因は、ユーザに対して着信コールがあり、その着信コールが、同様のコールのためにそのユーザに対してすでに確立されているチャネルに接続されることを示します。
static int	CAUSE_CALLIDINUSE	この原因は、ネットワークが、コールが再開される可能性があるインターフェイスのドメイン内にある一時停止されたコールですでに使用されているコールの ID（null コール ID を含む）を含むコールの一時停止要求を受け取ったことを意味します。
static int	CAUSE_CALLMANAGER_FAILURE	この原因は、コール マネージャの障害によって発生した障害を示します。
static int	CAUSE_CALLREJECTED	この原因は、この原因を送信した装置が、このコールの受け入れを拒否していることを示します。
static int	CAUSE_CALLSPLIT	この原因は、コール スプリットを示します。つまり、会議または転送です。
static int	CAUSE_CHANTYPENIMPL	この原因は、この原因を送信した装置が、要求されたチャネル タイプをサポートしていないことを示します。
static int	CAUSE_CHANUNACCEPTABLE	この原因は、最後に指定されたチャネルでは、このコールで使用するための送信エンティティが受け入れられないことを示します。
static int	CAUSE_CTICCMSIP400BADREQUEST	この原因は、不正な要求のためにコールが拒否されることを示します。
static int	CAUSE_CTICCMSIP401UNAUTHORIZED	この原因は、要求は有効であるものの、権限を持っていないことを示します。
static int	CAUSE_CTICCMSIP402PAYMENT REQUIRED	これは、使用するためには支払いが必要であることを示します。
static int	CAUSE_CTICCMSIP403FORBIDDEN	この原因は、サーバが要求を認識したものの、対応を拒否していることを示します。
static int	CAUSE_CTICCMSIP404NOTFOUND	この原因は、サーバで要求の URI を検出できなかったことを示します。

表 6-59　CiscoCallEv のフィールド （続き）

インターフェイス	フィールド	説明
static int	CAUSE_CTICCMSIP405METHODNOTALLOWED	この原因は、Request-Line で指定されたメソッドが認識されたものの、Request-URI で指定されたアドレスで許可されていないことを示します。
static int	CAUSE_CTICCMSIP406NOTACCEPTABLE	この原因は、要求の要件が満たされていないために、要求を処理できないことを意味します。
static int	CAUSE_CTICCMSIP407PROXYAUTHENTICATIONREQUIRED	この原因は、要求を行うための権限がなく、要求のためにはプロキシ認証が必要であることを示します。
static int	CAUSE_CTICCMSIP408REQUESTTIMEOUT	この原因は、要求に対するタイム アウト エラーを意味します。
static int	CAUSE_CTICCMSIP410GONE	この原因は、要求されたリソースがサーバで利用できなくなっており、転送アドレスが不明であることを示します。
static int	CAUSE_CTICCMSIP411LENGTHREQUIRED	この原因は、インターワーキング メッセージの長さが必要であることを示します。
static int	CAUSE_CTICCMSIP413REQUESTENTITYTOOLONG	この原因は、要求のエンティティ自体が、サーバが処理を想定しているサイズ、または処理できるサイズよりも大きいために、サーバが要求の処理を拒否していることを示します。
static int	CAUSE_CTICCMSIP414REQUESTURITOOLONG	この原因は、Request-URI がサーバが解釈を想定している長さ、または解釈できる長さよりも長いために、サーバが要求の処理を拒否していることを示します。
static int	CAUSE_CTICCMSIP415UNSUPPORTEDMEDIATYPE	この原因は、要求のメッセージの本文で、要求されたメソッドのために、サーバでサポートされないメディア タイプが記述されているために、サーバが要求の処理を拒否していることを意味します。
static int	CAUSE_CTICCMSIP416UNSUPPORTEDURISCHEME	この原因は、Request-URI の URI のスキームがサーバで認識されないために、サーバが要求を処理できないことを示します。
static int	CAUSE_CTICCMSIP420BADEXTENSION	この原因は、Proxy-Require または Require ヘッダー フィールドで指定されたプロトコル拡張がサーバで認識されないことを示します。
static int	CAUSE_CTICCMSIP421EXTENSTIONREQUIRED	この原因は、UAS が要求を処理するために特別な拡張が必要であるものの、この拡張が要求の Supported ヘッダー フィールドの一覧に表示されていないことを示します。
static int	CAUSE_CTICCMSIP423INTERVALTOOBRIEF	この原因は、要求によってリフレッシュされるリソースの期限切れ時間が短すぎるために、サーバが要求を拒否していることを示します。

表 6-59　CiscoCallEv のフィールド （続き）

インターフェイス	フィールド	説明
static int	CAUSE_CTICCMSIP480TEMPORARILY UNAVAILABLE	この原因は、着信側のエンド システムが正常に通信できるものの、着信側が現在、利用不能である（たとえば、ログインしていない、ログインしているが、着信側との通信が不可能な状態にある、または「Do Not Disturb（サイレント)」機能が有効になっている）ことを示します。
static int	CAUSE_CTICCMSIP481CALLLEGDOES NOTEXIST	この原因は、UAS が既存のダイアログまたはトランザクションと一致していない要求を受け取ったことを意味します。
static int	CAUSE_CTICCMSIP482LOOPDETECTED	この原因は、サーバがループを検出したことを示します。
static int	CAUSE_CTICCMSIP483TOOMANYHOOPS	この原因は、サーバが、Max-Forwards ヘッダー フィールドの値がゼロ（または実際のホップよりも小さい値）である要求を受け取ったことを示します。
static int	CAUSE_CTICCMSIP484ADDRESSIN COMPLETE	この原因は、サーバが Request-URI が不完全な要求を受け取ったことを示します。
static int	CAUSE_CTICCMSIP485AMBIGUOUS	この原因は、Request-URI があいまいであることを示します。
static int	CAUSE_CTICCMSIP486BUSYHERE	この原因は、着信側のエンド システムが正常に通信できるものの、着信側が現在、このエンド システムで追加のコールを受け入れることを拒否しているか、受け入れることができないことを示します。
static int	CAUSE_CTICCMSIP487REQUEST TERMINATED	この原因は、この要求が BYE 要求または CANCEL 要求によって終了したことを示します。
static int	CAUSE_CTICCMSIP488NOTACCEPTABLE HERE	この原因は、606（受け入れられない）と同じ意味であるものの、Request-URI によって指定された特定のリソースだけに適用され、要求が成功する場合もあることを示します。
static int	CAUSE_CTICCMSIP491REQUEST PENDING	この原因は、同じダイアログ内で保留中の要求がある UAS によって要求が受け入れられたことを示します。
static int	CAUSE_CTICCMSIP493 UNDECIPHERABLE	この原因は、受信者が適切な復号化キーを保持していないか、提供されていない、暗号化された MIME 本文を含む UAS によって要求が受け入れられたことを示します。
static int	CAUSE_CTICCMSIP500SERVERINTERNAL ERROR	この原因は、サーバで要求の処理を妨げる予期しない条件が発生したことを示します。
static int	CAUSE_CTICCMSIP501NOT IMPLEMENTED	この原因は、サーバが要求を処理するために必要な機能をサポートしていないことを示します。

表 6-59　CiscoCallEv のフィールド （続き）

インターフェイス	フィールド	説明
static int	CAUSE_CTICCMSIP502BADGATEWAY	この原因は、ゲートウェイまたはプロキシの役割を果たしているサーバが、要求の処理を試行するためにアクセスしたダウンストリーム サーバから無効な応答を受信したことを示します。
static int	CAUSE_CTICCMSIP503SERVICEUNAVAILABLE	この原因は、一時的なサーバのオーバーロードまたはメンテナンスのために、サーバが一時的に要求を処理できないことを示します。
static int	CAUSE_CTICCMSIP504SERVERTIMEOUT	この原因は、サーバが要求の処理を試行するためにアクセスした外部サーバから応答をタイムリーに受信できないことを示します。
static int	CAUSE_CTICCMSIP505SIPVERSIONNOTSUPPORTED	この原因は、サーバが、要求で使用された SIP プロトコル バージョンをサポートしていないか、サポートを拒否していることを示します。
static int	CAUSE_CTICCMSIP513MESSAGETOOLARGE	この原因は、メッセージの長さがサーバの機能を超えたために、サーバが要求を処理できないことを示します。
static int	CAUSE_CTICCMSIP600BUSYEVERYWHERE	この原因は、着信側のエンド システムが正常に通信しているものの、着信側がビジーであり、この時点でコールに応答しようとしていないことを示します。
static int	CAUSE_CTICCMSIP603DECLINE	この原因は、着信側のマシンが正常に通信しているものの、ユーザが明示的に参加しようとしていないか、参加できないことを示します。
static int	CAUSE_CTICCMSIP604DOESNOTEXISTANYWHERE	この原因は、ユーザが Request-URI でどこにも存在していないと示した認証情報をサーバが持っていることを示します。
static int	CAUSE_CTICCMSIP606NOTACCEPTABLE	この原因は、ユーザのエージェントが正常に通信しているものの、セッションに関する要求されたメディア、帯域幅、またはアドレス指定形式などが受け入れられなかったことを示します。
static int	CAUSE_CTICONFERENCEFULL	この原因は、電話会議がいっぱいで、参加者を追加できないことを示します。
static int	CAUSE_CTIDEVICENOTPREEMPTABLE	この原因は、デバイスのプリエンプション処理ができないことを示します。
static int	CAUSE_CTIDROPCONFEREE	この原因は、参加者が会議からドロップしたことによる接続解除を示します。
static int	CAUSE_CTIMANAGER_FAILURE	この原因は、CTI マネージャの障害によって発生した障害を意味します。

表 6-59　CiscoCallEv のフィールド （続き）

インターフェイス	フィールド	説明
static int	CAUSE_CTIPRECEDENCECALLBLOCKED	この原因は、予測可能な回路がないか、着信側が回避可能レベルが同じか、それより高いコールによってビジーであることを示します。
static int	CAUSE_CTIPRECEDENCELEVELEXCEEDED	この原因は、コールの優先順位レベルが認証レベルを超えていることを示します。
static int	CAUSE_CTIPRECEDENCEOUTOFBANDWIDTH	この原因は、優先コールが帯域幅不足で先に進めることができないことを示します。
static int	CAUSE_CTIPREEMPTFORREUSE	この原因は、コールが優先され、回路が優先交換で再利用されるために予約されていることを示します。
static int	CAUSE_CTIPREEMPTNOREUSE	この原因は、コールが優先されることを示します。
static int	CAUSE_DESTINATIONOUTOFORDER	この原因は、ユーザが指定した宛先に、その宛先へのインターフェイスが正常に機能していないためにアクセスできないことを示します。
static int	CAUSE_DESTNUMMISSANDDCNOTSUB	この原因は、指定された CUG が存在しないことを示します。
static int	CAUSE_DPARK	コールがダイレクト パークされたコールであることを示します。
static int	CAUSE_DPARK_REMINDER	ダイレクト パークされたコールのリマインダ コールであることを示します。
static int	CAUSE_DPARK_UNPARK	ダイレクト パークされたコールが、現在パーク解除されていることを示します。
static int	CAUSE_EXCHANGEROUTINGERROR	この原因は、交換機が、指定された宛先にコールをルーティングできなかったことを示します。
static int	CAUSE_FAC_CMC	Forced Authorization Code（FAC）または Client Matter Code（CMC）がコールのルーティングに必要であることを示します。
static int	CAUSE_FACILITYREJECTED	この原因は、ユーザが要求した補足サービスがネットワークによって提供されない場合に返されます。
static int	CAUSE_IDENTIFIEDCHANDOESNOTEXIST	この原因は、この原因を送信した装置が、コール用のインターフェイス上でアクティブになっていないチャネルを使用するという要求を受け取ったことを示します。
static int	CAUSE_IENIMPL	この原因は、この原因を送信した装置が、認識されない情報要素/パラメータが含まれるメッセージを受信したことを示します。これは、情報要素/パラメータが定義されていないか、または定義されているものの、この原因を送信した装置によって実装されないためです。

表 6-59　CiscoCallEv のフィールド （続き）

インターフェイス	フィールド	説明
static int	CAUSE_INBOUNDBLINDTRANSFER	コールが着信ブラインド転送コールであることを示します。
static int	CAUSE_INBOUNDCONFERENCE	コールが着信会議コールであることを示します。
static int	CAUSE_INBOUNDTRANSFER	コールが着信転送コールであることを示します。
static int	CAUSE_INCOMINGCALLBARRED	この原因は、この番号への着信コールが拒否されたことを示します。
static int	CAUSE_INCOMPATABLEDDESTINATION	この原因は、この原因を送信した装置が、低いレイヤとの互換性を持つコールを確立するという要求を受け取ったことを示します。
static int	CAUSE_INTERWORKINGUNSPECIFIED	この原因は、インターワーキング コールが終了したことを示します。
static int	CAUSE_INVALIDCALLREFVALUE	この原因は、この原因を送信した装置が、現在ユーザネットワーク インターフェイスで使用されていないコール参照を持つメッセージを受信したことを示します。
static int	CAUSE_INVALIDIECONTENTS	この原因は、この原因を送信した装置が、実装したものの、1 つ以上のフィールドがこの原因を送信した装置によって実装されなかった方法でコード化される情報要素を受け取ったことを示します。
static int	CAUSE_INVALIDMESSAGEUNSPECIFIED	この原因は、無効なメッセージ クラスの他の原因が適用されない場合にだけ、無効なメッセージ イベントを報告するために使用されます。
static int	CAUSE_INVALIDNUMBERFORMAT	この原因は、着信側の番号が有効な形式ではないか、不完全であるために、この着信側にアクセスできないことを示します。
static int	CAUSE_INVALIDTRANSITNETSEL	この原因は、不正な形式の中継ネットワーク ID を受信しました。
static int	CAUSE_MANDATORYIEMISSING	この原因は、この原因を送信した装置が、情報要素が欠落しているメッセージを受信し、欠落している部分があると、このメッセージを処理できないことを示します。
static int	CAUSE_MSGNCOMPATABLEWCS	この原因は、このコールの状態との互換性がないメッセージを受信したことを示します。
static int	CAUSE_MSGTYPENCOMPATWCS	この原因は、この原因を送信した装置が、手順でこのコールの状態で受信できるメッセージであることが示されていないメッセージを受信したか、または互換性のないコールの状態を示す STATUS メッセージを受信したことを示します。

表 6-59　CiscoCallEv のフィールド （続き）

インターフェイス	フィールド	説明
static int	CAUSE_MSGTYPENIMPL	この原因は、この原因を送信した装置が、定義されていないメッセージであるか、定義されているものの、この原因を送信した装置で実装されていないために認識されないメッセージ タイプのメッセージを受信したことを示します。
static int	CAUSE_NETOUTOFORDER	この原因は、ネットワークが正常に動作していないことと、この状態が比較的長い時間続くことが予測されるということを示します。
static int	CAUSE_NOANSWERFROMUSER	この原因は、着信側に警告が発せられているものの、所定の期間内の応答指示に応じない場合に使用されます。
static int	CAUSE_NOCALLSUSPENDED	この原因は、ネットワークが、現在インターフェイスのドメイン内で一時停止されている（復帰可能な）コールの存在することを示していないコール ID 情報要素を含むコール復帰要求を受け取ったことを示します。
static int	CAUSE_NOCIRCAVAIL	この原因は、現在、コールを処理するために利用できる適切な回路/チャネルがないことを示します。
static int	CAUSE_NOERROR	これは通常、エラーが発生しておらず、オペレーションが正常に完了する場合に指定されます。
static int	CAUSE_NONSELECTEDUSERCLEARING	この原因は、ユーザ対して着信コールがなかったことを示します。
static int	CAUSE_NORMALCALLCLEARING	この原因は、コールに関与しているいずれかのユーザがコールのクリアを要求したために、コールがクリアされたことを示します。
static int	CAUSE_NORMALUNSPECIFIED	この原因は、通常のクラスのその他の原因が適用されない場合にだけ、通常のイベントを報告するために使用されます。
static int	CAUSE_NOROUTETODDESTINATION	この原因は、コールがルーティングされたネットワークが、目的の宛先に対してサービスを提供していないために、この着信側にアクセスできないことを示します。
static int	CAUSE_NOROUTETOTRANSITNET	この原因は、この原因を送信した装置が、認識していない特定の中継ネットワークを介したコールのルーティング要求を受け取ったことを示します。
static int	CAUSE_NOUSERRESPONDING	この原因は、着信側が、割り当てられた期間内にアラートまたは接続が指示されたコール確立メッセージに応答しない場合に使用されます。
static int	CAUSE_NUMBERCHANGED	この原因は、発信側によって指定された着信側番号が、現在では割り当てられていない場合に、着信側に返されます。

表 6-59　CiscoCallEv のフィールド （続き）

インターフェイス	フィールド	説明
static int	CAUSE_ONLYRDIVEARERCAPAVAIL	この原因は、発信側が制限解除ベアラ サービスを要求したものの、この原因を送信した装置が、要求されたベアラ機能の制限バージョンだけをサポートしていることを示します。
static int	CAUSE_OUTBOUNDCONFERENCE	コールが発信会議コールであることを示します。
static int	CAUSE_OUTBOUNDTRANSFER	コールが発信転送コールであることを示します。
static int	CAUSE_OUTOFBANDWIDTH	この原因は、帯域幅不足のためにコールを処理できなかったことを示します。
static int	CAUSE_PROTOCOLERRORUNSPECIFIED	この原因は、プロトコル エラー クラスのその他の原因が適用されない場合にだけ、プロトコル エラーを報告するために使用されます。
static int	CAUSE_QSIG_PR	これは、コールの QSIG パス置換を示します。
static int	CAUSE_QUALOFSERVNAVAIL	この原因は、推奨事項 X.213 で定義されている、要求された Quality of Service（QoS）を報告するために使用されます。
static int	CAUSE_QUIET_CLEAR	これは、コール マネージャがダウンしたためにコールがクリアされたものの、エンドポイント間のメディアは接続されたままであることを示します。
static int	CAUSE_RECOVERYONTIMEREXPIRY	この原因は、エラー処理手順に関連付けられたタイマーの満了によって手順が開始されたことを示します。
static int	CAUSE_REDIRECTED	この原因は、コールが別の人にリダイレクトされることを示します。
static int	CAUSE_REQCALLIDHASBEENCLEARED	この原因は、ネットワークが、一時停止されていたコールが（ネットワークのタイムアウトまたはリモート ユーザによって）一時停止中にクリアされたことを示すコール ID 情報要素を含むコール復帰要求を受け取ったことを示します。
static int	CAUSE_REQCIRCNAVIL	この原因は、要求エンティティによって指定された回路またはチャネルが、インターフェイスの他の側で提供できない場合に返されます。
static int	CAUSE_REQFACILITYNIMPL	この原因は、この原因を送信した装置が、要求された内容をサポートしていないことを示します。
static int	CAUSE_REQFACILITYNOTSUBSCRIBED	この原因は、ユーザが補足サービスを要求したことを示します。補足サービスは、この原因を生成した装置によって実装されますが、このユーザは使用を許可されていません。

表 6-59　CiscoCallEv のフィールド （続き）

インターフェイス	フィールド	説明
static int	CAUSE_RESOURCESNAVAIL	この原因は、リソースを利用できないというイベントを報告するために使用されます。
static int	CAUSE_RESPONSETOSTATUSENQUIRY	この原因は、STATUS メッセージが生成された理由が STATUS INQUIRY の受信の前である場合、STATUS メッセージに含まれます。
static int	CAUSE_SERVNOTAVAILUNSPECIFIED	この原因は、利用できないクラスのサービスまたはオプションのその他の原因が適用されない場合にだけ、サービスまたはオプションを利用できないというイベントを報告するために使用されます。
static int	CAUSE_SERVOPERATIONVIOLATED	この原因は、発信側が発信 CUG コールの CUG のメンバであっても示します。
static int	CAUSE_SERVOROPTNAVAILORIMPL	この原因は、実装されないクラスのサービスまたはオプションのその他の原因が適用されない場合にだけ、サービスまたはオプションが実装されないというイベントを報告するために使用されます。
static int	CAUSE_SUBSCRIBERABSENT	この原因の値は、モバイル端末がログオフした場合に使用されます。
static int	CAUSE_SUSPCALLBUTNOTTHISONE	この原因は、現在、一時停止中のコールで使用されているコール ID とは異なるコール ID でコールの復帰が試行されたことを示します。
static int	CAUSE_SWITCHINGEQUIPMENT CONGESTION	この原因は、この原因を生成した装置の切り替えによってトラフィックが増大することを示します。
static int	CAUSE_TEMPORARYFAILURE	この原因は、ネットワークが正常に動作していないことと、この状態が長い時間続き、ユーザがすぐにでも別のコールを試行することが予測されるということを示します。
static int	CAUSE_UNALLOCATEDNUMBER	この原因は、発信側のユーザによって要求された宛先が、無効な番号であるためにアクセスできないことを示します。
static int	CAUSE_USERBUSY	この原因は、着信側のユーザがビジー状態であるために、別のコールを受け入れることができないことを示すために使用されます。

継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE,

CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.CallEv から
getCall

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

メソッド

表 6-60　CiscoCallEv のメソッド

インターフェイス	メソッド	説明
Int	getCiscoCause()	このイベントに関する Cisco Unified Communications Manager の原因を返します。監視対象のエンドポイントでイベントが発生した原因を識別しないと、適切に機能しないアプリケーションもあります。たとえば、コールに応答がなかったために接続が解除されたのか（CAUSE_NOANSWERFROMUSER）、あるいはコールが拒否されたために接続が解除されたのか（CAUSE_CALLREJECTED）を識別しなければならない場合があります。戻り値：このイベントに関する Cisco Unified Communications Manager の原因

表 6-60　CiscoCallEv のメソッド

インターフェイス	メソッド	説明
Int	getCiscoFeatureReason()	このイベントに関する Cisco Unified Communications Manager の機能原因を返します。イベントが発生した原因を識別しないと、適切に機能しないアプリケーションもあります。このインターフェイスは、現行の機能および新機能における JTAPI Call イベントに CiscoFeatureReason を提供します。転送などの既存の機能は、現状どおり以前の CiscoCallEv.getCiscoCause() インターフェイスから CiscoCause を受け取ります。このインターフェイスは転送の REASON_TRANSFER を提供します。注意：アプリケーションは、未知の原因を処理してデフォルト動作を提示できる必要があります。これは将来新しい原因が追加され、このインターフェイスの下位互換性がなくなる可能性があるためです。指定可能な値は CiscoFeatureReason インターフェイスで定義されています。戻り値：このイベントに関する Cisco Unified Communications Manager の機能原因

関連資料

詳細については、「定数フィールド値」（P.F-1）と CallEv を参照してください。

CiscoCallFeatureCancelledEv

このイベントはアプリケーションに、操作のキャンセルが発生したことを通知します。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(2)	変更を記録するために履歴表が作成されました。

宣言

public interface CiscoCallFeatureCancelledEv

メソッド

表 6-61　CiscoCallFeatureCancelledEv のメソッド

インターフェイス	メソッド	説明
CiscoCall	getConsultCall()	コンサルト オペレーションがキャンセルされるコンサルト コールを返します。コンサルト コールが存在しない場合、NULL を返します。

関連資料

「定数フィールド値」（P.F-1）を参照してください。

CiscoCallID

CiscoCallID オブジェクトは、各コールに関連付けられる一意のオブジェクトです。アプリケーションでは、オブジェクト自体または intValue() メソッドで返されたオブジェクトの整数表現を使用できます。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

CiscoObjectContainer

宣言

Public interface CiscoCallID extends CiscoObjectContainer

フィールド

なし

メソッド

表 6-62　CiscoCallID のメソッド

インターフェイス	メソッド	説明
Int	intValue()	このオブジェクトの整数表現を返します。戻り値：このオブジェクトの Int An 整数表現
CiscoCall	getCall()	この CiscoCallID に対応している CiscoCall を返します。
int	getCallManagerID()	この CiscoCallID に関連付けられたコールの Cisco Unified Communications Manager NodeID を返します。
int	getGlobalCallID()	この CiscoCallID に関連付けられたコールの GlobalCallID を返します。

継承したメソッド

インターフェイス com.cisco.jtapi.extensions.CiscoObjectContainer から
getObject, setObject

関連資料

なし

CiscoMediaCallSecurityIndicator

CiscoMediaCallSecurityIndicator では、コールのセキュリティ インジケータを取得できます。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

宣言

public interface CiscoMediaCallSecurityIndicator

フィールド

なし

メソッド

表 6-63　CiscoMediaCallSecurityIndicator のメソッド

インターフェイス	メソッド	説明
CiscoCallID	getCallID()	CiscoCallID を返します。
int	getCiscoMediaSecurityIndicator()	次の定数の 1 つのメディアのセキュリティ インジケータを返します。 CiscoMediaSecurityIndicator.MEDIA_ENCRYPT_USER_NOT_AUTHORIZED CiscoMediaSecurityIndicator.MEDIA_ENCRYPTED_KEYS_UNAVAILABLE CiscoMediaSecurityIndicator.MEDIA_NOT_ENCRYPTED
CiscoRTPHandle	getCiscoRTPHandle()	CiscoProvider.getCall を使用して CiscoRTPHandle object.Applications が取得できるコール参照を返します。コール オブザーバがない場合や、このイベントの配信時にコール オブザーバがなかった場合、CiscoProvider.getCall は null を返す可能性があります。

関連資料

CiscoRTPParams を参照してください。

CiscoCallSecurityStatusChangedEv

コール全体のセキュリティ ステータスが変更されると、アプリケーションは CiscoCallSecurityStatusChangedEv を受信します。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

javax.telephony.events.CallEv, CiscoCallEv, CiscoEv, javax.telephony.events.Ev

宣言

public interface CiscoCallSecurityStatusChangedEv extends CiscoCallEv

フィールド

表 6-64　CiscoCallSecurityStatusChangedEv のフィールド

インターフェイス	フィールド
Static int	ID

継承したフィールド

インターフェイス com.cisco.jtapi.extensions.CiscoCallEv から

CAUSE_ACCESSINFORMATIONDISCARDED, CAUSE_BARGE,
CAUSE_BCBPRESENTLYAVAIL, CAUSE_BCNAUTHORIZED, CAUSE_BEARERCAPNIMPL,
CAUSE_CALLBEINGDELIVERED, CAUSE_CALLIDINUSE,
CAUSE_CALLMANAGER_FAILURE, CAUSE_CALLREJECTED, CAUSE_CALLSPLIT,
CAUSE_CHANTYPENIMPL, CAUSE_CHANUNACCEPTABLE,
CAUSE_CTICCMSIP400BADREQUEST, CAUSE_CTICCMSIP401UNAUTHORIZED,
CAUSE_CTICCMSIP402PAYMENTREQUIRED, CAUSE_CTICCMSIP403FORBIDDEN,
CAUSE_CTICCMSIP404NOTFOUND, CAUSE_CTICCMSIP405METHODNOTALLOWED,
CAUSE_CTICCMSIP406NOTACCEPTABLE,
CAUSE_CTICCMSIP407PROXYAUTHENTICATIONREQUIRED,
CAUSE_CTICCMSIP408REQUESTTIMEOUT, CAUSE_CTICCMSIP410GONE,
CAUSE_CTICCMSIP411LENGTHREQUIRED,
CAUSE_CTICCMSIP413REQUESTENTITYTOOLONG,
CAUSE_CTICCMSIP414REQUESTURITOOLONG,
CAUSE_CTICCMSIP415UNSUPPORTEDMEDIATYPE,
CAUSE_CTICCMSIP416UNSUPPORTEDURISCHEME,
CAUSE_CTICCMSIP420BADEXTENSION, CAUSE_CTICCMSIP421EXTENSTIONREQUIRED,
CAUSE_CTICCMSIP423INTERVALTOOBRIEF,
CAUSE_CTICCMSIP480TEMPORARILYUNAVAILABLE,
CAUSE_CTICCMSIP481CALLLEGDOESNOTEXIST, CAUSE_CTICCMSIP482LOOPDETECTED,
CAUSE_CTICCMSIP483TOOMANYHOOPS, CAUSE_CTICCMSIP484ADDRESSINCOMPLETE,
CAUSE_CTICCMSIP485AMBIGUOUS, CAUSE_CTICCMSIP486BUSYHERE,
CAUSE_CTICCMSIP487REQUESTTERMINATED,
CAUSE_CTICCMSIP488NOTACCEPTABLEHERE, CAUSE_CTICCMSIP491REQUESTPENDING,
CAUSE_CTICCMSIP493UNDECIPHERABLE,
CAUSE_CTICCMSIP500SERVERINTERNALERROR,
CAUSE_CTICCMSIP501NOTIMPLEMENTED, CAUSE_CTICCMSIP502BADGATEWAY,
CAUSE_CTICCMSIP503SERVICEUNAVAILABLE, CAUSE_CTICCMSIP504SERVERTIMEOUT,
CAUSE_CTICCMSIP505SIPVERSIONNOTSUPPORTED,
CAUSE_CTICCMSIP513MESSAGETOOLARGE, CAUSE_CTICCMSIP600BUSYEVERYWHERE,
CAUSE_CTICCMSIP603DECLINE, CAUSE_CTICCMSIP604DOESNOTEXISTANYWHERE,
CAUSE_CTICCMSIP606NOTACCEPTABLE, CAUSE_CTICONFERENCEFULL,
CAUSE_CTIDEVICENOTPREEMPTABLE, CAUSE_CTIDROPCONFEREE,
CAUSE_CTIMANAGER_FAILURE, CAUSE_CTIPRECEDENCECALLBLOCKED,
CAUSE_CTIPRECEDENCELEVELEXCEEDED,
CAUSE_CTIPRECEDENCEOUTOFBANDWIDTH, CAUSE_CTIPREEMPTFORREUSE,
CAUSE_CTIPREEMPTNOREUSE, CAUSE_DESTINATIONOUTOFORDER,
CAUSE_DESTNUMMISSANDDCNOTSUB, CAUSE_DPARK, CAUSE_DPARK_REMINDER,
CAUSE_DPARK_UNPARK, CAUSE_EXCHANGEROUTINGERROR, CAUSE_FAC_CMC,

CAUSE_FACILITYREJECTED, CAUSE_IDENTIFIEDCHANDOESNOTEXIST, CAUSE_IENIMPL, CAUSE_INBOUNDBLINDTRANSFER, CAUSE_INBOUNDCONFERENCE, CAUSE_INBOUNDTRANSFER, CAUSE_INCOMINGCALLBARRED, CAUSE_INCOMPATABLEDDESTINATION, CAUSE_INTERWORKINGUNSPECIFIED, CAUSE_INVALIDCALLREFVALUE, CAUSE_INVALIDIECONTENTS, CAUSE_INVALIDMESSAGEUNSPECIFIED, CAUSE_INVALIDNUMBERFORMAT, CAUSE_INVALIDTRANSITNETSEL, CAUSE_MANDATORYIEMISSING, CAUSE_MSGNCOMPATABLEWCS, CAUSE_MSGTYPENCOMPATWCS, CAUSE_MSGTYPENIMPL, CAUSE_NETOUTOFORDER, CAUSE_NOANSWERFROMUSER, CAUSE_NOCALLSUSPENDED, CAUSE_NOCIRCAVAIL, CAUSE_NOERROR, CAUSE_NONSELECTEDUSERCLEARING, CAUSE_NORMALCALLCLEARING, CAUSE_NORMALUNSPECIFIED, CAUSE_NOROUTETODDESTINATION, CAUSE_NOROUTETOTRANSITNET, CAUSE_NOUSERRESPONDING, CAUSE_NUMBERCHANGED, CAUSE_ONLYRDIVEARERCAPAVAIL, CAUSE_OUTBOUNDCONFERENCE, CAUSE_OUTBOUNDTRANSFER, CAUSE_OUTOFBANDWIDTH, CAUSE_PROTOCOLERRORUNSPECIFIED, CAUSE_QSIG_PR, CAUSE_QUALOFSERVNAVAIL, CAUSE_QUIET_CLEAR, CAUSE_RECOVERYONTIMEREXPIRY, CAUSE_REDIRECTED, CAUSE_REQCALLIDHASBEENCLEARED, CAUSE_REQCIRCNAVIL, CAUSE_REQFACILITYNIMPL, CAUSE_REQFACILITYNOTSUBSCRIBED, CAUSE_RESOURCESNAVAIL, CAUSE_RESPONSETOSTATUSENQUIRY, CAUSE_SERVNOTAVAILUNSPECIFIED, CAUSE_SERVOPERATIONVIOLATED, CAUSE_SERVOROPTNAVAILORIMPL, CAUSE_SUBSCRIBERABSENT, CAUSE_SUSPCALLBUTNOTTHISONE, CAUSE_SWITCHINGEQUIPMENTCONGESTION, CAUSE_TEMPORARYFAILURE, CAUSE_UNALLOCATEDNUMBER, CAUSE_USERBUSY

インターフェイス javax.telephony.events.Ev から

CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から

CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

表 6-65　CiscoCallSecurityStatusChangedEv のメソッド

インターフェイス	メソッド	説明
javax.telephony.events.Ev	getID()	指定元：interface javax.telephony.events.Ev の getID
getCallSecurityStatus()	getCallSecurityStatus()	コール セキュリティ ステータスを返します。このインターフェイスは、以下を返すことができます。CiscoCall.CALLSECURITY_UNKNOWN, CiscoCall.CALLSECURITY_NOTAUTHENTICATED, CiscoCall.CALLSECURITY_AUTHENTICATED, CiscoCall.CALLSECURITY_ENCRYPTED

継承したメソッド

インターフェイス com.cisco.jtapi.extensions.CiscoCallEv から
getCiscoCause, getCiscoFeatureReason

インターフェイス javax.telephony.events.Ev から
getCause, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.CallEv から
getCall

インターフェイス javax.telephony.events.Ev から
getCause, getMetaCode, getObserved, isNewMetaEvent

関連資料

詳細については、「定数フィールド値」（P.F-1）を参照してください。

CiscoConferenceChain

このインターフェイスは、会議チェーンで繋がれている電話会議の会議チェーン接続へのリンクを提供します。このオブジェクトは CiscoConferenceChainAddedEv および CiscoConferenceChainRemovedEv から取得できます。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1)	変更を記録するために履歴表が作成されました。

宣言

public interface CiscoConferenceChain

フィールド

なし

メソッド

表 6-66　CiscoConferenceChain のメソッド

インターフェイス	メソッド	説明
javax.telephony.Connection[]	getChainedConferenceConnections()	1 つの会議にチェーニングされている電話会議の接続の配列を返します。アプリケーションは、このリストを使用してすべての繋がれている電話会議を取得できます。会議にチェーニングされたすべての接続のリストを取得するには、プロバイダーは、各会議で最低 1 つの通話者にオブザーバを必要とします。
CiscoCall[]	getChainedConferenceCalls()	1 つの会議にチェーニングされているコールの配列を返します。このインターフェイスでは、会議チェーンに含まれているプロバイダーで監視されているコールだけが返されます。

関連資料

詳細については、CiscoConferenceChainAddedEv と CiscoConferenceChainRemovedEv を参照してください。

CiscoConferenceChainAddedEv

会議チェーン接続がコールに追加されると、CiscoConferenceChainAddedEv イベントが送信されます。このイベントは、新しい会議チェーン接続が追加されるたびに送信されます。このイベントは CallControlCallObserver インターフェイスによって報告されます。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

すべてのスーパーインターフェイス

javax.telephony.events.CallEv, CiscoCallEv, CiscoEv, javax.telephony.events.Ev

宣言

public interface CiscoConferenceChainAddedEv extends CiscoCallEv

フィールド

表 6-67　CiscoConferenceChainAddedEv のフィールド

インターフェイス	フィールド
static int	ID

継承したフィールド

インターフェイス com.cisco.jtapi.extensions.CiscoCallEv から
CAUSE_ACCESSINFORMATIONDISCARDED, CAUSE_BARGE,
CAUSE_BCBPRESENTLYAVAIL,CAUSE_BCNAUTHORIZED, CAUSE_BEARERCAPNIMPL,
CAUSE_CALLBEINGDELIVERED, CAUSE_CALLIDINUSE,
CAUSE_CALLMANAGER_FAILURE, CAUSE_CALLREJECTED, CAUSE_CALLSPLIT,
CAUSE_CHANTYPENIMPL, CAUSE_CHANUNACCEPTABLE,
CAUSE_CTICCMSIP400BADREQUEST, CAUSE_CTICCMSIP401UNAUTHORIZED,
CAUSE_CTICCMSIP402PAYMENTREQUIRED,CAUSE_CTICCMSIP403FORBIDDEN,
CAUSE_CTICCMSIP404NOTFOUND, CAUSE_CTICCMSIP405METHODNOTALLOWED,
CAUSE_CTICCMSIP406NOTACCEPTABLE,
CAUSE_CTICCMSIP407PROXYAUTHENTICATIONREQUIRED,
CAUSE_CTICCMSIP408REQUESTTIMEOUT, CAUSE_CTICCMSIP410GONE,
CAUSE_CTICCMSIP411LENGTHREQUIRED,
CAUSE_CTICCMSIP413REQUESTENTITYTOOLONG,
CAUSE_CTICCMSIP414REQUESTURITOOLONG,
CAUSE_CTICCMSIP415UNSUPPORTEDMEDIATYPE,
CAUSE_CTICCMSIP416UNSUPPORTEDURISCHEME,
CAUSE_CTICCMSIP420BADEXTENSION, CAUSE_CTICCMSIP421EXTENSTIONREQUIRED,
CAUSE_CTICCMSIP423INTERVALTOOBRIEF,
CAUSE_CTICCMSIP480TEMPORARILYUNAVAILABLE,
CAUSE_CTICCMSIP481CALLLEGDOESNOTEXIST, CAUSE_CTICCMSIP482LOOPDETECTED,
CAUSE_CTICCMSIP483TOOMANYHOOPS, CAUSE_CTICCMSIP484ADDRESSINCOMPLETE,
CAUSE_CTICCMSIP485AMBIGUOUS, CAUSE_CTICCMSIP486BUSYHERE,
CAUSE_CTICCMSIP487REQUESTTERMINATED,
CAUSE_CTICCMSIP488NOTACCEPTABLEHERE, CAUSE_CTICCMSIP491REQUESTPENDING,
CAUSE_CTICCMSIP493UNDECIPHERABLE,
CAUSE_CTICCMSIP500SERVERINTERNALERROR,
CAUSE_CTICCMSIP501NOTIMPLEMENTED, CAUSE_CTICCMSIP502BADGATEWAY,
CAUSE_CTICCMSIP503SERVICEUNAVAILABLE, CAUSE_CTICCMSIP504SERVERTIMEOUT,
CAUSE_CTICCMSIP505SIPVERSIONNOTSUPPORTED,

CAUSE_CTICCMSIP513MESSAGETOOLARGE, CAUSE_CTICCMSIP600BUSYEVERYWHERE, CAUSE_CTICCMSIP603DECLINE, CAUSE_CTICCMSIP604DOESNOTEXISTANYWHERE, CAUSE_CTICCMSIP606NOTACCEPTABLE, CAUSE_CTICONFERENCEFULL, CAUSE_CTIDEVICENOTPREEMPTABLE, CAUSE_CTIDROPCONFEREE, CAUSE_CTIMANAGER_FAILURE, CAUSE_CTIPRECEDENCECALLBLOCKED, CAUSE_CTIPRECEDENCELEVELEXCEEDED, CAUSE_CTIPRECEDENCEOUTOFBANDWIDTH, CAUSE_CTIPREEMPTFORREUSE, CAUSE_CTIPREEMPTNOREUSE, CAUSE_DESTINATIONOUTOFORDER, CAUSE_DESTNUMMISSANDDCNOTSUB, CAUSE_DPARK, CAUSE_DPARK_REMINDER, CAUSE_DPARK_UNPARK, CAUSE_EXCHANGEROUTINGERROR, CAUSE_FAC_CMC, CAUSE_FACILITYREJECTED, CAUSE_IDENTIFIEDCHANDOESNOTEXIST, CAUSE_IENIMPL, CAUSE_INBOUNDBLINDTRANSFER, CAUSE_INBOUNDCONFERENCE, CAUSE_INBOUNDTRANSFER, CAUSE_INCOMINGCALLBARRED, CAUSE_INCOMPATABLEDDESTINATION, CAUSE_INTERWORKINGUNSPECIFIED, CAUSE_INVALIDCALLREFVALUE, CAUSE_INVALIDIECONTENTS, CAUSE_INVALIDMESSAGEUNSPECIFIED, CAUSE_INVALIDNUMBERFORMAT, CAUSE_INVALIDTRANSITNETSEL, CAUSE_MANDATORYIEMISSING, CAUSE_MSGNCOMPATABLEWCS, CAUSE_MSGTYPENCOMPATWCS, CAUSE_MSGTYPENIMPL, CAUSE_NETOUTOFORDER, CAUSE_NOANSWERFROMUSER, CAUSE_NOCALLSUSPENDED, CAUSE_NOCIRCAVAIL, CAUSE_NOERROR, CAUSE_NONSELECTEDUSERCLEARING, CAUSE_NORMALCALLCLEARING, CAUSE_NORMALUNSPECIFIED, CAUSE_NOROUTETODDESTINATION, CAUSE_NOROUTETOTRANSITNET, CAUSE_NOUSERRESPONDING, CAUSE_NUMBERCHANGED, CAUSE_ONLYRDIVEARERCAPAVAIL, CAUSE_OUTBOUNDCONFERENCE, CAUSE_OUTBOUNDTRANSFER, CAUSE_OUTOFBANDWIDTH, CAUSE_PROTOCOLERRORUNSPECIFIED, CAUSE_QSIG_PR, CAUSE_QUALOFSERVNAVAIL, CAUSE_QUIET_CLEAR, CAUSE_RECOVERYONTIMEREXPIRY, CAUSE_REDIRECTED, CAUSE_REQCALLIDHASBEENCLEARED, CAUSE_REQCIRCNAVIL, CAUSE_REQFACILITYNIMPL, CAUSE_REQFACILITYNOTSUBSCRIBED, CAUSE_RESOURCESNAVAIL, CAUSE_RESPONSETOSTATUSENQUIRY, CAUSE_SERVNOTAVAILUNSPECIFIED, CAUSE_SERVOPERATIONVIOLATED, CAUSE_SERVOROPTNAVAILORIMPL, CAUSE_SUBSCRIBERABSENT, CAUSE_SUSPCALLBUTNOTTHISONE, CAUSE_SWITCHINGEQUIPMENTCONGESTION, CAUSE_TEMPORARYFAILURE, CAUSE_UNALLOCATEDNUMBER, CAUSE_USERBUSY

インターフェイス javax.telephony.events.Ev から

CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から

CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY,

META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

表 6-68　CiscoConferenceChainAddedEv のメソッド

インターフェイス	メソッド	説明
javax.telephony.Connection	getAddedConnection()	直前にコールに追加された会議チェーン接続を返します。
CiscoConferenceChain	getConferenceChain()	チェーニングされているコールのすべての会議接続が含まれている CiscoConferenceChain を返します。

継承したメソッド

インターフェイス com.cisco.jtapi.extensions.CiscoCallEv から
getCiscoCause, getCiscoFeatureReason

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.CallEv から
getCall

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

詳細については、「定数フィールド値」（P.F-1）を参照してください。

CiscoConferenceChainRemovedEv

会議チェーン接続がコールから削除されると、CiscoConferenceChainRemovedEv イベントが送信されます。このイベントは、会議チェーン接続が削除されるたびに送信されます。このイベントは CallControlCallObserver インターフェイスによって報告されます。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

javax.telephony.events.CallEv, CiscoCallEv, CiscoEv, javax.telephony.events.Ev

宣言

public interface CiscoConferenceChainRemovedEv extends CiscoCallEv

フィールド

表 6-69　CiscoConferenceChainRemovedEv のフィールド

インターフェイス	フィールド
static int	ID

継承したフィールド

インターフェイス com.cisco.jtapi.extensions.CiscoCallEv から
CAUSE_ACCESSINFORMATIONDISCARDED, CAUSE_BARGE,
CAUSE_BCBPRESENTLYAVAIL, CAUSE_BCNAUTHORIZED, CAUSE_BEARERCAPNIMPL,
CAUSE_CALLBEINGDELIVERED, CAUSE_CALLIDINUSE,
CAUSE_CALLMANAGER_FAILURE, CAUSE_CALLREJECTED, CAUSE_CALLSPLIT,
CAUSE_CHANTYPENIMPL, CAUSE_CHANUNACCEPTABLE,
CAUSE_CTICCMSIP400BADREQUEST, CAUSE_CTICCMSIP401UNAUTHORIZED,
CAUSE_CTICCMSIP402PAYMENTREQUIRED, CAUSE_CTICCMSIP403FORBIDDEN,
CAUSE_CTICCMSIP404NOTFOUND, CAUSE_CTICCMSIP405METHODNOTALLOWED,
CAUSE_CTICCMSIP406NOTACCEPTABLE,
CAUSE_CTICCMSIP407PROXYAUTHENTICATIONREQUIRED,
CAUSE_CTICCMSIP408REQUESTTIMEOUT, CAUSE_CTICCMSIP410GONE,
CAUSE_CTICCMSIP411LENGTHREQUIRED,
CAUSE_CTICCMSIP413REQUESTENTITYTOOLONG,
CAUSE_CTICCMSIP414REQUESTURITOOLONG,
CAUSE_CTICCMSIP415UNSUPPORTEDMEDIATYPE,
CAUSE_CTICCMSIP416UNSUPPORTEDURISCHEME,
CAUSE_CTICCMSIP420BADEXTENSION, CAUSE_CTICCMSIP421EXTENSTIONREQUIRED,
CAUSE_CTICCMSIP423INTERVALTOOBRIEF,
CAUSE_CTICCMSIP480TEMPORARILYUNAVAILABLE,
CAUSE_CTICCMSIP481CALLLEGDOESNOTEXIST, CAUSE_CTICCMSIP482LOOPDETECTED,
CAUSE_CTICCMSIP483TOOMANYHOOPS, CAUSE_CTICCMSIP484ADDRESSINCOMPLETE,
CAUSE_CTICCMSIP485AMBIGUOUS, CAUSE_CTICCMSIP486BUSYHERE,
CAUSE_CTICCMSIP487REQUESTTERMINATED,
CAUSE_CTICCMSIP488NOTACCEPTABLEHERE, CAUSE_CTICCMSIP491REQUESTPENDING,
CAUSE_CTICCMSIP493UNDECIPHERABLE,
CAUSE_CTICCMSIP500SERVERINTERNALERROR,
CAUSE_CTICCMSIP501NOTIMPLEMENTED, CAUSE_CTICCMSIP502BADGATEWAY,
CAUSE_CTICCMSIP503SERVICEUNAVAILABLE, CAUSE_CTICCMSIP504SERVERTIMEOUT,
CAUSE_CTICCMSIP505SIPVERSIONNOTSUPPORTED,

CAUSE_CTICCMSIP513MESSAGETOOLARGE, CAUSE_CTICCMSIP600BUSYEVERYWHERE, CAUSE_CTICCMSIP603DECLINE, CAUSE_CTICCMSIP604DOESNOTEXISTANYWHERE, CAUSE_CTICCMSIP606NOTACCEPTABLE, CAUSE_CTICONFERENCEFULL, CAUSE_CTIDEVICENOTPREEMPTABLE, CAUSE_CTIDROPCONFEREE, CAUSE_CTIMANAGER_FAILURE, CAUSE_CTIPRECEDENCECALLBLOCKED, CAUSE_CTIPRECEDENCELEVELEXCEEDED, CAUSE_CTIPRECEDENCEOUTOFBANDWIDTH, CAUSE_CTIPREEMPTFORREUSE, CAUSE_CTIPREEMPTNOREUSE, CAUSE_DESTINATIONOUTOFORDER, CAUSE_DESTNUMMISSANDDCNOTSUB, CAUSE_DPARK, CAUSE_DPARK_REMINDER, CAUSE_DPARK_UNPARK, CAUSE_EXCHANGEROUTINGERROR, CAUSE_FAC_CMC, CAUSE_FACILITYREJECTED, CAUSE_IDENTIFIEDCHANDOESNOTEXIST, CAUSE_IENIMPL, CAUSE_INBOUNDBLINDTRANSFER, CAUSE_INBOUNDCONFERENCE, CAUSE_INBOUNDTRANSFER, CAUSE_INCOMINGCALLBARRED, CAUSE_INCOMPATABLEDDESTINATION, CAUSE_INTERWORKINGUNSPECIFIED, CAUSE_INVALIDCALLREFVALUE, CAUSE_INVALIDIECONTENTS, CAUSE_INVALIDMESSAGEUNSPECIFIED, CAUSE_INVALIDNUMBERFORMAT, CAUSE_INVALIDTRANSITNETSEL, CAUSE_MANDATORYIEMISSING, CAUSE_MSGNCOMPATABLEWCS, CAUSE_MSGTYPENCOMPATWCS, CAUSE_MSGTYPENIMPL, CAUSE_NETOUTOFORDER, CAUSE_NOANSWERFROMUSER, CAUSE_NOCALLSUSPENDED, CAUSE_NOCIRCAVAIL, CAUSE_NOERROR, CAUSE_NONSELECTEDUSERCLEARING, CAUSE_NORMALCALLCLEARING, CAUSE_NORMALUNSPECIFIED, CAUSE_NOROUTETODDESTINATION, CAUSE_NOROUTETOTRANSITNET, CAUSE_NOUSERRESPONDING, CAUSE_NUMBERCHANGED, CAUSE_ONLYRDIVEARERCAPAVAIL, CAUSE_OUTBOUNDCONFERENCE, CAUSE_OUTBOUNDTRANSFER, CAUSE_OUTOFBANDWIDTH, CAUSE_PROTOCOLERRORUNSPECIFIED, CAUSE_QSIG_PR, CAUSE_QUALOFSERVNAVAIL, CAUSE_QUIET_CLEAR, CAUSE_RECOVERYONTIMEREXPIRY, CAUSE_REDIRECTED, CAUSE_REQCALLIDHASBEENCLEARED, CAUSE_REQCIRCNAVIL, CAUSE_REQFACILITYNIMPL, CAUSE_REQFACILITYNOTSUBSCRIBED, CAUSE_RESOURCESNAVAIL, CAUSE_RESPONSETOSTATUSENQUIRY, CAUSE_SERVNOTAVAILUNSPECIFIED, CAUSE_SERVOPERATIONVIOLATED, CAUSE_SERVOROPTNAVAILORIMPL, CAUSE_SUBSCRIBERABSENT, CAUSE_SUSPCALLBUTNOTTHISONE, CAUSE_SWITCHINGEQUIPMENTCONGESTION, CAUSE_TEMPORARYFAILURE, CAUSE_UNALLOCATEDNUMBER,CAUSE_USERBUSY

インターフェイス javax.telephony.events.Ev から

CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から

CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY,

META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

表 6-70　CiscoConferenceChainRemovedEf のメソッド

インターフェイス	メソッド	説明
CiscoConferenceChain	getConferenceChain()	チェーニングされているコールのすべての会議接続が含まれている CiscoConferenceChain を返します。戻り値：接続。
javax.telephony.Connection	getRemovedConnection()	直前にコールから削除された会議チェーン接続を返します。戻り値：CiscoConferenceChain。

継承したメソッド

インターフェイス com.cisco.jtapi.extensions.CiscoCallEv から
getCiscoCause, getCiscoFeatureReason

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.CallEv から
getCall

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

詳細については、「定数フィールド値」（P.F-1）を参照してください。

CiscoConferenceEndEv

CiscoConferenceEndEv イベントは、会議の動作が完了したことを示します。このイベントは CallControlCallObserver インターフェイスによって報告されます。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

javax.telephony.events.CallEv, CiscoCallEv, CiscoEv, javax.telephony.events.Ev

宣言

public interface CiscoConferenceEndEv extends CiscoCallEv

フィールド

表 6-71　CiscoConferenceEndEv のフィールド

インターフェイス	フィールド
static int	ID

継承したフィールド

インターフェイス com.cisco.jtapi.extensions.CiscoCallEv から
CAUSE_ACCESSINFORMATIONDISCARDED, CAUSE_BARGE,
CAUSE_BCBPRESENTLYAVAIL, CAUSE_BCNAUTHORIZED, CAUSE_BEARERCAPNIMPL,
CAUSE_CALLBEINGDELIVERED, CAUSE_CALLIDINUSE,
CAUSE_CALLMANAGER_FAILURE, CAUSE_CALLREJECTED, CAUSE_CALLSPLIT,
CAUSE_CHANTYPENIMPL, CAUSE_CHANUNACCEPTABLE,
CAUSE_CTICCMSIP400BADREQUEST, CAUSE_CTICCMSIP401UNAUTHORIZED,
CAUSE_CTICCMSIP402PAYMENTREQUIRED, CAUSE_CTICCMSIP403FORBIDDEN,
CAUSE_CTICCMSIP404NOTFOUND, CAUSE_CTICCMSIP405METHODNOTALLOWED,
CAUSE_CTICCMSIP406NOTACCEPTABLE,
CAUSE_CTICCMSIP407PROXYAUTHENTICATIONREQUIRED,
CAUSE_CTICCMSIP408REQUESTTIMEOUT, CAUSE_CTICCMSIP410GONE,
CAUSE_CTICCMSIP411LENGTHREQUIRED,
CAUSE_CTICCMSIP413REQUESTENTITYTOOLONG,
CAUSE_CTICCMSIP414REQUESTURITOOLONG,
CAUSE_CTICCMSIP415UNSUPPORTEDMEDIATYPE,
CAUSE_CTICCMSIP416UNSUPPORTEDURISCHEME,
CAUSE_CTICCMSIP420BADEXTENSION, CAUSE_CTICCMSIP421EXTENSTIONREQUIRED,
CAUSE_CTICCMSIP423INTERVALTOOBRIEF,
CAUSE_CTICCMSIP480TEMPORARILYUNAVAILABLE,
CAUSE_CTICCMSIP481CALLLEGDOESNOTEXIST, CAUSE_CTICCMSIP482LOOPDETECTED,
CAUSE_CTICCMSIP483TOOMANYHOOPS, CAUSE_CTICCMSIP484ADDRESSINCOMPLETE,
CAUSE_CTICCMSIP485AMBIGUOUS, CAUSE_CTICCMSIP486BUSYHERE,
CAUSE_CTICCMSIP487REQUESTTERMINATED,
CAUSE_CTICCMSIP488NOTACCEPTABLEHERE, CAUSE_CTICCMSIP491REQUESTPENDING,
CAUSE_CTICCMSIP493UNDECIPHERABLE,
CAUSE_CTICCMSIP500SERVERINTERNALERROR,
CAUSE_CTICCMSIP501NOTIMPLEMENTED, CAUSE_CTICCMSIP502BADGATEWAY,
CAUSE_CTICCMSIP503SERVICEUNAVAILABLE, CAUSE_CTICCMSIP504SERVERTIMEOUT,
CAUSE_CTICCMSIP505SIPVERSIONNOTSUPPORTED,

CAUSE_CTICCMSIP513MESSAGETOOLARGE, CAUSE_CTICCMSIP600BUSYEVERYWHERE, CAUSE_CTICCMSIP603DECLINE, CAUSE_CTICCMSIP604DOESNOTEXISTANYWHERE, CAUSE_CTICCMSIP606NOTACCEPTABLE, CAUSE_CTICONFERENCEFULL, CAUSE_CTIDEVICENOTPREEMPTABLE, CAUSE_CTIDROPCONFEREE, CAUSE_CTIMANAGER_FAILURE, CAUSE_CTIPRECEDENCECALLBLOCKED, CAUSE_CTIPRECEDENCELEVELEXCEEDED, CAUSE_CTIPRECEDENCEOUTOFBANDWIDTH, CAUSE_CTIPREEMPTFORREUSE, CAUSE_CTIPREEMPTNOREUSE, CAUSE_DESTINATIONOUTOFORDER, CAUSE_DESTNUMMISSANDDCNOTSUB, CAUSE_DPARK, CAUSE_DPARK_REMINDER, CAUSE_DPARK_UNPARK, CAUSE_EXCHANGEROUTINGERROR, CAUSE_FAC_CMC, CAUSE_FACILITYREJECTED, CAUSE_IDENTIFIEDCHANDOESNOTEXIST, CAUSE_IENIMPL, CAUSE_INBOUNDBLINDTRANSFER, CAUSE_INBOUNDCONFERENCE, CAUSE_INBOUNDTRANSFER, CAUSE_INCOMINGCALLBARRED, CAUSE_INCOMPATABLEDDESTINATION, CAUSE_INTERWORKINGUNSPECIFIED, CAUSE_INVALIDCALLREFVALUE, CAUSE_INVALIDIECONTENTS, CAUSE_INVALIDMESSAGEUNSPECIFIED, CAUSE_INVALIDNUMBERFORMAT, CAUSE_INVALIDTRANSITNETSEL, CAUSE_MANDATORYIEMISSING, CAUSE_MSGNCOMPATABLEWCS, CAUSE_MSGTYPENCOMPATWCS, CAUSE_MSGTYPENIMPL, CAUSE_NETOUTOFORDER, CAUSE_NOANSWERFROMUSER, CAUSE_NOCALLSUSPENDED, CAUSE_NOCIRCAVAIL, CAUSE_NOERROR, CAUSE_NONSELECTEDUSERCLEARING, CAUSE_NORMALCALLCLEARING, CAUSE_NORMALUNSPECIFIED, CAUSE_NOROUTETODDESTINATION, CAUSE_NOROUTETOTRANSITNET, CAUSE_NOUSERRESPONDING, CAUSE_NUMBERCHANGED, CAUSE_ONLYRDIVEARERCAPAVAIL, CAUSE_OUTBOUNDCONFERENCE, CAUSE_OUTBOUNDTRANSFER, CAUSE_OUTOFBANDWIDTH, CAUSE_PROTOCOLERRORUNSPECIFIED, CAUSE_QSIG_PR, CAUSE_QUALOFSERVNAVAIL, CAUSE_QUIET_CLEAR, CAUSE_RECOVERYONTIMEREXPIRY, CAUSE_REDIRECTED, CAUSE_REQCALLIDHASBEENCLEARED, CAUSE_REQCIRCNAVIL, CAUSE_REQFACILITYNIMPL, CAUSE_REQFACILITYNOTSUBSCRIBED, CAUSE_RESOURCESNAVAIL, CAUSE_RESPONSETOSTATUSENQUIRY, CAUSE_SERVNOTAVAILUNSPECIFIED, CAUSE_SERVOPERATIONVIOLATED, CAUSE_SERVOROPTNAVAILORIMPL, CAUSE_SUBSCRIBERABSENT, CAUSE_SUSPCALLBUTNOTTHISONE, CAUSE_SWITCHINGEQUIPMENTCONGESTION, CAUSE_TEMPORARYFAILURE, CAUSE_UNALLOCATEDNUMBER, CAUSE_USERBUSY

インターフェイス javax.telephony.events.Ev から

CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から

CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY,

META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

表 6-72　CiscoConferenceEndEv のメソッド

インターフェイス	メソッド	説明
javax.telephony.Address	getConferenceControllerAddress()	現在このコール（最初のコール）の会議コントローラとして機能しているアドレスを返します。
javax.telephony.Call	getConferencedCall()	マージされたコールを返します。このコールは Call.INVALID の状態です。
javax.telephony.Call[]	getFailedCalls()	会議に参加できなかったコールのリストを返します。
javax.telephony.Call	getFinalCall()	会議の終了後にアクティブになっているコールを返します。
javax.telephony.TerminalConnection	getHeldConferenceController()	現在このコール（最後のコール）の会議コントローラとして機能している TerminalConnection を返します。これは、会議開始時の状態が HELD であった TerminalConnection です。会議コントローラが監視されていない場合、このメソッドは null または TerminalConnection を返します。
javax.telephony.TerminalConnection	getTalkingConferenceController()	現在このコール（最初のコール）の会議コントローラとして機能している TerminalConnection を返します。これは、当初 TALKING 状態だった TerminalConnection です。会議コントローラが監視されていない場合、このメソッドは null または TerminalConnection を返します。

表 6-72　CiscoConferenceEndEv のメソッド

インターフェイス	メソッド	説明
boolean	isSuccess()	会議 が正常に実行されたか失敗したかに応じて、ブール値（true または false）を返します。アプリケーションは、このインターフェイスを使用して、会議が正常に実行されたかどうかを確認できます。 会議が正常に実行されない原因として、次のような状況が考えられます。 • アプリケーションが Call.conference(otherCalls[]) 要求を発行し、1 つ以上のコールが会議に参加できなかった場合、この会議は失敗と見なされます。getFailedCalls() を使用して、失敗したコールを見つけます。 • 会議ブリッジがなく、会議が完了できなかった場合。getFailedCalls() を使用して、会議に参加できなかったコールのリストを取得します。 • 会議が開催できる前に会議に参加していた側がドロップした場合。

継承したメソッド

インターフェイス com.cisco.jtapi.extensions.CiscoCallEv から
getCiscoCause, getCiscoFeatureReason

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.CallEv から
getCall

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

「定数フィールド値」（P.F-1）を参照してください。

isSuccess() も参照してください。

CiscoConferenceStartEv

CiscoConferenceStartEv イベントは、会議の動作が開始されたことを示します。このイベントは CallControlCallObserver インターフェイスによって報告されます。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	回線をまたいで参加（Join Across Lines）/Connected Conference 機能に getControllerTerminalName() メソッドが追加されました。

スーパーインターフェイス

javax.telephony.events.CallEv, CiscoCallEv, CiscoEv, javax.telephony.events.Ev

宣言

public interface CiscoConferenceStartEv extends CiscoCallEv

フィールド

表 6-73　CiscoConferenceStartEv のフィールド

インターフェイス	フィールド
static int	ID

継承したフィールド

インターフェイス com.cisco.jtapi.extensions.CiscoCallEv から
CAUSE_ACCESSINFORMATIONDISCARDED, CAUSE_BARGE, CAUSE_BCBPRESENTLYAVAIL, CAUSE_BCNAUTHORIZED, CAUSE_BEARERCAPNIMPL, CAUSE_CALLBEINGDELIVERED, CAUSE_CALLIDINUSE, CAUSE_CALLMANAGER_FAILURE, CAUSE_CALLREJECTED, CAUSE_CALLSPLIT, CAUSE_CHANTYPENIMPL, CAUSE_CHANUNACCEPTABLE, CAUSE_CTICCMSIP400BADREQUEST, CAUSE_CTICCMSIP401UNAUTHORIZED, CAUSE_CTICCMSIP402PAYMENTREQUIRED, CAUSE_CTICCMSIP403FORBIDDEN, CAUSE_CTICCMSIP404NOTFOUND, CAUSE_CTICCMSIP405METHODNOTALLOWED, CAUSE_CTICCMSIP406NOTACCEPTABLE, CAUSE_CTICCMSIP407PROXYAUTHENTICATIONREQUIRED, CAUSE_CTICCMSIP408REQUESTTIMEOUT, CAUSE_CTICCMSIP410GONE, CAUSE_CTICCMSIP411LENGTHREQUIRED, CAUSE_CTICCMSIP413REQUESTENTITYTOOLONG, CAUSE_CTICCMSIP414REQUESTURITOOLONG, CAUSE_CTICCMSIP415UNSUPPORTEDMEDIATYPE, CAUSE_CTICCMSIP416UNSUPPORTEDURISCHEME,

CAUSE_CTICCMSIP420BADEXTENSION, CAUSE_CTICCMSIP421EXTENSTIONREQUIRED,
CAUSE_CTICCMSIP423INTERVALTOOBRIEF,
CAUSE_CTICCMSIP480TEMPORARILYUNAVAILABLE,
CAUSE_CTICCMSIP481CALLLEGDOESNOTEXIST, CAUSE_CTICCMSIP482LOOPDETECTED,
CAUSE_CTICCMSIP483TOOMANYHOOPS, CAUSE_CTICCMSIP484ADDRESSINCOMPLETE,
CAUSE_CTICCMSIP485AMBIGUOUS, CAUSE_CTICCMSIP486BUSYHERE,
CAUSE_CTICCMSIP487REQUESTTERMINATED,
CAUSE_CTICCMSIP488NOTACCEPTABLEHERE, CAUSE_CTICCMSIP491REQUESTPENDING,
CAUSE_CTICCMSIP493UNDECIPHERABLE,
CAUSE_CTICCMSIP500SERVERINTERNALERROR,
CAUSE_CTICCMSIP501NOTIMPLEMENTED, CAUSE_CTICCMSIP502BADGATEWAY,
CAUSE_CTICCMSIP503SERVICEUNAVAILABLE, CAUSE_CTICCMSIP504SERVERTIMEOUT,
CAUSE_CTICCMSIP505SIPVERSIONNOTSUPPORTED,
CAUSE_CTICCMSIP513MESSAGETOOLARGE, CAUSE_CTICCMSIP600BUSYEVERYWHERE,
CAUSE_CTICCMSIP603DECLINE, CAUSE_CTICCMSIP604DOESNOTEXISTANYWHERE,
CAUSE_CTICCMSIP606NOTACCEPTABLE, CAUSE_CTICONFERENCEFULL,
CAUSE_CTIDEVICENOTPREEMPTABLE, CAUSE_CTIDROPCONFEREE,
CAUSE_CTIMANAGER_FAILURE, CAUSE_CTIPRECEDENCECALLBLOCKED,
CAUSE_CTIPRECEDENCELEVELEXCEEDED,
CAUSE_CTIPRECEDENCEOUTOFBANDWIDTH, CAUSE_CTIPREEMPTFORREUSE,
CAUSE_CTIPREEMPTNOREUSE, CAUSE_DESTINATIONOUTOFORDER,
CAUSE_DESTNUMMISSANDDCNOTSUB, CAUSE_DPARK, CAUSE_DPARK_REMINDER,
CAUSE_DPARK_UNPARK, CAUSE_EXCHANGEROUTINGERROR, CAUSE_FAC_CMC,
CAUSE_FACILITYREJECTED, CAUSE_IDENTIFIEDCHANDOESNOTEXIST, CAUSE_IENIMPL,
CAUSE_INBOUNDBLINDTRANSFER, CAUSE_INBOUNDCONFERENCE,
CAUSE_INBOUNDTRANSFER, CAUSE_INCOMINGCALLBARRED,
CAUSE_INCOMPATABLEDDESTINATION, CAUSE_INTERWORKINGUNSPECIFIED,
CAUSE_INVALIDCALLREFVALUE, CAUSE_INVALIDIECONTENTS,
CAUSE_INVALIDMESSAGEUNSPECIFIED, CAUSE_INVALIDNUMBERFORMAT,
CAUSE_INVALIDTRANSITNETSEL, CAUSE_MANDATORYIEMISSING,
CAUSE_MSGNCOMPATABLEWCS, CAUSE_MSGTYPENCOMPATWCS,
CAUSE_MSGTYPENIMPL, CAUSE_NETOUTOFORDER, CAUSE_NOANSWERFROMUSER,
CAUSE_NOCALLSUSPENDED, CAUSE_NOCIRCAVAIL, CAUSE_NOERROR,
CAUSE_NONSELECTEDUSERCLEARING, CAUSE_NORMALCALLCLEARING,
CAUSE_NORMALUNSPECIFIED, CAUSE_NOROUTETODDESTINATION,
CAUSE_NOROUTETOTRANSITNET, CAUSE_NOUSERRESPONDING,
CAUSE_NUMBERCHANGED, CAUSE_ONLYRDIVEARERCAPAVAIL,
CAUSE_OUTBOUNDCONFERENCE, CAUSE_OUTBOUNDTRANSFER,
CAUSE_OUTOFBANDWIDTH, CAUSE_PROTOCOLERRORUNSPECIFIED, CAUSE_QSIG_PR,
CAUSE_QUALOFSERVNAVAIL, CAUSE_QUIET_CLEAR,
CAUSE_RECOVERYONTIMEREXPIRY, CAUSE_REDIRECTED,
CAUSE_REQCALLIDHASBEENCLEARED, CAUSE_REQCIRCNAVIL,
CAUSE_REQFACILITYNIMPL, CAUSE_REQFACILITYNOTSUBSCRIBED,
CAUSE_RESOURCESNAVAIL, CAUSE_RESPONSETOSTATUSENQUIRY,
CAUSE_SERVNOTAVAILUNSPECIFIED, CAUSE_SERVOPERATIONVIOLATED,
CAUSE_SERVOROPTNAVAILORIMPL, CAUSE_SUBSCRIBERABSENT,
CAUSE_SUSPCALLBUTNOTTHISONE, CAUSE_SWITCHINGEQUIPMENTCONGESTION,
CAUSE_TEMPORARYFAILURE, CAUSE_UNALLOCATEDNUMBER, CAUSE_USERBUSY

インターフェイス javax.telephony.events.Ev から

CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE,
CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT,
CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE,

CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から

CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

表 6-74　CiscoConferenceStartEv のメソッド

インターフェイス	メソッド	説明
javax.telephony.Address	getConferenceControllerAddress()	現在このコール（最初のコール）の会議コントローラとして機能しているアドレスを返します。
javax.telephony.Call	getConferencedCall()	会議に追加されるコールを返します。これは、会議の最初のコールへマージされるコールです。このインターフェイスは、会議に参加しているコールのリストから最初のコールを返します。
javax.telephony.Call[]	getConferencedCalls()	会議に追加されるコールのリストを返します。これらのコールは、会議の最後のコールへマージされるコールです。
javax.telephony.Call	getFinalCall()	会議の終了後にアクティブになっているコールを返します。このコールには、すべてのコールがマージされます。
javax.telephony.TerminalConnection	getHeldConferenceController()	現在このコール（最初のコール）の会議コントローラとして機能している TerminalConnection を返します。これは、当初 HELD 状態だった TerminalConnection です。会議コントローラが監視されていない場合、このメソッドは null を返します。複数コールを結合する場合、このメソッドは、最初の HELD 状態のコントローラを返します。
javax.telephony.TerminalConnection[]	getHeldConferenceControllers()	会議に参加していて、HELD 状態の会議コントローラ端末上のすべての TerminalConnections を返します。

表 6-74　CiscoConferenceStartEv のメソッド （続き）

インターフェイス	メソッド	説明
javax.telephony.Address	getOriginalConferenceController Address()	会議を開始した参加者のアドレスを返します。
javax.telephony.Terminal Connection	getTalkingConferenceController()	現在このコール（最初のコール）の会議コントローラとして機能している TerminalConnection を返します。これは、当初 TALKING 状態だった TerminalConnection です。会議コントローラが監視されていない場合、このメソッドは null を返します。TALKING 状態のコントローラがない場合、このメソッドは null を返します。TALKING 状態のコントローラがない会議にコールを参加させることができます。
String	getControllerTerminalName()	会議が行われるコントローラの端末名を返します。

継承したメソッド

インターフェイス com.cisco.jtapi.extensions.CiscoCallEv から
getCiscoCause, getCiscoFeatureReason

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.CallEv から
getCall

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

「定数フィールド値」（P.F-1）を参照してください。

CiscoConnection

CiscoConnection インターフェイスは、Cisco 固有の機能を追加することによって CallControlConnection インターフェイスを拡張します。アプリケーションは getReason メソッドを使用して、接続が確立された原因を取得できます。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	追加された 2 つの新しいメソッド：getPartyInfo と任意の通話者のドロップ（Drop Any Party）機能の disconnect(CiscoPartyInfopartyInfo

すべてのスーパーインターフェイス

javax.telephony.callcontrol.CallControlConnection, CiscoObjectContainer, javax.telephony.Connection

宣言

public interface CiscoConnection extends javax.telephony.callcontrol.CallControlConnection, CiscoObjectContainer

フィールド

表 6-75　CiscoConnection のフィールド

インターフェイス	フィールド	説明
static int	ADDRESS_SEARCH_SPACE	リダイレクトは、リダイレクト コントローラのアドレスの検索スペースを使用して実行します。
static int	CALLED_ADDRESS_DEFAULT	Cisco JTAPI のデフォルト動作を適用します。
static int	CALLED_ADDRESS_SET_TO_PREFERRED CALLEDPARTY	元の着信アドレスは、preferredOriginalCalledParty フィールドに存在する値に設定します。
static int	CALLED_ADDRESS_SET_TO_REDIRECT_ DESTINATION	着側のアドレスをリダイレクト先にリセットします。
static int	CALLED_ADDRESS_UNCHANGED	リダイレクトの完了後も元の calledAddress は変更されません。
static int	CALLINGADDRESS_SEARCH_SPACE	リダイレクトは、発信元アドレスの検索スペースを使用して実行します。
static int	DEFAULT_SEARCH_SPACE	リダイレクトは、デフォルトの検索スペースを使用して実行します。
static int	REASON_DIRECTCALL	この接続は、直接コールの結果として確立されました。
static int	REASON_FORWARDALL	この接続は、無条件転送の結果として確立されました。

表 6-75　CiscoConnection のフィールド

インターフェイス	フィールド	説明
static int	REASON_FORWARDBUSY	この接続は、使用中での転送の結果として確立されました。
static int	REASON_FORWARDNOANSWER	この接続は、無応答時転送の結果として確立されました。
static int	REASON_OUTBOUND	この接続は、宛先側の接続ではなく発側の接続です。
static int	REASON_REDIRECT	この接続は、リダイレクションの結果として確立されました。
static int	REASON_TRANSFERREDCALL	この接続は、転送の結果として確立されました。
static int	REDIRECT_DROP_ON_FAILURE	このリダイレクト モードが指定されている場合は、宛先が有効であるかどうか、宛先が使用可能であるかどうかを確認せずにリダイレクトが実行されます。
static int	REDIRECT_NORMAL	このリダイレクト モードが指定されている場合は、宛先が有効かつ使用可能な場合にだけリダイレクトが実行されます。

継承したフィールド

インターフェイス javax.telephony.callcontrol.CallControlConnection から
ALERTING, DIALING, DISCONNECTED, ESTABLISHED, FAILED, IDLE, INITIATED, NETWORK_ALERTING, NETWORK_REACHED, OFFERED, OFFERING, QUEUED, UNKNOWN

インターフェイス javax.telephony.Connection から
CONNECTED, INPROGRESS

メソッド

表 6-76　CiscoConnection のメソッド

インターフェイス	メソッド	説明
Boolean	getAddressPI()	接続が作成されるアドレスに関連付けられた Presentation Indicator（PI）を返します。
CiscoConnectionID	getConnectionID()	この CiscoConnection の CiscoConnectionID を返します。
java.lang.String	getDParkPrefixCode()	コールを取得するために DPark DN と一緒にダイヤルする必要があるプレフィクス コードを返します。

表 6-76　CiscoConnection のメソッド （続き）

インターフェイス	メソッド	説明
int	getReason()	この接続が確立された原因を返します。エンドポイントで接続が確立された原因を識別しないと、適切に機能しないアプリケーションもあります。 接続の確立の原因を示す定数は、次のとおりです。 • CiscoConnection.REASON_DIRECTCALL CiscoConnection.REASON_TRANSFERREDCALL • CiscoConnection.REASON_FORWARDNOANSWER • CiscoConnection.REASON_FORWARDBUSY • CiscoConnection.REASON_FORWARDALL • CiscoConnection.REASON_REDIRECT • CiscoConnection.REASON_NORMAL
javax.telephony.TerminalConnection	getRequestController()	接続の現在の要求コントローラを返します。
java.lang.String	park()	このメソッドは、システム パーク DN にコールをパークして、その パーク DN のアドレスを返します。

表 6-76　CiscoConnection のメソッド （続き）

インターフェイス	メソッド	説明
javax.telephony.Connection	redirect(java.lang.String destinationAddress, int mode)	このメソッドは CallControlConnection.redirect() メソッドをオーバーロードします。 **例外** javax.telephony.InvalidStateException, javax.telephony.InvalidPartyException, javax.telephony.MethodNotSupported Exception, javax.telephony.PrivilegeViolationException, javax.telephony.ResourceUnavailable Exception **パラメータ** Mode：パラメータには次の 2 つのいずれかの値を指定できます。 • CiscoConnection.REDIRECT_DROP_ON_FAILURE：このモードが指定されている場合は、宛先が有効であるかどうか、宛先が使用可能であるかどうかを確認せずにリダイレクトが実行されます。宛先が有効でない場合、または使用中の場合は、元のコールがドロップされます。 • CiscoConnection.REDIRECT_NORMAL：このモードが指定されている場合は、宛先が有効であるかどうか、宛先が使用可能であるかどうかを確認した後にだけリダイレクトが実行されます。これは CallControlConnection.redirect() メソッドの動作と同じです。失敗した場合でも、元のコールはドロップされません。

表 6-76　CiscoConnection のメソッド （続き）

インターフェイス	メソッド	説明
javax.telephony.Connection	redirect(java.lang.String destinationAddress, int mode, int callingSearchSpace)	このメソッドは CallControlConnection.redirect() メソッドをオーバーロードします。このメソッドは 2 種類の新しいパラメータ redirectMode および callingSearchSpace を受け付けます。 redirectMode は、実行するリダイレクトの種類を選択します。callingSearchSpace は、発側の検索スペースまたはリダイレクト コントローラの検索スペースのどちらかを使用する実装を指定します。 **パラメータ** mode：次のいずれかです。 • CiscoConnection.REDIRECT_DROP_ON_FAILURE：このモードが指定されている場合は、宛先が有効であるかどうか、宛先が使用可能であるかどうかを確認せずにリダイレクトが実行されます。宛先が有効でない場合、または使用中の場合は、元のコールがドロップされます。 • CiscoConnection.REDIRECT_NORMAL：このモードが指定されている場合は、宛先が有効であるかどうか、宛先が使用可能であるかどうかを確認した後にだけリダイレクトが実行されます。これは CallControlConnection.redirect() メソッドの動作と同じです。失敗した場合でも、元のコールはドロップされません。 callingSearchSpace：次のいずれかです。 • CiscoConnection.DEFAULT_SEARCH_SPACE • CiscoConnection.CALLINGADDRESS_SEARCH_SPACE • CiscoConnection.ADDRESS_SEARCH_SPACE **例外** javax.telephony.InvalidStateException, javax.telephony.InvalidPartyException, javax.telephony.MethodNotSupported Exception, javax.telephony.PrivilegeViolationException, javax.telephony.ResourceUnavailable Exception

表 6-76　CiscoConnection のメソッド （続き）

インターフェイス	メソッド	説明
javax.telephony.Connection	redirect(java.lang.String destinationAddress, int mode, int callingSearchSpace, int calledAddressOption)	このメソッドは CallControlConnection.redirect() メソッドをオーバーロードします。 このメソッドは 3 種類の新しいパラメータ mode、callingSearchSpace、および calledAddressOption を受け付けます。redirectMode は、実行するリダイレクトの種類を選択します。callingSearchSpace は、発側の検索スペースまたはリダイレクト コントローラの検索スペースのどちらかを使用する実装を指定します。calledAddressOption パラメータは、元の着信フィールドをリセットするかどうかを制御します。 **パラメータ** mode：次のいずれかです。 • CiscoConnection.REDIRECT_DROP_ON_FAILURE • CiscoConnection.REDIRECT_NORMAL callingSearchSpace：次のいずれかです。 • CiscoConnection.DEFAULT_SEARCH_SPACE • CiscoConnection.CALLINGADDRESS_SEARCH_SPACE • CiscoConnection.ADDRESS_SEARCH_SPACE calledAddressOption：次のいずれかです。 • CiscoConnection.CALLED_ADDRESS_DEFAULT • CiscoConnection.CALLED_ADDRESS_UNCHANGED • CiscoConnection.CALLED_ADDRESS_SET_TO_REDIRECT_DESTINATION **例外** javax.telephony.InvalidStateException, javax.telephony.InvalidPartyException, javax.telephony.MethodNotSupported Exception, javax.telephony.PrivilegeViolationException, javax.telephony.ResourceUnavailable Exception

表 6-76　CiscoConnection のメソッド（続き）

インターフェイス	メソッド	説明
javax.telephony.Connection	redirect(java.lang.String destinationAddress, int mode, int callingSearchSpace, int calledAddressOption, java.lang.String preferredOriginalCalledParty, java.lang.String facCode, java.lang.String cmcCode)	このメソッドは CallControlConnection.redirect() メソッドをオーバーロードします。このメソッドは 3 種類の新しいパラメータ mode、callingSearchSpace、および calledAddressOption を受け付けます。 redirectMode は、実行するリダイレクトの種類を選択します。callingSearchSpace は、発側の検索スペースまたはリダイレクト コントローラの検索スペースのどちらかを使用する実装を指定します。calledAddressOption パラメータは、元の着信フィールドをリセットするかどうかを制御します。 FAC コードおよび CMC コードが不明であるか無効な場合、コールが提供されない可能性や、platformException に次のエラー コードのいずれかが含まれる可能性があります。 • CiscoJTAPIException.CTIERR_FAC_CMC_REASON_FAC_NEEDED • CiscoJTAPIException.CTIERR_FAC_CMC_REASON_CMC_NEEDED • CiscoJTAPIException.CTIERR_FAC_CMC_REASON_FAC_CMC_NEEDED • CiscoJTAPIException.CTIERR_FAC_CMC_REASON_FAC_INVALID • CiscoJTAPIException.CTIERR_FAC_CMC_REASON_CMC_INVALID **パラメータ** mode：次のいずれかです。 • CiscoConnection.REDIRECT_DROP_ON_FAILURE • CiscoConnection.REDIRECT_NORMAL callingSearchSpace：次のいずれかです。 • CiscoConnection.DEFAULT_SEARCH_SPACE • CiscoConnection.CALLINGADDRESS_SEARCH_SPACE • CiscoConnection.ADDRESS_SEARCH_SPACE

表 6-76　CiscoConnection のメソッド（続き）

インターフェイス	メソッド	説明
		• preferredOriginalCalledParty：destinationAddress に対してコールが提供されたときに originalCalledParty フィールドになる DN。このフィールド * を使用する必要がある場合、アプリケーションは calledAddressOption に CALLED_ADDRESS_SET_TO_PREFERREDCALLEDPARTY を設定する必要があります。アプリケーションがこのフィールドを指定していない場合、デフォルト値の null を渡す必要があります。 calledAddressOption：次のいずれかです。 • CiscoConnection.CALLED_ADDRESS_DEFAULT • CiscoConnection.CALLED_ADDRESS_UNCHANGED • CiscoConnection.CALLED_ADDRESS_SET_TO_REDIRECT_DESTINATION preferredOriginalCalledParty：destinationAddress に対してコールが提供されたときに originalCalledParty フィールドになる DN。このフィールド * を使用する必要がある場合、アプリケーションは calledAddressOption に CALLED_ADDRESS_SET_TO_PREFERREDCALLEDPARTY を設定する必要があります。アプリケーションがこのフィールドを指定していない場合、デフォルト値の null を渡す必要があります。 facCode：destinationAddress がコールをオファーするのに Forced Authorization Code（FAC）を要求する場合に必要です。このパラメータで FAC を渡します。destinationAddress が FAC コードを要求しない場合、デフォルト値の null を渡します。 cmcCode：destinationAddress がコールをオファーするのに Client Matter Code（CMC）を要求する場合に必要です。このパラメータで CMC を渡します。destinationAddress が CMC コードを要求しない場合、デフォルト値の null を渡します。

表 6-76　CiscoConnection のメソッド （続き）

インターフェイス	メソッド	説明
javax.telephony.Connection	redirect(java.lang.String destinationAddress, int mode, int callingSearchSpace, int calledAddressOption, java.lang.String preferredOriginalCalledParty, java.lang.String facCode, java.lang.String cmcCode, int featurePriority)	このメソッドは CallControlConnection.redirect() メソッドをオーバーロードします。これは新しいパラメータ、featurePriority をオーバーロードします。コール優先度を設定します。featurePriority パラメータは、次のいずれかになります。 • CiscoCall.FEATUREPRIORITY_NORMAL • CiscoCall.FEATUREPRIORITY_URGENT • CiscoCall.FEATUREPRIORITY_EMERGENCY **戻り値** Connection **例外** javax.telephony.InvalidStateException, javax.telephony.InvalidPartyException, javax.telephony.MethodNotSupported Exception, javax.telephony.PrivilegeViolationException, javax.telephony.ResourceUnavailable Exception

表 6-76　CiscoConnection のメソッド （続き）

インターフェイス	メソッド	説明
javax.telephony.Connection	redirect(java.lang.String destinationAddress, int mode, int callingSearchSpace, java.lang.String preferredOriginalCalledParty)	このメソッドは CallControlConnection.redirect() メソッドをオーバーロードします。このメソッドは 3 種類の新しいパラメータ mode、callingSearchSpace、および preferredOriginalCalledParty を受け付けます。 redirectMode は、実行するリダイレクトの種類を選択します。callingSearchSpace は、発側の検索スペースまたはリダイレクト コントローラの検索スペースのどちらかを使用する実装を指定します。 **パラメータ** mode：次のいずれかです。 • CiscoConnection.REDIRECT_DROP_ON_FAILURE • CiscoConnection.REDIRECT_NORMAL callingSearchSpace：次のいずれかです。 • CiscoConnection.DEFAULT_SEARCH_SPACE • CiscoConnection.CALLINGADDRESS_SEARCH_SPACE • CiscoConnection.ADDRESS_SEARCH_SPACE preferredOriginalCalledParty：destinationAddress に対してコールが提供されたときに originalCalledParty フィールドになる DN。 **例外** javax.telephony.InvalidStateException, javax.telephony.InvalidPartyException, javax.telephony.MethodNotSupported Exception, javax.telephony.PrivilegeViolationException, javax.telephony.ResourceUnavailable Exception
void	setRequestController(javax.telephony.TerminalConnection tc)	このインターフェイスは、要求側の TerminalConnection に対して提供されます。
com.cisco.jtapi.extensions.CiscoPartyInfo[]	getPartyInfo()	参加者のリストを返します。

表 6-76　CiscoConnection のメソッド （続き）

インターフェイス	メソッド	説明
java.lang.void	disconnect(CiscoPartyInfoparty Info)	渡されたパラメータ値と CiscoPartyInfo が一致している参加者を接続解除し、一致しない場合は例外をスローします。 **例外** PrivilegeViolationException, InvalidStateException

継承したメソッド

インターフェイス javax.telephony.callcontrol.CallControlConnection から
accept, addToAddress, getCallControlState, park, redirect, reject

インターフェイス javax.telephony.Connection から
disconnect, getAddress, getCall, getCapabilities, getConnectionCapabilities, getState, getTerminalConnections

インターフェイス com.cisco.jtapi.extensions.CiscoObjectContainer から
getObject, setObject

資料

なし

CiscoConnectionID

CiscoConnectionID オブジェクトは、各接続に関連付けられる一意のオブジェクトです。A アプリケーションでは、オブジェクト自体または intValue() メソッドで返されたオブジェクトの整数表現を使用できます。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

CiscoObjectContainer

宣言

public interface CiscoConnectionID extends CiscoObjectContainer

フィールド

なし

メソッド

表 6-77　CiscoConnectionID のメソッド

インターフェイス	メソッド	説明
CiscoConnection	getConnection()	CiscoConnectionID の CiscoConnection を返します。
Int	intValue()	このオブジェクトの整数表現を返します。

継承したメソッド

インターフェイス com.cisco.jtapi.extensions.CiscoObjectContainer から
getObject, setObject

関連資料

なし

CiscoConsultCall

CiscoConsultCall インターフェイスは CiscoCall インターフェイスを拡張し、コンサルト転送またはコンサルト会議の一部として作成されたコールのプロパティへのアクセスを可能にします。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

javax.telephony.Call, javax.telephony.callcontrol.CallControlCall, CiscoCall, CiscoObjectContainer

宣言

public interface CiscoConsultCall extends CiscoCall

フィールド

なし

継承したフィールド

インターフェイス com.cisco.jtapi.extensions.CiscoCall から
CALLSECURITY_AUTHENTICATED, CALLSECURITY_ENCRYPTED, CALLSECURITY_NOTAUTHENTICATED, CALLSECURITY_UNKNOWN, FEATUREPRIORITY_EMERGENCY, FEATUREPRIORITY_NORMAL, FEATUREPRIORITY_URGENT, PLAYTONE_BOTHLOCALANDREMOTE, PLAYTONE_LOCALONLY, PLAYTONE_NOLOCAL_OR_REMOTE, PLAYTONE_REMOTEONLY, SILENT_MONITOR

インターフェイス javax.telephony.Call から
ACTIVE, IDLE, INVALID

メソッド

表 6-78　CiscoConsultCall のメソッド

インターフェイス	メソッド	説明
javax.telephony.Terminal Connection	getConsultingTerminalConnection()	この CiscoConsultCall の作成に使用されたコンサルティング TerminalConnection を返します。このコールがコンサルト転送またはコンサルト会議の一部として作成されていた場合、元のコールのコンサルテーションを実行するのに使用された TerminalConnection が getConsultingTerminalConnection メソッドによって返されます。このメソッドでは、ConsultCall と最初のコールを相互に関連付けることができます。最初のコールは、自身に関連付けられている ConsultCall を識別するためのメソッドを持っていません。戻り値：このコールがコンサルテーションの結果作成されたものでない場合は null、このコールがコンサルテーションの結果作成されたものである場合は、元のコールのコンサルティング TerminalConnection。

表 6-78　CiscoConsultCall のメソッド （続き）

インターフェイス	メソッド	説明
javax.telephony.Connection[]	consultWithoutMedia (javax.telephony.TerminalConnection tc, java.lang.String dialedDigits)	メディアを設定することなく、コンサルト コールを開始する機能をアプリケーションに提供します。このインターフェイスは、アプリケーションがコンサルト コールを作成し、コンサルト コールのメディアが確立する前に転送が完了する場合に起動できます。 コンサルト コールが応答され、コンサルト コールの設定中にアプリケーションが転送を完了した場合、Cisco Unified Communication Manager は誤りのある競合状態で稼動する可能性があります。 この問題を回避するために、コンサルト コールのメディアの設定を待機しないアプリケーションがこのメソッドを使用して、コンサルト コールを設定することができます。 CallEvent の観点からすると、このメソッドの動作は CallControlCall.consult(TerminalConnection tc, String dialedDigits) に類似しています。 このメソッドは、この Call と他のアクティブ Call の間のコンサルテーションを作成します。ただし、メディアは確立されません。このコンサルト コールを転送することは可能ですが、会議に追加できません。Cisco JTAPI では、CallControlCall.setConferenceEnable() でこのメソッドをサポートしません。Cisco JTAPI では、CallControlCall.setTransferEnable() でだけこのメソッドをサポートします。 **例外** javax.telephony.InvalidStateException javax.telephony.InvalidArgumentException javax.telephony.MethodNotSupported Exception javax.telephony.ResourceUnavailableException javax.telephony.PrivilegeViolationException javax.telephony.InvalidPartyException

継承したメソッド

インターフェイス com.cisco.jtapi.extensions.CiscoCall から

conference, connect, getCalledAddressPI, getCalledPartyInfo, getCallID, getCallingAddressPI, getCallSecurityStatus, getConferenceChain, getCurrentCalledAddress, getCurrentCalledAddressPI, getCurrentCalledDisplayNamePI, getCurrentCalledPartyDisplayName, getCurrentCalledPartyInfo, getCurrentCalledPartyUnicodeDisplayName, getCurrentCalledPartyUnicodeDisplayNamelocale, getCurrentCallingAddress, getCurrentCallingAddressPI, getCurrentCallingDisplayNamePI,

getCurrentCallingPartyDisplayName, getCurrentCallingPartyInfo, getCurrentCallingPartyUnicodeDisplayName, getCurrentCallingPartyUnicodeDisplayNamelocale, getGlobalizedCallingParty, getLastRedirectedPartyInfo, getLastRedirectingAddressPI, getLastRedirectingPartyInfo, getModifiedCalledAddress, getModifiedCallingAddress, startMonitor, startMonitor, transfer

インターフェイス javax.telephony.callcontrol.CallControlCall から
addParty, conference, consult, consult, drop, getCalledAddress, getCallingAddress, getCallingTerminal, getConferenceController, getConferenceEnable, getLastRedirectedAddress, getTransferController, getTransferEnable, offHook, setConferenceController, setConferenceEnable, setTransferController, setTransferEnable, transfer, transfer

インターフェイス javax.telephony.Call から
addObserver, connect, getCallCapabilities, getCapabilities, getConnections, getObservers, getProvider, getState, removeObserver

インターフェイス com.cisco.jtapi.extensions.CiscoObjectContainer から
getObject, setObject

関連資料

詳細については、CiscoCall を参照してください。

CiscoConsultCallActiveEv

CiscoConsultCallActiveEv イベント インターフェイスは JTAPI の CallActiveEv イベントを拡張します。このイベントは、コール オブジェクトの状態が Call.ACTIVE に変わり、（手動またはプログラムによる）コンサルト転送動作またはコンサルト会議動作の結果としてコールが発信されたことを示します。アプリケーションは CiscoConsultCall.getConsultingTerminalConnection メソッドを使用して、最初の（コンサルト）コールのコンサルティング TerminalConnection を取得できます。

このイベントは CallObserver インターフェイスによってアプリケーションに報告されます。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

javax.telephony.events.CallActiveEv, javax.telephony.events.CallEv, CiscoCallEv, CiscoEv, javax.telephony.events.Ev

宣言

public interface CiscoConsultCallActiveEv extends CiscoCallEv, javax.telephony.events.CallActiveEv

フィールド

なし

継承したフィールド

インターフェイス com.cisco.jtapi.extensions.CiscoCallEv から
CAUSE_ACCESSINFORMATIONDISCARDED, CAUSE_BARGE,
CAUSE_BCBPRESENTLYAVAIL, CAUSE_BCNAUTHORIZED, CAUSE_BEARERCAPNIMPL,
CAUSE_CALLBEINGDELIVERED, CAUSE_CALLIDINUSE,
CAUSE_CALLMANAGER_FAILURE, CAUSE_CALLREJECTED, CAUSE_CALLSPLIT,
CAUSE_CHANTYPENIMPL, CAUSE_CHANUNACCEPTABLE,
CAUSE_CTICCMSIP400BADREQUEST, CAUSE_CTICCMSIP401UNAUTHORIZED,
CAUSE_CTICCMSIP402PAYMENTREQUIRED, CAUSE_CTICCMSIP403FORBIDDEN,
CAUSE_CTICCMSIP404NOTFOUND, CAUSE_CTICCMSIP405METHODNOTALLOWED,
CAUSE_CTICCMSIP406NOTACCEPTABLE,
CAUSE_CTICCMSIP407PROXYAUTHENTICATIONREQUIRED,
CAUSE_CTICCMSIP408REQUESTTIMEOUT, CAUSE_CTICCMSIP410GONE,
CAUSE_CTICCMSIP411LENGTHREQUIRED,
CAUSE_CTICCMSIP413REQUESTENTITYTOOLONG,
CAUSE_CTICCMSIP414REQUESTURITOOLONG,
CAUSE_CTICCMSIP415UNSUPPORTEDMEDIATYPE,
CAUSE_CTICCMSIP416UNSUPPORTEDURISCHEME,
CAUSE_CTICCMSIP420BADEXTENSION, CAUSE_CTICCMSIP421EXTENSTIONREQUIRED,
CAUSE_CTICCMSIP423INTERVALTOOBRIEF,
CAUSE_CTICCMSIP480TEMPORARILYUNAVAILABLE,
CAUSE_CTICCMSIP481CALLLEGDOESNOTEXIST, CAUSE_CTICCMSIP482LOOPDETECTED,
CAUSE_CTICCMSIP483TOOMANYHOOPS, CAUSE_CTICCMSIP484ADDRESSINCOMPLETE,
CAUSE_CTICCMSIP485AMBIGUOUS, CAUSE_CTICCMSIP486BUSYHERE,
CAUSE_CTICCMSIP487REQUESTTERMINATED,
CAUSE_CTICCMSIP488NOTACCEPTABLEHERE, CAUSE_CTICCMSIP491REQUESTPENDING,
CAUSE_CTICCMSIP493UNDECIPHERABLE,
CAUSE_CTICCMSIP500SERVERINTERNALERROR,
CAUSE_CTICCMSIP501NOTIMPLEMENTED, CAUSE_CTICCMSIP502BADGATEWAY,
CAUSE_CTICCMSIP503SERVICEUNAVAILABLE, CAUSE_CTICCMSIP504SERVERTIMEOUT,
CAUSE_CTICCMSIP505SIPVERSIONNOTSUPPORTED,
CAUSE_CTICCMSIP513MESSAGETOOLARGE, CAUSE_CTICCMSIP600BUSYEVERYWHERE,
CAUSE_CTICCMSIP603DECLINE, CAUSE_CTICCMSIP604DOESNOTEXISTANYWHERE,
CAUSE_CTICCMSIP606NOTACCEPTABLE, CAUSE_CTICONFERENCEFULL,
CAUSE_CTIDEVICENOTPREEMPTABLE, CAUSE_CTIDROPCONFEREE,
CAUSE_CTIMANAGER_FAILURE, CAUSE_CTIPRECEDENCECALLBLOCKED,
CAUSE_CTIPRECEDENCELEVELEXCEEDED,
CAUSE_CTIPRECEDENCEOUTOFBANDWIDTH, CAUSE_CTIPREEMPTFORREUSE,
CAUSE_CTIPREEMPTNOREUSE, CAUSE_DESTINATIONOUTOFORDER,
CAUSE_DESTNUMMISSANDDCNOTSUB, CAUSE_DPARK, CAUSE_DPARK_REMINDER,
CAUSE_DPARK_UNPARK, CAUSE_EXCHANGEROUTINGERROR, CAUSE_FAC_CMC,
CAUSE_FACILITYREJECTED, CAUSE_IDENTIFIEDCHANDOESNOTEXIST, CAUSE_IENIMPL,
CAUSE_INBOUNDBLINDTRANSFER, CAUSE_INBOUNDCONFERENCE,
CAUSE_INBOUNDTRANSFER, CAUSE_INCOMINGCALLBARRED,
CAUSE_INCOMPATABLEDDESTINATION, CAUSE_INTERWORKINGUNSPECIFIED,
CAUSE_INVALIDCALLREFVALUE, CAUSE_INVALIDIECONTENTS,

CAUSE_INVALIDMESSAGEUNSPECIFIED, CAUSE_INVALIDNUMBERFORMAT, CAUSE_INVALIDTRANSITNETSEL, CAUSE_MANDATORYIEMISSING, CAUSE_MSGNCOMPATABLEWCS, CAUSE_MSGTYPENCOMPATWCS, CAUSE_MSGTYPENIMPL, CAUSE_NETOUTOFORDER, CAUSE_NOANSWERFROMUSER, CAUSE_NOCALLSUSPENDED, CAUSE_NOCIRCAVAIL, CAUSE_NOERROR, CAUSE_NONSELECTEDUSERCLEARING, CAUSE_NORMALCALLCLEARING, CAUSE_NORMALUNSPECIFIED, CAUSE_NOROUTETODDESTINATION, CAUSE_NOROUTETOTRANSITNET, CAUSE_NOUSERRESPONDING, CAUSE_NUMBERCHANGED, CAUSE_ONLYRDIVEARERCAPAVAIL, CAUSE_OUTBOUNDCONFERENCE, CAUSE_OUTBOUNDTRANSFER, CAUSE_OUTOFBANDWIDTH, CAUSE_PROTOCOLERRORUNSPECIFIED, CAUSE_QSIG_PR, CAUSE_QUALOFSERVNAVAIL, CAUSE_QUIET_CLEAR, CAUSE_RECOVERYONTIMEREXPIRY, CAUSE_REDIRECTED, CAUSE_REQCALLIDHASBEENCLEARED, CAUSE_REQCIRCNAVIL, CAUSE_REQFACILITYNIMPL, CAUSE_REQFACILITYNOTSUBSCRIBED, CAUSE_RESOURCESNAVAIL, CAUSE_RESPONSETOSTATUSENQUIRY, CAUSE_SERVNOTAVAILUNSPECIFIED, CAUSE_SERVOPERATIONVIOLATED, CAUSE_SERVOROPTNAVAILORIMPL, CAUSE_SUBSCRIBERABSENT, CAUSE_SUSPCALLBUTNOTTHISONE, CAUSE_SWITCHINGEQUIPMENTCONGESTION, CAUSE_TEMPORARYFAILURE, CAUSE_UNALLOCATEDNUMBER,CAUSE_USERBUSY

インターフェイス javax.telephony.events.Ev から

CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から

CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から

CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

javax.telephony.TerminalConnectiongetHeldTerminalConnection() 推奨されません。CiscoConsultCall.getConsultingTerminalConnection() に置き換えられました。

表 6-79　CiscoConsultCallActiveEv のメソッド

インターフェイス	メソッド	説明
javax.telephony.TerminalConnection	getHeldTerminalConnection()	**推奨されないメソッドです。** CiscoConsultCall.getConsultingTerminal Connection() に置き換えられました。 この CiscoConsultCall の作成に使用されたコンサルティング TerminalConnection を返します。このメソッドを使用して、コンサルト コールと最初のコールを相互に関連付けることができます。最初のコールは、自身に関連付けられているコンサルト コールを識別するためのメソッドを持っていません。戻り値：このイベントに関連付けられているコールを作成したコールのコンサルティング TerminalConnection。

継承したメソッド

インターフェイス com.cisco.jtapi.extensions.CiscoCallEv から
getCiscoCause, getCiscoFeatureReason

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.CallEv から
getCall

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.CallEv から
getCall

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

詳細については、Call、CallObserver、CallActiveEv と「定数フィールド値」（P.F-1）を参照してください。

CiscoEv

このコードの JTAPI javax.telephony.events.Ev インターフェイスを拡張する CiscoEv インターフェイスは、Cisco によって拡張されたすべての JTAPI イベントの基本インターフェイスになります。このパッケージのイベントはすべて、直接的または間接的にこのインターフェイスを拡張します。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

javax.telephony.events.Ev

サブインターフェイス

CiscoAddrActivatedEv, CiscoAddrActivatedOnTerminalEv, CiscoAddrAddedToTerminalEv, CiscoAddrAutoAcceptStatusChangedEv, CiscoAddrCreatedEv, CiscoAddrEv, CiscoAddrInServiceEv, CiscoAddrIntercomInfoChangedEv, CiscoAddrIntercomInfoRestorationFailedEv, CiscoAddrOutOfServiceEv, CiscoAddrRecordingConfigChangedEv, CiscoAddrRemovedEv, CiscoAddrRemovedFromTerminalEv, CiscoAddrRestrictedEv, CiscoAddrRestrictedOnTerminalEv, CiscoCallChangedEv, CiscoCallEv, CiscoCallSecurityStatusChangedEv, CiscoConferenceChainAddedEv, CiscoConferenceChainRemovedEv, CiscoConferenceEndEv, CiscoConferenceStartEv, CiscoConsultCallActiveEv, CiscoMediaOpenLogicalChannelEv, CiscoOutOfServiceEv, CiscoProvCallParkEv, CiscoProvEv, CiscoProvFeatureEv, CiscoProvTerminalCapabilityChangedEv, CiscoRestrictedEv, CiscoRTPInputKeyEv, CiscoRTPInputStartedEv, CiscoRTPInputStoppedEv, CiscoRTPOutputKeyEv, CiscoRTPOutputStartedEv, CiscoRTPOutputStoppedEv, CiscoTermActivatedEv, CiscoTermButtonPressedEv, CiscoTermCreatedEv, CiscoTermDataEv, CiscoTermDeviceStateActiveEv, CiscoTermDeviceStateAlertingEv, CiscoTermDeviceStateHeldEv, CiscoTermDeviceStateIdleEv, CiscoTermDeviceStateWhisperEv, CiscoTermDNDOptionChangedEv, CiscoTermDNDStatusChangedEv, CiscoTermEv, CiscoTermInServiceEv, CiscoTermOutOfServiceEv, CiscoTermRegistrationFailedEv, CiscoTermRemovedEv, CiscoTermRestrictedEv, CiscoTermSnapshotCompletedEv, CiscoTermSnapshotEv, CiscoToneChangedEv, CiscoTransferEndEv, CiscoTransferStartEv

宣言

public interface CiscoEv extends javax.telephony.events.Ev

フィールド

なし

継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

なし

継承したメソッド

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

詳細については、Ev を参照してください。

CiscoFeatureReason

CiscoFeatureReason インターフェイスは、配信される各イベントに関連付けられた機能原因を指定します。

インターフェイス履歴

Cisco Unified Communications Manager リリース	説明
7.1(1 および 2)	パーク モニタリングおよび Assisted DPark 機能のための新しい原因、FORWARD_NO_RETRIEVE が追加されました。

宣言

public interface CiscoFeatureReason

フィールド

表 6-80　CiscoFeatureReason のフィールド

インターフェイス	フィールド	説明
static int	REASON_BARGE	イベントに関する理由が割り込み機能であることを示します。
static int	REASON_BLINDTRANSFER	理由がシングル ステップ転送であることを示します。
static int	REASON_CALLPICKUP	イベントに関する理由がピックアップであることを示します。
static int	REASON_CCM_REDIRECTION	イベントに関する理由が SIP 3xx 機能であることを示します。
static int	REASON_CLICK_TO_CONFERENCE	クリック ツー会議機能を使用して接続が作成されたか削除されたことを示します。
static int	REASON_CONFERENCE	イベントに関する理由が会議であることを示します。
static int	REASON_DPARK_CALLPARK	イベントに関する理由が DPARK 機能であることを示します。
static int	REASON_DPARK_REVERSION	イベントに関する理由が DPARK の復帰であることを示します。
static int	REASON_DPARK_UNPARK	イベントに関する理由が DPARK のパーク解除であることを示します。
static int	REASON_FAC_CMC	イベントに関する理由が FAC 機能、CMC 機能であることを示します。
static int	REASON_FORWARDALL	イベントに関する理由が転送であることを示します。
static int	REASON_FORWARDBUSY	イベントに関する理由が話中転送であることを示します。
static int	REASON_FORWARDNOANSWER	イベントに関する理由が無応答時転送であることを示します。
static int	REASON_FORWARD_NO_RETRIEVE	イベントに関する理由が取得転送なしであることを示します。
static int	REASON_IMMDIVERT	イベントに関する理由が即時転送であることを示します。
static int	REASON_MOBILITY	イベントに関する理由が Mobility Manager 機能であることを示します。
static int	REASON_MOBILITY_CELLPICKUP	イベントに関する理由が Mobility Manager 機能であることを示します。
static int	REASON_MOBILITY_FOLLOWME	イベントに関する理由が Mobility Manager 機能であることを示します。

表 6-80　CiscoFeatureReason のフィールド

インターフェイス	フィールド	説明
static int	REASON_MOBILITY_HANDIN	イベントに関する理由が Mobility Manager 機能であることを示します。
static int	REASON_MOBILITY_HANDOUT	イベントに関する理由が Mobility Manager 機能であることを示します。
static int	REASON_MOBILITY_IVR	イベントに関する理由が Mobility Manager 機能であることを示します。
static int	REASON_NORMAL	イベントに関する理由が通常であることを示します。
static int	REASON_PARK	イベントに関する理由がパーク機能であることを示します。
static int	REASON_PARKREMAINDER	イベントに関する理由がパーク リマインダであることを示します。
static int	REASON_QSIG_PR	イベントに関する理由が QSIG パス置換であることを示します。
static int	REASON_REDIRECT	イベントに関する理由がリダイレクトであることを示します。
static int	REASON_REFER	Cisco Unified Communications Manager で参照が実行されるために送信されるイベントに返されます。
static int	REASON_REPLACE	REASON_REPLACE：この理由は Cisco Unified Communications Manager で置換機能が実行されるために送信されるイベントに返されます。
static int	REASON_SILENTMONITORING	イベントに関する理由がサイレント モニタリングであることを示します。
static int	REASON_TRANSFER	イベントに関する理由が転送であることを示します。
static int	REASON_UNPARK	イベントに関する理由がパーク解除であることを示します。

関連資料

詳細については、「定数フィールド値」（P.F-1）を参照してください。

CiscoIntercomAddress

CiscoIntercomAddress インターフェイスは、インターコム アドレス用に Unified CM 固有の機能を追加することによって CiscoAddress インターフェイスを拡張します。このインターフェイスを使用すると、アプリケーションからインターコム コールを開始したり、他のインターコム固有の機能を利用できます。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

javax.telephony.Address, CiscoAddress, CiscoObjectContainer

宣言

public interface CiscoIntercomAddress extends CiscoAddress

フィールド

なし

継承したフィールド

インターフェイス com.cisco.jtapi.extensions.CiscoAddress から
APPLICATION_CONTROLLED_RECORDING, AUTO_RECORDING, AUTOACCEPT_OFF, AUTOACCEPT_ON, AUTOANSWER_OFF, AUTOANSWER_UNKNOWN, AUTOANSWER_WITHHEADSET, AUTOANSWER_WITHSPEAKERSET, EXTERNAL, EXTERNAL_UNKNOWN, IN_SERVICE, INTERNAL, MONITORING_TARGET, NO_RECORDING, OUT_OF_SERVICE, RINGER_DEFAULT, RINGER_DISABLE, RINGER_ENABLE, UNKNOWN

メソッド

表 6-81　CiscoIntercomAddress のメソッド

インターフェイス	メソッド	説明
void	setIntercomTarget(java.lang.String targetDNjava.lang.String targetAsciiLabel, java.lang.String targetUnicodeLabel)	電話機のインターコム回線の横に表示されるインターコム ターゲット DN、インターコム ターゲット ラベルおよびインターコム ターゲット Unicode ラベルを設定します。電話機に Unicode ラベル機能がある場合には Unicode ラベルが表示され、そうでない場合は、ASCII ターゲット ラベルが表示されます。 **例外** javax.telephony.InvalidPartyException はターゲット DN が無効であることを意味します。 javax.telephony.InvalidStateException は、アドレス、端末、またはプロバイダーがイン サービスではないことを意味します。 **パラメータ** • targetDN：インターコム コールの宛先 DN • targetAsciiLabel：電話機ターゲットのインターコム回線の横に表示される ASCII 表示ラベル • UnicodeLabel：電話機に表示される Unicode 表示ラベル
Boolean	isIntercomTargetSet()	アプリケーションが現在の値をオーバーライドした場合は true を返し、現在の値がデータベースで設定されたデフォルト値と一致する場合は false を返します。
void	resetIntercomTarget()	インターコム ターゲット DN、インターコム ターゲット ラベルおよびインターコム ターゲット Unicode ラベルをデフォルト値にリセットします。 **例外** javax.telephony.InvalidPartyException, javax.telephony.InvalidStateException
java.lang.String	getIntercomTargetNumber()	アプリケーションで設定される現在のインターコム ターゲット DN を返します。アプリケーションがインターコム ターゲット DN を設定していない場合、このインターフェイスは Cisco Unified CM Administration で設定されるデフォルトのインターコム ターゲット DN を返します。戻り値：インターコム ターゲットの DN 番号（文字列）。

表 6-81　CiscoIntercomAddress のメソッド （続き）

インターフェイス	メソッド	説明
java.lang.String	getIntercomTargetAsciiLabel()	アプリケーションで設定される現在のインターコム ターゲット ラベルを返します。アプリケーションがインターコム ターゲット ラベルを設定していない場合、このインターフェイスは Cisco Unified CM Administration で設定されるデフォルトのインターコム ターゲット ラベルを返します。戻り値：インターコム ターゲットのラベル文字列。
java.lang.String	getIntercomTargetUnicodeLabel()	アプリケーションで設定される現在のインターコム ターゲット Unicode ラベルを返します。アプリケーションが Unicode ラベルを設定していない場合、このインターフェイスは Cisco Unified CM Administration で設定されるデフォルトのインターコム ターゲット Unicode ラベルを返します。戻り値：インターコム Unicode ターゲットのラベル文字列。
java.lang.String	getDefaultIntercomTargetNumber()	Cisco Unified CM Administration から設定されるデフォルトのインターコム ターゲット DN を返します。戻り値：デフォルトのインターコム ターゲットの DN 番号（文字列）。
java.lang.String	getDefaultIntercomTargetAsciiLabel()	Cisco Unified CM Administration から設定されるデフォルトのインターコム ターゲット ラベルを返します。戻り値：デフォルトのインターコム ターゲットのラベル文字列。
java.lang.String	getDefaultIntercomTargetUnicodeLabel()	Cisco Unified CM Administration から設定されるデフォルトのインターコム ターゲット ラベルを返します。戻り値：デフォルトの Unicode インターコム ターゲットのラベル文字列。

表 6-81　CiscoIntercomAddress のメソッド （続き）

インターフェイス	メソッド	説明
javax.telephony.Connection[]	connectIntercom(javax.telephony.Terminal terminal, java.lang.String targetNumber)	発信側インターコム アドレスから着信側インターコム アドレスにインターコム コールをかけます。戻り値：発側と着側のインターコム アドレスの接続リスト。 **例外** javax.telephony.InvalidPartyException：ターゲット DN は有効な番号ではありません。 javax.telephony.InvalidArgumentException：アドレスが CiscoIntercomAddress でないか、または端末が Terminal ではありません。 javax.telephony.InvalidStateException ：アドレス、端末、またはプロバイダーがイン サービスではありません。 javax.telephony.ResourceUnavailableException：オペレーションを完了するためのリソースを利用できません。 javax.telephony.PrivilegeViolationException：アプリケーションにこのオペレーションを実行するための権限がありません。

継承したメソッド

インターフェイス com.cisco.jtapi.extensions.CiscoAddress から
clearCallConnections, getAddressCallInfo, getAutoAcceptStatus, getAutoAnswerStatus, getInServiceAddrTerminals, getPartition, getRecordingConfig, getRegistrationState, getRestrictedAddrTerminals, getState, getType, isRestricted, setAutoAcceptStatus, setMessageWaiting, setRingerStatus

インターフェイス javax.telephony.Address から
addCallObserver, addObserver, getAddressCapabilities, getCallObservers, getCapabilities, getConnections, getName, getObservers, getProvider, getTerminals, removeCallObserver, removeObserver

インターフェイス com.cisco.jtapi.extensions.CiscoObjectContainer から
getObject, setObject

関連資料

詳細については、CiscoAddress を参照してください。

CiscoJtapiException

CiscoJtapiException インターフェイスは CTI 要求を返すエラー コードを定義します。このインターフェイスを実装するために、Cisco JTAPI は すべての JTAPI 例外を拡張しています。例外を CiscoJtapiException にキャストし、メソッド getErrorCode() を呼び出すことによってエラー コードを取得できます。

たとえば、「e」がアプリケーションで検出された例外の場合は、次のコードでその例外が CiscoJtapiException のインスタンスかどうかを確かめます。

```
try {
   // some code here
 }
catch ( Exception e ) {
   if( e instanceof CiscoJtapiException){
     CiscoJtapiException ce = com.cisco.cti.client.CTIFAILURE.(CiscoJtapiException) e
     int errorCode = com.cisco.cti.client.CTIFAILURE.ce.getErrorCode()   //returns the
ErrorCode.
   }
 }
```

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.0	このインターフェイスが追加されました。
7.1(1 および 2)	論理パーティション設定機能の新しいエラー コード、CiscoJtapiException.CTIERR_REDIRECT_CALL_PARTITIONING_POLICY と CiscoJtapiException.CTIERR_FEATURE_NOT_AVAILABLE が追加されました。

宣言

public interface CiscoJtapiException

フィールド

表 6-82　CiscoJtapiException のフィールド

インターフェイス	フィールド	説明
static int	ASSOCIATED_LINE_NOT_OPEN	このエラーは、オープンされていない回線上で要求が発行されたことを示します。
static int	CALL_ALREADY_EXISTS	このエラーは、回線上にすでに別のコールが存在していることを示します。
static int	CALL_DROPPED	機能（保留、保留解除、転送、または会議）要求の後、要求が完了する前にコールがドロップされました。

表 6-82　CiscoJtapiException のフィールド （続き）

インターフェイス	フィールド	説明
static int	CALLHANDLE_NOTINCOMINGCALL	このエラーは、存在しないか、または正しい状態にないコールに応答しようとしたことを示します。
static int	CALLHANDLE_UNKNOWN_TO_LINECONTROL	このエラーは、回線制御側で検知されていないコールをリダイレクトしようとしたことを示します。
static int	CANNOT_OPEN_DEVICE	このエラーは、関連するデバイスが登録解除のため、デバイスのオープンに失敗したことを示します。
static int	CANNOT_TERMINATE_MEDIA_ON_PHONE	このエラーは、デバイスに実際の電話がある場合に、メディアを終端できない（常に、電話がメディアを終端する）ことを示します。
static int	CFWDALL_ALREADY_SET	このエラーは、すでにオンになっている CFWALL をオンにしようとしたことを示します。
static int	CFWDALL_DESTN_INVALID	このエラーは、無効な宛先に CFWALL を実行しようとしたことを示します。
static int	CLUSTER_LINK_FAILURE	このエラーは、クラスタ内にある Cisco Unified CM の 1 つへのリンクが失敗したことを示します（ネットワーク エラー）。
static int	COMMAND_NOT_IMPLEMENTED_ON_DEVICE	このエラーは、デバイスがコマンドをサポートしていないことを示します。
static int	CONFERENCE_ALREADY_PRESENT	このエラーは、すでに会議に参加している通話者に会議を実行しようとしたことを示します。
static int	CONFERENCE_FAILED	このエラーは、会議の開催が成功しなかったことを示します。
static int	CONFERENCE_FULL	このエラーは、すべての会議ブリッジが使用中であることを示します。
static int	CONFERENCE_INACTIVE	このエラーは、コンサルト会議が有効ではないときに、会議を開催しようとしたことを示します。
static int	CONFERENCE_INVALID_PARTICIPANT	このエラーは、自分自身または無効な関係者に会議を実行しようとしたことを示します。
static int	CTIERR_ACCESS_TO_DEVICE_DENIED	このエラーは、デバイスへのアクセスが拒否されたことを示します。
static int	CTIERR_APP_SOFTKEYS_ALREADY_CONTROLLED	このエラーは、アプリケーションのソフトキーが別のアプリケーションによってすでに制御されていることを示します。
static int	CTIERR_APPLICATION_DATA_SIZE_EXCEEDED	このエラーは、アプリケーションのデータ サイズが限度を超えていることを示します。

表 6-82　CiscoJtapiException のフィールド （続き）

インターフェイス	フィールド	説明
static int	CTIERR_BIB_NOT_CONFIGURED	このエラーは、ビルトインブリッジ（BIB）が構成されていないことを示します。
static int	CTIERR_BIB_RESOURCE_NOT_AVAILABLE	このエラーは、ビルトインブリッジ（BIB）リソースが利用できないことを示します。
static int	CTIERR_CALL_MANAGER_NOT_AVAILABLE	このエラーは、Communications Manager が現在利用できないことを示します。
static int	CTIERR_CALL_NOT_EXISTED	このエラーは、コールが存在していないことを示します。
static int	CTIERR_CALL_PARK_NO_DN	このエラーは、コール パーク DN がないことを示します。
static int	CTIERR_CALL_REQUEST_ALREADY_OUTSTANDING	このエラーは、コール要求が未処理であることを示します。
static int	CTIERR_CALL_UNPARK_FAILED	このエラーは、コールのパーク解除が失敗したことを示します。
static int	CTIERR_CAPABILITIES_DO_NOT_MATCH	このエラーは、機能が適合しないことを示します。
static int	CTIERR_CLOSE_DELAY_NOT_SUPPORTED_WITH_REG_TYPE	このエラーは、この登録タイプではクローズ遅延がサポートされないことを示します。
static int	CTIERR_CONFERENCE_ALREADY_EXISTED	このエラーは、会議がすでに存在していることを示します。
static int	CTIERR_CONFERENCE_NOT_EXISTED	このエラーは、会議が存在していないことを示します。
static int	CTIERR_CONNECTION_ON_INVALID_PORT	このエラーは、アプリケーションが無効なポートに接続しようとしていることを示します。
static int	CTIERR_CONSULT_CALL_FAILURE	このエラーは、コンサルト コールが失敗したことを示します。
static int	CTIERR_CONSULTCALL_ALREADY_OUTSTANDING	このエラーは、コンサルト コールが未処理であることを示します。
static int	CTIERR_CRYPTO_CAPABILITY_MISMATCH	このエラーは、デバイスの暗号アルゴリズムが現在のデバイス登録と適合してないためにデバイス登録が失敗ことを示します。
static int	CTIERR_CTIHANDLER_PROCESS_CREATION_FAILED	このエラーは、CTIHandler プロセスの作成が失敗したことを示します。
static int	CTIERR_DB_INITIALIZATION_ERROR	このエラーは、DB 初期化エラーを示します。
static int	CTIERR_DEVICE_ALREADY_OPENED	このエラーは、デバイスがすでにオープンされていることを示します。

表 6-82　CiscoJtapiException のフィールド （続き）

インターフェイス	フィールド	説明
static int	CTIERR_DEVICE_NOT_OPENED_YET	このエラーは、デバイスがまだオープンされていないことを示します。
static int	CTIERR_DEVICE_OWNER_ALIVE_TIMER_STARTED	このエラーは、デバイスの登録に失敗したことを示します。
static int	CTIERR_DEVICE_REGISTRATION_FAILED_NOT_SUPPORTED_MEDIATYPE	このエラーは、無効なメディア タイプであることを示します。インターコム回線の場合、CTIPort はダイナミック メディア ポート登録で登録する必要があります。
static int	CTIERR_DEVICE_RESTRICTED	このエラーは、デバイスが制限されていることを示します。
static int	CTIERR_DEVICE_SHUTTING_DOWN	このエラーは、デバイスがシャット ダウン中であることを示します。
static int	CTIERR_DIRECTORY_LOGIN_TIMEOUT	このエラーは、ディレクトリのログインがタイム アウトになっていることを示します。
static int	CTIERR_DUPLICATE_CALL_REFERENCE	このエラーは、コールの参照値が重複していることを示します。
static int	CTIERR_DYNREG_IPADDRMODE_MISMATCH	これは、シスコのメディア / ルート端末が複数のアドレッシング モードで登録されている場合に登録に失敗したことを示します。
static int	CTIERR_FAC_CMC_REASON_CMC_INVALID	入力した Client Matter Code (CMC) が無効です。
static int	CTIERR_FAC_CMC_REASON_CMC_NEEDED	コールをオファーするのに CMC が必要です。
static int	CTIERR_FAC_CMC_REASON_FAC_CMC_NEEDED	コールをオファーするのに Forced Authorization Code（FAC）および CMC が必要です。
static int	CTIERR_FAC_CMC_REASON_FAC_INVALID	入力された FAC が無効です。
static int	CTIERR_FAC_CMC_REASON_FAC_NEEDED	コールをオファーするのに FAC が必要です。
static int	CTIERR_FEATURE_ALREADY_REGISTERED	このエラーは、機能がすでに登録されていることを示します。
static int	CTIERR_FEATURE_DATA_REJECT	このエラーは、機能データが拒否されたことを示します。
static int	CTIERR_FEATURE_SELECT_FAILED	このエラーは、機能の選択に失敗したことを示します。
static int	CTIERR_ILLEGAL_DEVICE_TYPE	このエラーは、デバイス タイプが正しくないことを示します。

表 6-82　CiscoJtapiException のフィールド （続き）

インターフェイス	フィールド	説明
static int	CTIERR_INCOMPATIBLE_AUTOINSTALL_PROTOCOL_VERSION	このエラーは、自動インストール プロトコルのバージョンに互換性がないことを示します。
static int	CTIERR_INCORRECT_MEDIA_CAPABILITY	不正なメディア機能が原因で、デバイス登録が失敗しました。
static int	CTIERR_INFORMATION_NOT_AVAILABLE	このエラーは、情報がないことを示します。
static int	CTIERR_INTERCOM_SPEEDDIAL_ALREADY_CONFIGURED	このエラーは、インターコム ターゲットの値がすでに構成され、アプリケーションがインターコム ターゲット DN でコールを開始しようとしていることを示します。
static int	CTIERR_INTERCOM_SPEEDDIAL_ALREADY_SET	このエラーは、インターコム ターゲットの値がすでに設定されており、アプリケーションが設定し直そうとしているためにインターコム要求が失敗したことを示します。
static int	CTIERR_INTERCOM_SPEEDDIAL_DESTN_INVALID	このエラーは、インターコム ターゲットの値がインターコム グループ内にないためにインターコム要求が失敗したことを示します。
static int	CTIERR_INTERCOM_TALKBACK_ALREADY_PENDING	このエラーは、インターコム応答要求がすでに処理中であることを示します。
static int	CTIERR_INTERCOM_TALKBACK_FAILURE	このエラーは、何らかの理由で応答要求が失敗したことを示します。
static int	CTIERR_INTERNAL_FAILURE	このエラーは、CTI 内部エラーが発生したことを示します。
static int	CTIERR_INVALID_CALLID	このエラーは、コール ID が無効であることを示します。
static int	CTIERR_INVALID_DEVICE_NAME	このエラーは、デバイス名が無効であることを示します。
static int	CTIERR_INVALID_DTMFDIGITS	Play DTMF 要求が無効な番号であるために失敗しました。
static int	CTIERR_INVALID_FILTER_SIZE	このエラーは、フィルタ サイズが無効であることを示します。
static int	CTIERR_INVALID_MEDIA_DEVICE	このエラーは、メディア デバイスが無効であることを示します。
static int	CTIERR_INVALID_MEDIA_PARAMETER	このエラーは、メディア パラメータが無効であることを示します。
static int	CTIERR_INVALID_MEDIA_PROCESS	このエラーは、無効なメディア プロセスが存在していることを示します。
static int	CTIERR_INVALID_MEDIA_RESOURCE_ID	このエラーは、メディア リソース ID が無効であることを示します。

表 6-82　CiscoJtapiException のフィールド （続き）

インターフェイス	フィールド	説明
static int	CTIERR_INVALID_MESSAGE_HEADER_INFO	このエラーは、ヘッダー情報が無効であることを示します。
static int	CTIERR_INVALID_MESSAGE_LENGTH	このエラーは、メッセージの長さが無効であることを示します。
static int	CTIERR_INVALID_MONITOR_DESTN	このエラーは、無効なモニタリング対象のためにモニタリング要求が失敗したことを示します。
static int	CTIERR_INVALID_MONITOR_DN_TYPE	このエラーは、モニタリング DN タイプが無効であることを示します。
static int	CTIERR_INVALID_MONITORMODE	このエラーは、モニタリングモードが無効であるためにモニタリング要求が失敗したことを示します。
static int	CTIERR_INVALID_PARAMETER	このエラーは、パラメータが無効であることを示します。
static int	CTIERR_INVALID_PARK_DN	このエラーは、DN が 無効なパーク DN であることを示します。
static int	CTIERR_INVALID_PARK_REGISTRATION_HANDLE	このエラーは、ハンドルが無効なパーク登録ハンドルであることを示します。
static int	CTIERR_INVALID_RESOURCE_TYPE	このエラーは、リソース タイプが無効であることを示します。
static int	CTIERR_IPADDRMODE_MISMATCH	これは、IP アドレッシング モードが一致しないために登録に失敗したことを示します。
static int	CTIERR_LINE_OUT_OF_SERVICE	このエラーは、回線がアウトオブサービスであることを示します。
static int	CTIERR_LINE_RESTRICTED	このエラーは、回線が制限されていることを示します。
static int	CTIERR_MAXCALL_LIMIT_REACHED	このエラーは、最大コール制限値に達したことを示します。
static int	CTIERR_MEDIA_ALREADY_TERMINATED_DYNAMIC	このエラーは、デバイスを動的メディア終端で登録しようとしたためにデバイスの登録が失敗したことを示します。
static int	CTIERR_MEDIA_ALREADY_TERMINATED_NONE	このエラーは、デバイスがすでにメディア終端がなしで登録されているためにデバイスの登録が失敗したことを示します。
static int	CTIERR_MEDIA_ALREADY_TERMINATED_STATIC	このエラーは、デバイスを静的メディア終端で登録しようとしたためにデバイスの登録が失敗したことを示します。
static int	CTIERR_MEDIA_CAPABILITY_MISMATCH	このエラーは、デバイスのメディア機能が現在のデバイス登録と適合してないためにデバイス登録が失敗したことを示します。

表 6-82　CiscoJtapiException のフィールド （続き）

インターフェイス	フィールド	説明
static int	CTIERR_MEDIA_RESOURCE_NAME_SIZE_EXCEEDED	このエラーは、メディア リソース名のサイズが限度を超えていることを示します。
static int	CTIERR_MEDIAREGISTRATIONTYPE_DO_NOT_MATCH	このエラーは、メディア登録のタイプが一致していないことを示します。
static int	CTIERR_MESSAGE_TOO_BIG	このエラーは、メッセージが長すぎることを示します。
static int	CTIERR_MORE_ACTIVE_CALLS_THAN_RESERVED	このエラーは、アクティブなコールが予約されたコールよりも多いことを示します。
static int	CTIERR_NO_EXISTING_CALLS	このエラーは、コールが存在していないことを示します。
static int	CTIERR_NO_EXISTING_CONFERENCE	このエラーは、会議が存在していないことを示します。
static int	CTIERR_NO_RECORDING_SESSION	このエラーは、録音セッションがないために録音要求に失敗したことを示します。
static int	CTIERR_NO_RESPONSE_FROM_MP	このエラーは、メディア リソースからの応答がないことを示します。
static int	CTIERR_NOT_PRESERVED_CALL	このエラーは、コールが維持されていないことを示します。
static int	CTIERR_OPERATION_FAILED_QUIETCLEAR	このエラーは、一時的な障害が原因で、このコールの機能が利用できないことを示します。
static int	CTIERR_OPERATION_NOT_ALLOWED	このエラーは、この操作が許可されていないことを示します。
static int	CTIERR_OUT_OF_BANDWIDTH	このエラーは、帯域幅の範囲外であることを示します。
static int	CTIERR_OWNER_NOT_ALIVE	このエラーは、デバイスの登録が失敗したことを示します。
static int	CTIERR_PENDING_ACCEPT_OR_ANSWER_REQUEST	このエラーは、保留中の受け入れ要求または応答要求があることを示します。
static int	CTIERR_PENDING_START_MONITORING_REQUEST	このエラーは、保留中のモニタリング開始要求があることを示します。
static int	CTIERR_PENDING_START_RECORDING_REQUEST	このエラーは、保留中の録音開始要求があることを示します。
static int	CTIERR_PENDING_STOP_RECORDING_REQUEST	このエラーは、保留中の録音停止要求があることを示します。
static int	CTIERR_PRIMARY_CALL_INVALID	このエラーは、モニタリング要求のプライマリ コールが無効であるか、またはアイドル状態であることを示します。
static int	CTIERR_PRIMARY_CALL_STATE_INVALID	このエラーは、モニタリング要求のプライマリ コールが無効な状態であることを示します。

表 6-82　CiscoJtapiException のフィールド （続き）

インターフェイス	フィールド	説明
static int	CTIERR_RECORDING_ALREADY_INPROGRESS	このエラーは、録音がすでに進行中であるために録音要求が失敗したことを示します。
static int	CTIERR_RECORDING_CONFIG_NOT_MATCHING	このエラーは、録音設定が一致しないことを示します。
static int	CTIERR_RECORDING_SESSION_INACTIVE	このエラーは、録音セッションが非アクティブであるために録音要求に失敗したことを示します。
static int	CTIERR_REDIRECT_UNAUTHORIZED_COMMAND_USAGE	このエラーは、未承認のコマンド使用のリダイレクトを示します。
static int	CTIERR_REGISTER_FEATURE_ACTIVATION_FAILED	このエラーは、登録機能のアクティベーションに失敗したことを示します。
static int	CTIERR_REGISTER_FEATURE_APP_ALREADY_REGISTERED	機能登録アプリケーションがすでに登録されています。
static int	CTIERR_REGISTER_FEATURE_PROVIDER_NOT_REGISTERED	機能登録プロバイダーが登録されていません。
static int	CTIERR_RESOURCE_NOT_AVAILABLE	このエラーは、要求を処理するためにリソースを利用できないことを示します。
static int	CTIERR_START_MONITORING_FAILED	このエラーは、モニタリング開始要求に失敗したことを示します。
static int	CTIERR_START_RECORDING_FAILED	このエラーは、録音開始要求に失敗したことを示します。
static int	CTIERR_STATION_SHUT_DOWN	このエラーは、ステーションのシャットダウンが存在することを示します。
static int	CTIERR_SYSTEM_ERROR	このエラーは、CTI システム エラーを示します。
static int	CTIERR_UDP_PASS_THROUGH_NOT_SUPPORTED	このエラーは、UDP データ パススルーがサポートされないことを示します。
static int	CTIERR_UNKNOWN_EXCEPTION	このエラーは、不明な例外が発生したことを示します。
static int	CTIERR_UNSUPPORTED_CALL_PARK_TYPE	このエラーは、コール パークのタイプがサポートされないことを示します。
static int	CTIERR_UNSUPPORTED_CFWD_TYPE	このエラーは、コール転送タイプがサポートされないことを示します。
static int	CTIERR_USER_NOT_AUTH_FOR_SECURITY	このエラーは、ユーザがセキュア接続のために認証されていないことを示します。
static int	CTIERR_REDIRECT_CALL_PARTITIONING_POLICY	このエラーは、リダイレクトが認証されていないことを示します。
static int	CTIERR_FEATURE_NOT_AVAILABLE	このエラーは、機能が利用できないことを示します。

表 6-82　CiscoJtapiException のフィールド （続き）

インターフェイス	フィールド	説明
static int	DARES_INVALID_REQ_TYPE	このエラーは、コール処理内部エラー、DaRes 無効要求タイプが存在することを示します。
static int	DATA_SIZE_LIMIT_EXCEEDED	このエラーは、XML データ オブジェクト サイズが、許容値よりも大きいことを示します。
static int	DB_ERROR	このエラーは、デバイス クエリーに不正なデバイス タイプがあることを示します。
static int	DB_ILLEGAL_DEVICE_TYPE	このエラーは、DB に不正なデバイス タイプがあることを示します。
static int	DB_NO_MORE_DEVICES	このエラーは使用されなくなりました。
static int	DESTINATION_BUSY	このエラーは、転送先が通話中であることを示します。
static int	DESTINATION_UNKNOWN	このエラーは、転送先が見つからないことを示します。
static int	DEVICE_ALREADY_REGISTERED	このエラーは、デバイスがすでに登録されているためにデバイス登録が失敗したことを示します。
static int	DEVICE_NOT_OPEN	このエラーは、デバイスがオープンされていないか、または登録されていないため、回線のオープンに失敗したことを示します。
static int	DEVICE_OUT_OF_SERVICE	このエラーは、デバイスがアウトオブサービスであることを示します。
static int	DIGIT_GENERATION_ALREADY_IN_PROGRESS	このエラーは、番号の生成が進行中であることを示します。
static int	DIGIT_GENERATION_CALLSTATE_CHANGED	このエラーは、コールの状態が無効で継続できないことを示します。
static int	DIGIT_GENERATION_WRONG_CALL_HANDLE	このエラーは、コール ハンドルが無効で、コールが存在しない可能性があることを示します。
static int	DIGIT_GENERATION_WRONG_CALL_STATE	このエラーは、コールの状態が無効で、番号を生成できないことを示します。
static int	DIRECTORY_LOGIN_FAILED	このエラーは、ディレクトリ ログインに失敗し、ディレクトリが初期化されないことを示します。
static int	DIRECTORY_LOGIN_NOT_ALLOWED	このエラーは、ディレクトリ ログインに失敗したことを示します。
static int	DIRECTORY_TEMPORARY_UNAVAILABLE	このエラーは、ディレクトリが一時的に利用不能になっていることを示します。
static int	EXISTING_FIRSTPARTY	このエラーは、すでにデバイス制御メディアが存在していることを示します。

表 6-82　CiscoJtapiException のフィールド （続き）

インターフェイス	フィールド	説明
static int	HOLDFAILED	このエラーは、保留が回線制御層またはコール制御層によって拒否されたことを示します。
static int	ILLEGAL_CALLINGPARTY	このエラーは、デバイス上にない発側を使用して、発呼しようとしたことを示します。
static int	ILLEGAL_CALLSTATE	このエラーは、要求を呼び出すには、回線が正しい状態にないことを示します。
static int	ILLEGAL_HANDLE	このエラーは、ハンドルが無効であることを示します。
static int	ILLEGAL_MESSAGE_FORMAT	このエラーは、QBE プロトコル エラーが存在することを示します。
static int	INCOMPATIBLE_PROTOCOL_VERSION	このエラーは、JTAPI と CTI のバージョンに互換性がないことを示します。CTI エラー プロトコルのバージョンをサポートしません。
static int	INVALID_LINE_HANDLE	このエラーは、無効な回線ハンドルで回線操作を実行しようとしたことを示します。
static int	INVALID_RING_OPTION	このエラーは、呼出音オプションが無効であることを示します。
static int	LINE_GREATER_THAN_MAX_LINE	このエラーは、回線がこのデバイス上で利用できる最大回線数よりも多いことを示します。
static int	LINE_INFO_DOES_NOT_EXIST	このエラーは、回線情報がデータベースに存在しないことを示します。
static int	LINE_NOT_PRIMARY	このエラーは、コール制御から内部エラーが返されたことを示します。
static int	LINECONTROL_FAILURE	このエラーは、新規のコールの状態が原因で、回線制御がそのコールの開始を拒否したことを示します。
static int	MAX_NUMBER_OF_CTI_CONNECTIONS_REACHED	CTI 接続の最大数に達しました。
static int	MSGWAITING_DESTN_INVALID	このエラーは、無効な DN に対してメッセージ待ちランプをセットしようとしたことを示します。メッセージ待ち宛先が見つかりません。
static int	NO_ACTIVE_DEVICE_FOR_THIRDPARTY	このエラーは、サード パーティに対してアクティブなデバイスがないことを示します。
static int	NO_CONFERENCE_BRIDGE	このエラーは、会議ブリッジがないことを示します。
static int	NOT_INITIALIZED	このエラーは、CTI の初期化が完了する前に、プロバイダーをオープンしようとしたことを示します。
static int	PROTOCOL_TIMEOUT	コール制御から内部エラーが返されました。

表 6-82　CiscoJtapiException のフィールド （続き）

インターフェイス	フィールド	説明
static int	PROVIDER_ALREADY_OPEN	このエラーは、プロバイダーを再オープンしようとしたことを示します。
static int	PROVIDER_CLOSED	すでにクローズされているプロバイダーをクローズしようとした。
static int	PROVIDER_NOT_OPEN	このエラーは、デバイスの一覧が不完全であるか、デバイスの一覧の照会がタイムアウトになったか、または照会が中断されたことを示します。
static int	REDIRECT_CALL_CALL_TABLE_FULL	このエラーは、コール制御から内部エラーが返されたことを示します。
static int	REDIRECT_CALL_DESTINATION_BUSY	このエラーは、転送先が通話中であることを示します。
static int	REDIRECT_CALL_DESTINATION_OUT_OF_ORDER	このエラーは、リダイレクト先が故障中ことを示します。
static int	REDIRECT_CALL_DIGIT_ANALYSIS_TIMEOUT	このエラーは、ディジット分析がタイムアウトになったことを示します。これはコール制御から返された内部エラーです。
static int	REDIRECT_CALL_DOES_NOT_EXIST	このエラーは、存在しないまたは現在有効ではないコールをリダイレクトしようとしたことを示します。
static int	REDIRECT_CALL_INCOMPATIBLE_STATE	このエラーは、コール制御から内部エラーが返されたことを示します。
static int	REDIRECT_CALL_MEDIA_CONNECTION_FAILED	このエラーは、メディア接続エラーを示します。これはコール制御から返された内部エラーです。
static int	REDIRECT_CALL_NORMAL_CLEARING	このエラーは、通常のコールのクリアのためにリダイレクトが失敗したことを示します。
static int	REDIRECT_CALL_ORIGINATOR_ABANDONED	このエラーは、リダイレクトされるコールの遠端がハングアップしていることを示します。
static int	REDIRECT_CALL_PARTY_TABLE_FULL	このエラーは、コール制御から内部エラーが返されたことを示します。
static int	REDIRECT_CALL_PENDING_REDIRECT_TRANSACTION	このエラーは、コール制御から内部エラーが返されたことを示します。
static int	REDIRECT_CALL_PROTOCOL_ERROR	このエラーは、プロトコル エラーを示します。これはコール制御から返された内部エラーです。
static int	REDIRECT_CALL_UNKNOWN_DESTINATION	このエラーは、不明な宛先にリダイレクトしようとしたことを示します。
static int	REDIRECT_CALL_UNKNOWN_ERROR	このエラーは、コール制御から内部エラーが返されたことを示します。

表 6-82　CiscoJtapiException のフィールド （続き）

インターフェイス	フィールド	説明
static int	REDIRECT_CALL_UNKNOWN_PARTY	このエラーは、不明な通話者が検出されたことを示します。これはコール制御から返された内部エラーです。
static int	REDIRECT_CALL_UNRECOGNIZED_MANAGER	このエラーは、コール制御から内部エラーが返されたことを示します。
static int	REDIRECT_CALLINFO_ERR	このエラーは、コール制御から内部エラーが返されたことを示します。
static int	REDIRECT_ERR	このエラーは、コール制御から内部エラーが返されたことを示します。
static int	RETRIEVEFAILED	このエラーは、コールの取得が回線制御またはコール制御によって拒否されたことを示します。
static int	RETRIEVEFAILED_ACTIVE_CALL_ON_LINE	このエラーは、回線に他のコールがすでに存在するために、保留コールの取得中にエラーが発生したことを示します。
static int	SSAPI_NOT_REGISTERED	このエラーは、内部サポート インターフェイスが初期化されていないときに、リダイレクト コマンドが発行されたことを示します。CTI が初期化を完了していないか、内部エラーが生じています。
static int	TIMEOUT	このエラーは、要求がタイムアウトになったことを示します。
static int	TRANSFER_INACTIVE	このエラーは、コンサルト転送が存在しないときに、転送を実行しようとしたことを示します。
static int	TRANSFERFAILED	このエラーは、コール レッグの 1 つが遠端でハング アップしているか、接続解除されたために、転送が失敗した可能性があることを示します。
static int	TRANSFERFAILED_CALLCONTROL_TIMEOUT	このエラーは、転送の間、コール制御から予期される応答を受信していないことを示します。
static int	TRANSFERFAILED_DESTINATION_BUSY	このエラーは、使用中状態の宛先に転送しようとしたことを示します。
static int	TRANSFERFAILED_DESTINATION_UNALLOCATED	このエラーは、登録されていない電話番号に転送しようとしたことを示します。
static int	TRANSFERFAILED_OUTSTANDING_TRANSFER	このエラーは、既存の転送が進行中であることを示します。
static int	UNDEFINED_LINE	このエラーは、指定された回線が、デバイスで見つからないことを示します。
static int	UNKNOWN_GLOBAL_CALL_HANDLE	このエラーは、グローバル コール ハンドルが不明であることを示します。

表 6-82　CiscoJtapiException のフィールド （続き）

インターフェイス	フィールド	説明
static int	UNRECOGNIZABLE_PDU	このエラーは、QBE プロトコル エラーが存在することを示します。
static int	UNSPECIFIED	このエラーは、詳細不明のエラーが発生したことを示します。

継承したフィールド

なし

メソッド

表 6-83　CiscoJtapiException のメソッド

インターフェイス	メソッド	説明
int	getErrorCode()	この例外の errorCode を整数として返します。
java.lang.String	getErrorDescription()	errorCode の詳細な説明を返します。
java.lang.String	getErrorDescription(int errorCode)	**推奨されません。** 代わりに、String getErrorDescription (); を使用してください。errorCode の詳細な説明を返します。
java.lang.String	getErrorName()	文字列の形式で例外を返します。
java.lang.String	getErrorName(int errorCode)	**推奨されません。** 代わりに、String getErrorName (); を使用してください。文字列の形式で例外を返します。

継承したメソッド

なし

関連資料

詳細については、「定数フィールド値」（P.F-1）を参照してください。

CiscoJtapiPeer

CiscoJtapiPeer は、com.cisco.services.tracing.TraceModule インターフェイスを拡張することによって、アプリケーションからトレース情報へのアクセスを可能にします。Cisco JTAPI の実装によって作成された JtapiPeer オブジェクトのインスタンスはすべて、このインターフェイスを実装します。Cisco JTAPI 実装のトレース設定を操作する必要があるアプリケーションは、CiscoJtapiPeer.getTraceManager メソッドを使用して TraceManager オブジェクトを取得できます。アプリケーションは TraceManager オブジェクトを com.cisco.services.tracing パッケージに記述されたとおりに操作できます。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

CiscoObjectContainer, javax.telephony.JtapiPeer, TraceModule

宣言

public interface CiscoJtapiPeer extends TraceModule, javax.telephony.JtapiPeer, CiscoObjectContainer

フィールド

なし

メソッド

表 6-84　CiscoJtapiPeer のメソッド

インターフェイス	メソッド	説明
CiscoJtapiProperties	getJtapiProperties()	アプリケーションが使用できる各種のメソッドを定義します。アプリケーションはそれらのメソッドによって、JTAPI レイヤで使用されるパラメータを変更できます。

表 6-84　CiscoJtapiPeer のメソッド （続き）

インターフェイス	メソッド	説明
void	setJtapiProperties(CiscoJtapiProperties jtapiproperties)	jtapi.ini ファイルの CiscoJtapiProperties で行われた変更を保存する機能をアプリケーションに付与し、JTAPIPeer のプロバイダーのプロパティでこれらの変更を有効にします。

継承したメソッド

インターフェイス com.cisco.services.tracing.TraceModule から
getTraceManager, getTraceModuleName

インターフェイス javax.telephony.JtapiPeer から
getName, getProvider, getServices

インターフェイス com.cisco.jtapi.extensions.CiscoObjectContainer から
getObject, setObject

関連資料

詳細については、CiscoJtapiProperties と TraceModule を参照してください。

CiscoJtapiProperties

Cisco Unified JTAPI の動作と機能は多くのパラメータによって調整されます。それらのパラメータは、CiscoJtapiPeer がインスタンス化される際に、jtapi.ini ファイルから読み込まれます。アプリケーションは、この CiscoJtapiproperties インターフェイスを介して、これらのパラメータを制御できます。

アプリケーションは CiscoJtapiproperties プロパティ オブジェクトを照会して、そのアプリケーションの機能に適するようにこれらのパラメータを変更できます。また、CiscoJtapiproperties インターフェイスを介してプロパティにアクセスすることにより、（アプリケーション側から）これらのパラメータを管理する場所が一元化されます。最も可視性の高いパラメータは、トレース レベルとトレース対象を定義しているパラメータです。

使用法

```
JtapiPeer peer = JtapiPeerFactory.getJtapiPeer ( null );
 if(peer instanceof CiscoJtapiPeer) {
 CiscoJtapiProperties jProps = ((CiscoJtapiPeer)peer).getJtapiProperties();
 jProps.setTracePath("¥¥D:¥¥Traces¥¥WorkFlow"); jProps.setUseJavaConsoleTrace(false);
 MyProviderObserver providerObserver = new MyProviderObserver ();
 provider = peer.getProvider ( providerName ); }
```

上記の例では、アプリケーションで Java コンソールのトレースをオフに設定し、トレースのパスを D:¥Traces¥WorkFlowApp1 に設定しています。ピアが取得されると、jtapi.ini ファイル内で設定されているパラメータが読み込まれ、CiscoJtapiProperties を実装するオブジェクトが作成されます。

jtapi.ini ファイルがクラスパスに存在しない場合は、デフォルト設定を使用してこのオブジェクトが作成されます。getProvider () の最初の呼び出し時に、Cisco Jtapi で使用されるパラメータが読み込まれて確定されます。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

宣言

public interface CiscoJtapiProperties

フィールド

なし

メソッド

表 6-85　CiscoJtapiProperties のメソッド

インターフェイス	メソッド	説明
void	deleteCertificates(java.lang.String username, java.lang.String instanceID, java.lang.String ccmCAPFAddress, java.lang.String certificatePath)	証明書ストアにインストールされた USER インスタンスの X.509 クライアント証明書を削除します。
void	deleteSecurityPropertyForInstance(java.lang.String username, java.lang.String instanceID, java.lang.String capfIp, java.lang.String certPath)	jtapi.ini ファイルからセキュリティを適切に削除し、ユーザ名またはインスタンス ID に対して以前にインストールされた証明書も削除します。
java.lang.String	getAlarmServiceHostname()	アラーム サービスのホスト名を取得します。
int	getAlarmServicePort()	アラーム サービスのポート番号を取得します。
boolean	getCallSecurityStatusChangedEv()	イベント CallSecurityStatusChangedEv を受け取るかどうかをアプリケーションに指示します（該当する場合）。

表 6-85　CiscoJtapiProperties のメソッド （続き）

インターフェイス	メソッド	説明
int	getCtiRequestTimeout()	プロバイダーのオープンを除く CTI 要求のタイムアウトを取得します（秒）。
java.lang.String[]	getDebuggingNames()	サポートされている jtapi デバッグ レベル トレースの名前を取得します。
boolean	getDebuggingValue(java.lang.String debuggingName)	デバッグ レベル トレースの有効状態または無効状態を取得します。
int	getDesiredServerHeartbeatInterval()	CTI Manager が JTAPI にハートビートを送信する間隔として指定されている時間を取得します（秒）。
boolean	getDiscConnBeforeCreatingInCPIC()	リダイレクト先だけがアプリケーションによって監視される場合のシナリオのイベントの順序を制御します。このインターフェイスは、ConnDisconnectedEv が ConnCreatedEv の前に送信される場合に true を返し、そうでない場合は false を返します。
java.lang.String	getFileNameBase()	各ログ ファイルのファイル名。
java.lang.String	getFileNameExtension()	ログ ファイルのファイル名拡張子を取得します。
int	getJavaSocketConnectTimeout()	SOCKET CONNECT TIMEOUT のサービス パラメータ値を返します（秒）。
int	getNumTraceFiles()	ロールオーバー前のトレース ファイルの数を取得します。
boolean	getPeriodicWakeupEnabled()	定期起動の有効状態を取得します。
int	getPeriodicWakeupInterval()	定期起動の間隔を取得します（ミリ秒）。
boolean	getProcessOfferingAfterNewcallevent()	jtapi.ini パラメータ ProcessOfferringAfterNewcallEvent' のブール値を取得します。デフォルトでは、このインターフェイスは false を返します。

表 6-85　CiscoJtapiProperties のメソッド （続き）

インターフェイス	メソッド	説明
int	getProviderOpenRequestTimeout()	プロバイダー オープン要求のタイムアウトを設定します（秒）。
int	getProviderOpenRetryAttempts()	CTI Manager への再接続試行の最大数のサービス パラメータ値を返します。
int	getProviderRetryInterval()	CTI Manager への接続再試行の間隔を取得します（秒）。
int	getQueueSizeThreshold()	アラームをトリガするイベント キュー サイズのしきい値を取得します。
boolean	getQueueStatsEnabled()	イベント キュー統計値の有効状態を取得します。
int	getRouteSelectTimeout()	ルート選択のタイムアウトを取得します（ミリ秒）。
java.util.Hashtable	getSecurityPropertyForInstance()	ユーザおよび InstanceIDs のすべてのパラメータが設定されたハッシュ テーブルを返します。 キーと値のペアについては、「User/InstanceID のハッシュ テーブル」（P.6-172）を参照してください。
java.util.Hashtable	getSecurityPropertyForInstance(java.lang.String user, java.lang.String instanceID)	ユーザおよび InstanceIDs のすべてのパラメータが設定されたハッシュ テーブルを返します。 キーと値のペアについては、「User/InstanceID のハッシュ テーブル」（P.6-172）を参照してください。
java.lang.String[]	getServices()	この実装がサポートしているサービスを返します。
java.lang.String	getSyslogCollector()	syslog コレクタのホスト名を取得します。
int	getSyslogCollectorUDPPort()	syslog コレクタの UDP ポートを取得します。
java.lang.String	getTraceDirectory()	トレース ファイルの書き込み先のパス ディレクトリ。
int	getTraceFileSize()	ロールオーバー前のトレース ファイルのサイズ。

表 6-85　CiscoJtapiProperties のメソッド （続き）

インターフェイス	メソッド	説明
java.lang.String[]	getTraceNames()	サポートされている jtapi トレースの名前を取得します。
java.lang.String	getTracePath()	トレース ファイルが配置されるパスを取得します。
boolean	getTraceValue(java.lang.String traceName)	トレースの有効状態または無効状態を取得します。
boolean	getUpdateJtapiCalledWithOriginalCalled()	更新された着側アドレスに対する Jtapi の動作を変更するパラメータ設定を照会します。
boolean	getUseAlarmService()	アラーム サービスの有効状態または無効状態を取得します。
boolean	getUseFileTrace()	jtapi ログ ファイル トレースの有効状態または無効状態を取得します。
boolean	getUseJavaConsoleTrace()	jtapi コンソール トレースの有効状態または無効状態を取得します。
boolean	getUseSameDir()	UseSameDir が true の場合は、同じディレクトリにトレース ファイルが書き込まれます。
boolean	getUseSyslog()	syslog トレースの有効状態または無効状態を取得します。
boolean	IsCertificateUpdated(java.lang.String user, java.lang.String instanceID)	指定したユーザ/インスタンス ID のクライアントおよびサーバの証明書が更新される場合、またはクライアントおよびサーバの証明書が更新されていない場合に関する情報を提供します。
void	setAlarmServiceHostname(java.lang.String hostname)	アラーム サービスのホスト名を設定します。
void	setAlarmServicePort(int portNumber)	アラーム サービスの受信を行うポート番号を設定します。
void	setCallSecurityStatusChangedEv(boolean val)	アプリケーションで、CallSecurityStatusChangedEv が true または false のどちらかを受信するためのフィルタを設定できます。

表 6-85　CiscoJtapiProperties のメソッド（続き）

インターフェイス	メソッド	説明
void	setCtiRequestTimeout(int seconds)	プロバイダーのオープンを除く CTI 要求のタイムアウトを設定します（秒）。
void	setDebuggingValue(java.lang.String debuggingName, boolean value)	特定のデバッグ レベル トレースを有効または無効にします。
void	setDesiredServerHeartbeatInterval(int seconds)	CTI Manager が JTAPI にハートビートを送信する間隔として指定されている時間を設定します（秒）。
void	setDiscConnBeforeCreatingInCPIC(boolean val)	イベントの順序を設定し、リダイレクト先でのリダイレクト中に作成される Connection よりも前に、Disconnect を送信します。
void	setFileNameBase(java.lang.String base)	ログ ファイルのファイル名を設定します。
void	setFileNameExtension(java.lang.String extn)	ログ ファイルのファイル名拡張子を設定します。
void	setJavaSocketConnectTimeout(int timeout)	アプリケーションでソケット接続タイムアウトを設定できるようにします（秒）。
void	setNumTraceFiles(int val)	ロールオーバー前のトレース ファイルの数を設定します。
void	setPeriodicWakeupEnabled(boolean enabled)	定期起動を有効または無効状態に設定します。
void	setPeriodicWakeupInterval(int milliseconds)	定期起動の間隔を設定します（ミリ秒）。
void	setProcessOfferingAfterNewcallevent(boolean val)	転送先だけがアプリケーションによって監視され、転送が Offering 状態で完了する転送シナリオのためのイベントの順序を制御します。
void	setProviderOpenRequestTimeout(int seconds)	プロバイダー オープン要求のタイムアウトを設定します（秒）。
void	setProviderOpenRetryAttempts(int retryAttempts)	アプリケーションで JTAPI の CTI Manager への再接続試行を設定できるようにします。
void	setProviderRetryInterval(int seconds)	CTI Manager への接続再試行の間隔を設定します（秒）。

表 6-85　CiscoJtapiProperties のメソッド （続き）

インターフェイス	メソッド	説明
void	setQueueSizeThreshold(int size)	アラームをトリガするイベント キュー サイズのしきい値を設定します。
void	setQueueStatsEnabled(boolean enabled)	イベント キュー統計値を有効または無効にします。
void	setRouteSelectTimeout(int milliseconds)	ルート選択のタイムアウトを設定します（ミリ秒）。
void	setSecurityPropertyForInstance(java.lang.String user, java.lang.String instanceID, java.lang.String authCode, java.lang.String tftp, java.lang.String tftpPort, java.lang.String capf, java.lang.String capfPort, java.lang.String certPath, boolean securityOption)	**推奨されません。** このメソッドは、オーバーロードされたメソッド setSecurityPropertyForInstance に置き換えられました。このメソッドは、Java キー ストアのパスフレーズである追加のパラメータ certStorePassphrase を取ります。このメソッドにはセキュリティの脆弱性がある可能性があります。
void	setSecurityPropertyForInstance(java.lang.String user, java.lang.String instanceID, java.lang.String authCode, java.lang.String tftp, java.lang.String tftpPort, java.lang.String capf, java.lang.String capfPort, java.lang.String certPath, boolean securityOption, java.lang.String certstorePassphrase)	アプリケーションにサーバ/クライアント証明書をダウンロードする機能を提供し、JTAPI の jtapi.ini ファイル内のアプリケーション インスタンスのセキュリティ プロパティを設定します。
void	setServices(java.lang.String[] services)	利用可能なサービスのリストを設定します。
void	setSyslogCollector(java.lang.String value)	syslog コレクタのホスト名を設定します。
void	setSyslogCollectorUDPPort(int port)	syslog コレクタの UDP ポートを設定します。
void	setTraceDirectory(java.lang.String dir)	jtapi トレース ファイルの書き込み先のディレクトリを設定します。
void	setTraceFileSize(int val)	トレース ファイルのサイズを設定します。
void	setTracePath(java.lang.String path)	jtapi トレースの書き込み先のディレクトリ ルートを設定します。
void	setTraceValue(java.lang.String traceName, boolean value)	特定のトレースを有効または無効にします。

表 6-85　CiscoJtapiProperties のメソッド （続き）

インターフェイス	メソッド	説明
void	setUpdateJtapiCalledWithOriginalCalled(boolean val)	このパラメータが True に設定されると、Jtapi 着信者情報が常に元の着信者で更新されます。
void	setUseAlarmService(boolean value)	アラーム サービスを有効または無効にします。
void	setUseFileTrace(boolean value)	jtapi ログ ファイル トレースを有効または無効にします。
void	setUseJavaConsoleTrace(boolean value)	jtapi コンソール トレースを有効または無効にします。
void	setUseSameDir(boolean value)	UseSameDir が true の場合は、同じディレクトリにトレース ファイルが書き込まれます。
void	setUseSyslog(boolean value)	syslog トレースを有効または無効にします。
void	updateCertificate(java.lang.String user, java.lang.String intanceID, java.lang.String authcode, java.lang.String ccmTFTPAddress, java.lang.String ccmTFTPPort, java.lang.String ccmCAPFAddress, java.lang.String ccmCAPFPort, java.lang.String certificatePath)	**推奨されません。** このメソッドは、オーバーロードされたメソッド updateCertifcate に置き換えられました。このメソッドは、Java キー ストアのパスフレーズである追加のパラメータ certStorePassphrase を取ります。このメソッドにはセキュリティの脆弱性がある可能性があります。
void	updateCertificate(java.lang.String user, java.lang.String intanceID, java.lang.String authcode, java.lang.String ccmTFTPAddress, java.lang.String ccmTFTPPort, java.lang.String ccmCAPFAddress, java.lang.String ccmCAPFPort, java.lang.String certificatePath, java.lang.String certStorePassphrase)	証明書ストアに USER インスタンスの X.509 クライアント証明書をインストールします。
void	updateServerCertificate(java.lang.String ccmTFTPAddress, java.lang.String ccmTFTPPort, java.lang.String ccmCAPFAddress, java.lang.String ccmCAPFPort, java.lang.String certificatePath)	**推奨されません。** このメソッドは、オーバーロードされたメソッド updateServerCertifcate に置き換えられました。このメソッドは、Java キー ストアのパスフレーズである追加のパラメータ certStorePassphrase を取ります。このメソッドにはセキュリティの脆弱性がある可能性があります。

表 6-85　CiscoJtapiProperties のメソッド（続き）

インターフェイス	メソッド	説明
void	updateServerCertificate(java.lang.String userName, java.lang.String instanceID, java.lang.String ccmTFTPAddress, java.lang.String ccmTFTPPort, java.lang.String ccmCAPFAddress, java.lang.String ccmCAPFPort, java.lang.String certificatePath, java.lang.String certStorePassphrase)	指定した証明書パスに X.509 サーバ証明書をインストールします。

User/InstanceID のハッシュテーブル

表 6-86　User/InstanceID のハッシュテーブル

キー	値
「user」	userName
文字列「instanceID」	InstanceID
文字列「Authcode」	authCode
文字列「CAPF」	capfServerIP-Address
文字列「CAPFPort」	capfServer の IP アドレス / ポート
文字列「TFTP」	tftpServer の IP アドレス
文字列「TFTPPort」	tftpServer の IP アドレス / ポート
文字列「CertPath」	証明書パス
文字列「securityOption」	セキュリティ オプションを表すブール値（有効なら true、無効なら false）。
文字列「certificateStatus」	証明書のステータスを表すブール値（更新済みなら true、未更新なら false）。

関連資料

CiscoLocales

このインターフェイスは、Cisco Unified JTAPI がサポートするすべてのロケールを列挙します。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

宣言

public interface CiscoLocales

フィールド

表 6-87　CiscoLocales のフィールド

インターフェイス	フィールド
static int	LOCALE_ARABIC_ALGERIA
static int	LOCALE_ARABIC_BAHRAIN
static int	LOCALE_ARABIC_EGYPT
static int	LOCALE_ARABIC_IRAQ
static int	LOCALE_ARABIC_JORDAN
static int	LOCALE_ARABIC_KUWAIT
static int	LOCALE_ARABIC_LEBANON
static int	LOCALE_ARABIC_MOROCCO
static int	LOCALE_ARABIC_OMAN
static int	LOCALE_ARABIC_QATAR
static int	LOCALE_ARABIC_SAUDI_ARABIA
static int	LOCALE_ARABIC_TUNISIA
static int	LOCALE_ARABIC_UNITED_ARAB_EMIRATES
static int	LOCALE_ARABIC_YEMEN
static int	LOCALE_BULGARIAN_BULGARIA
static int	LOCALE_CATALAN_SPAIN
static int	LOCALE_CHINESE_HONG_KONG
static int	LOCALE_CROATIAN_CROATIA

表 6-87　CiscoLocales のフィールド

インターフェイス	フィールド
static int	LOCALE_CZECH_CZECH_REPUBLIC
static int	LOCALE_DANISH_DENMARK
static int	LOCALE_DUTCH_NETHERLAND
static int	LOCALE_ENGLISH_UNITED_KINGDOM
static int	LOCALE_ENGLISH_UNITED_STATES
static int	LOCALE_FINNISH_FINLAND
static int	LOCALE_FRENCH_FRANCE
static int	LOCALE_GERMAN_GERMANY
static int	LOCALE_GREEK_GREECE
static int	LOCALE_HEBREW_ISRAEL
static int	LOCALE_HUNGARIAN_HUNGARY
static int	LOCALE_ITALIAN_ITALY
static int	LOCALE_JAPANESE_JAPAN
static int	LOCALE_KOREAN_KOREA
static int	LOCALE_NORWEGIAN_NORWAY
static int	LOCALE_POLISH_POLAND
static int	LOCALE_PORTUGUESE_BRAZIL
static int	LOCALE_PORTUGUESE_PORTUGAL
static int	LOCALE_ROMANIAN_ROMANIA
static int	LOCALE_RUSSIAN_RUSSIA
static int	LOCALE_SERBIAN_REPUBLIC_OF_MONTENEGRO
static int	LOCALE_SERBIAN_REPUBLIC_OF_SERBIA
static int	LOCALE_SIMPLIFIED_CHINESE_CHINA
static int	LOCALE_SLOVAK_SLOVAKIA
static int	LOCALE_SLOVENIAN_SLOVENIA
static int	LOCALE_SPANISH_SPAIN
static int	LOCALE_SWEDISH_SWEDEN

表 6-87　CiscoLocales のフィールド

インターフェイス	フィールド
static int	LOCALE_THAI_THAILAND
static int	LOCALE_TRADITIONAL_CHINESE_CHINA

メソッド

なし

関連資料

「定数フィールド値」（P.F-1）を参照してください。

CiscoMediaConnectionMode

CiscoMediaConnectionMode インターフェイスは、すべてのメディア接続モードを列挙します。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

宣言

public interface CiscoMediaConnectionMode

フィールド

表 6-88　CiscoMediaConnectionMode のフィールド

インターフェイス	フィールド	説明
static int	NONE	送信チャネルも受信チャネルもアクティブではありません。
static int	RECEIVE_ONLY	受信チャネルだけがアクティブです。
static int	TRANSMIT_AND_RECEIVE	送信チャネルも受信チャネルもアクティブです。
static int	TRANSMIT_ONLY	送信チャネルだけがアクティブです。

メソッド

なし

関連資料

「定数フィールド値」（P.F-1）を参照してください。

CiscoMediaEncryptionAlgorithmType

CiscoMediaEncryptionAlgorithmType インターフェイスは、暗号化に使用される SRTP アルゴリズム タイプを示します。このインターフェイスは、アプリケーションが iscoRTPInputKeyEv および CiscoRTPOutputKeyEv で取得できるすべてのセキュリティ インジケータ値を列挙します。アプリケーションが CTIPort およびメディアの終端 RP 上で独自のメディアを終了させている場合、次のアルゴリズムのうち 1 つだけを登録 API で提供する必要があります。

インターフェイス履歴

Cisco Unified Communications Manager リリース	説明
3.x	拡張が追加されました。

スーパーインターフェイス

public interface CiscoMediaEncryptionAlgorithmType

フィールド

表 6-89　CiscoMediaEncryptionAlgorithmType のフィールド

インターフェイス	フィールド	説明
static int	AES_128_COUNTER	使用されるアルゴリズムは、Advanced Encryption Standard（AES; 高度暗号化規格）に基づきます。これは、コンピュータのセキュリティ標準です。暗号化スキームは、128 ビットのデータ ブロックを暗号化および復号化する対称ブロック サイファです。

関連資料

詳細については、「定数フィールド値」（P.F-1）を参照してください。

CiscoMediaEncryptionKeyInfo

CiscoMediaEncryptionKeyInfo インターフェイスでは、アプリケーションが SRTP キーに関する情報を取得できます。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

宣言

public interface CiscoMediaEncryptionKeyInfo

フィールド

なし

メソッド

表 6-90　CiscoMediaEncryptionKeyInfo のメソッド

インターフェイス	メソッド	説明
int	getAlgorithmID()	現在のストリームのメディア暗号化アルゴリズム ID を返します。
int	getIsMKIPresent()	MKI が存在するかどうかを示します。
byte[]	getKey()	ストリームのマスター鍵を返します。
int	getKeyLength()	このキーの keyLength を返します。
byte[]	getSalt()	ストリームのソルト鍵を返します。
int	getSaltLength()	このソルトの saltLength を返します。
int	keyDerivationRate()	このセッションの SRTP 鍵逸脱率を示します。

関連資料

CiscoRTPInputKeyEv と CiscoRTPOutputKeyEv を参照してください。

CiscoMediaOpenLogicalChannelEv

CiscoMediaOpenLogicalChannelEv イベントは、動的に登録される CiscoMediaTerminal または CiscoRouteTerminal に対してメディアが確立されるたびに送信されます。アプリケーションは、このイベントを受信すると、CiscoMediaTerminal または CiscoRouteTerminal に対して setRTPParams を起動し、メディアを終端する IP アドレスおよびポート番号をこのイベントで配信される rtpHandle とともに渡す必要があります。

アプリケーションは、CiscoProvider.getCall（CiscoRTPHandle）を使用してコール参照を取得できます。アプリケーションは、setRTPParams メソッドが起動されない限り、遠端やローカル エンドが機能を起動できないことに注意する必要があります。アプリケーションが指定時間内にこのイベントに応答しない場合、コールは切断されます。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.0(1)	getAddressingModeForMedia() メソッドが追加されました。

スーパーインターフェイス

CiscoEv, CiscoTermEv, javax.telephony.events.Ev, javax.telephony.events.TermEv

宣言

public interface CiscoMediaOpenLogicalChannelEv extends CiscoTermEv

フィールド

表 6-91　CiscoMediaOpenLogicalChannelEv のフィールド

インターフェイス	フィールド
static int	ID

継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

表 6-92　CiscoMediaOpenLogicalChannelEv のメソッド

インターフェイス	メソッド	説明
int	getAddressingModeForMedia()	int Application を返し、要求された IP アドレッシング モードに対する次の値を受け取ります。 • CiscoTerminal.IP_ADDRESSING_IPv4：アプリケーションが setRTPParams 要求で IPv4 形式の IP アドレスを提供する必要があることを意味します。 • CiscoTerminal.IP_ADDRESSING_IPv6：アプリケーションが set RTP Params 要求で IPv6 形式の IP アドレスを提供する必要があることを意味します。
CiscoRTPHandle	getCiscoRTPHandle()	CiscoRTPHandle オブジェクトを返します。アプリケーションは、このハンドルを RTP パラメータとともに CiscoMediaTerminal または CiscoRouteTerminal に渡す必要があります。アプリケーションは、CiscoProvider.getCall を使用してコール参照を取得できます。コール オブザーバがない場合や、このイベントの配信時にコール オブザーバがなかった場合、CiscoProvider.getCall は null を返す可能性があります。
int	getMediaConnectionMode()	CiscoMediaConnectionMode を返します。アプリケーションは次のいずれかの値を受け取ります。 • CiscoMediaConnectionMode.RECEIVE_ONLY：一方向メディアによる受信だけであることを意味します。 • CiscoMediaConnectionMode.TRANSMIT_AND_RECEIVE：双方向メディアであることを意味します。 通常、アプリケーションは値 NONE を受け取ることはありません。受け取った場合、アプリケーションはそのイベントを無視してエラーを記録します。

表 6-92　CiscoMediaOpenLogicalChannelEv のメソッド

インターフェイス	メソッド	説明
int	getPacketSize()	遠端のパケット サイズを返します（ミリ秒単位）。 getPacketSize
int	getPayLoadType()	次の定数の 1 つの遠端ペイロード形式を返します。 • CiscoRTPPayload.NONSTANDARD • CiscoRTPPayload.G711ALAW64K • CiscoRTPPayload.G711ALAW56K • CiscoRTPPayload.G711ULAW64K • CiscoRTPPayload.G711ULAW56K • CiscoRTPPayload.G722_64K • CiscoRTPPayload.G722_56K • CiscoRTPPayload.G722_48K • CiscoRTPPayload.G7231 • CiscoRTPPayload.G728 • CiscoRTPPayload.G729 • CiscoRTPPayload.G729ANNEXA • CiscoRTPPayload.IS11172AUDIOCAP • CiscoRTPPayload.IS13818AUDIOCAP • CiscoRTPPayload.ACY_G729AASSN • CiscoRTPPayload.DATA64 • CiscoRTPPayload.DATA56 • CiscoRTPPayload.GSM • CiscoRTPPayload.ACTIVEVOICE

継承したメソッド

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.TermEv から
getTerminal

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

「定数フィールド値」（P.F-1）および CiscoRTPParams を参照してください。

CiscoMediaSecurityIndicator

CiscoMediaSecurityIndicator は CiscoRTPInputKeyEv、CiscoRTPOutputKeyEv、および CiscoSnapShotRTPEv で送信されます。コール セキュリティ ステータスを表示します。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

宣言

public interface CiscoMediaSecurityIndicator

フィールド

表 6-93　CiscoMediaSecurityIndicator のフィールド

インターフェイス	フィールド	説明
static int	MEDIA_ENCRYPT_KEYS_AVAILABLE	セキュリティを確保したモードでメディアを終端し、鍵を参照できます。
static int	MEDIA_ENCRYPT_KEYS_UNAVAILABLE	セキュリティを確保したモードでメディアを終端しますが、SRTP が Cisco Unified Communications Manager Administration で有効にされていないため、鍵は参照できません。
static int	MEDIA_ENCRYPT_USER_NOT_AUTHORIZED	セキュリティを確保したモードでメディアを終端しますが、ユーザが鍵を取得する権限を持っていないため、鍵は参照できません。
static int	MEDIA_NOT_ENCRYPTED	メディアのこのコールは暗号化されていません。

関連資料

「定数フィールド値」（P.F-1）を参照してください。

CiscoMediaTerminal

CiscoMediaTerminal は、アプリケーションによる RTP メディア ストリームの終端を可能にする、特殊な CiscoTerminal です。CiscoMediaTerminal は、通常の CiscoTerminal と異なり、物理的なテレフォニー エンドポイントではないため、サードパーティ方式で監視、制御することができます。CiscoMediaTerminal は論理的なテレフォニー エンドポイントなので、メディアを終端する必要があるアプリケーションに関連付けることができます。そのようなアプリケーションには、音声メッセージング システム、Interactive Voice Response（IVR; 対話式音声自動応答）、ソフトフォンなどが含まれます。

（注）　Cisco Unified JTAPI により CiscoMediaTerminal として表現されるのは、CTIPort だけです。

メディアの終端プロセスには 2 つの段階があります。アプリケーションはまず、特定の端末向けのメディアを終端するために、Terminal.addObserver メソッドを使用して、CiscoTerminalObserver インターフェイスを実装するオブザーバを追加します。次に、CiscoMediaTerminal.register メソッドを使用して、その IP アドレス、およびその端末向けの着信 RTP ストリームの送信先となるポート番号を登録します。

コールごとに動的に IP アドレスおよびポート番号を提供するには、アプリケーションは、サポートしている機能だけを提示して登録する必要があります。アプリケーションは、メディアが確立されるたびに送信される CiscoMediaOpenLogicalChannelEv に応答する必要があります。このタイプに登録するアプリケーションは、メディアが確立されていない限り、このイベントが受信されたときに、遠端やローカル エンドがあらゆる機能を実行できないことに注意する必要があります。アプリケーションが指定時間内にこのイベントに応答しない場合、コールはドロップされます。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1x	IPv6 のサポートが追加されました。

スーパーインターフェイス

CiscoObjectContainer, CiscoTerminal, javax.telephony.Terminal

宣言

public interface CiscoMediaTerminal extends CiscoTerminal

フィールド

なし

継承したフィールド

インターフェイス com.cisco.jtapi.extensions.CiscoTerminal から
ASCII_ENCODING, DEVICESTATE_ACTIVE, DEVICESTATE_ALERTING, DEVICESTATE_HELD, DEVICESTATE_IDLE, DEVICESTATE_UNKNOWN, DEVICESTATE_WHISPER, DND_OPTION_CALL_REJECT, DND_OPTION_NONE, DND_OPTION_RINGER_OFF, IN_SERVICE, IP_ADDRESSING_MODE_IPV4, IP_ADDRESSING_MODE_IPV4_V6, IP_ADDRESSING_MODE_IPV6, IP_ADDRESSING_MODE_UNKNOWN, IP_ADDRESSING_MODE_UNKNOWN_ANATRED, NOT_APPLICABLE, OUT_OF_SERVICE, UCS2UNICODE_ENCODING, UNKNOWN_ENCODING

メソッド

表 6-94　Cisco*MediaTerminal* のメソッド

インターフェイス	メソッド	説明
void	register(java.net.InetAddress address, int port, CiscoMediaCapability[] capabilities)	このメソッドは MediaTerminal を登録し、MediaTerminal が登録されると、このメソッドは正常に終了します。 CiscoMediaTerminal は CiscoTerminal.UNREGISTERED 状態になっている必要があり、プロバイダーは Provider.IN_SERVICE 状態になっている必要があります。 このメソッドには 3 種類の引数があります。 • 最初の引数は、この端末の RTP メディア ストリームが終端されるインターネット アドレスを指定します。 • 2 番目の引数は RTP パケットが向けられる UDP ポートを指定します。 • 3 番目の引数は、アプリケーションがこの端末に対してサポートする RTP 符号化方式の種類を示します。 **パラメータ** • address : この端末の着信 IPv4 RTP ストリームが終端されるインターネット アドレスを指定します。 • port : RTP ストリームの受信に使用されるこの端末の UDP ポート。 • capabilities : アプリケーションがこの端末に対してサポートする RTP 符号化方式の種類のリスト。 **例外** CiscoRegistrationException

表 6-94　Cisco*MediaTerminal* のメソッド （続き）

インターフェイス	メソッド	説明
void	register(java.net.InetAddress address, int port)	**推奨されません。** アドレスとポートを指定して Terminal を登録します。デフォルトの符号化方式は G.711（64 KHz u-law）、パケット サイズは 30 ミリ秒です。 **パラメータ** • address：この端末の着信 IPv4 RTP ストリームのインターネット アドレス • port : RTP ストリームの受信に使用されるこの端末の UDP ポート **例外** CiscoRegistrationException
void	register(java.net.InetAddress address, int port,CiscoMediaCapability[] capabilities, int[] algorithmIDs)	このメソッドは MediaTerminal を登録します。CiscoMediaTerminal は CiscoTerminal.UNREGISTERED 状態になっている必要があり、プロバイダーは Provider.IN_SERVICE 状態になっている必要があります。 MediaTerminal が登録されると、このメソッドは正常に終了します。このメソッドでは、アプリケーションが CTIManager で TLS リンクを確立し、Cisco Unified Communications Manager Administration でユーザに対して SRTP Enabled フラグを有効にする必要があります。そうしないと、PrivilegeViolationException が送出されます。 **パラメータ** • address：この端末の着信 IPv4 RTP ストリームのインターネット アドレス • port : RTP ストリームの受信に使用されるこの端末の UDP ポート • capabilities：この端末でサポートされている RTP 符号化方式のリスト • AlgorithmIDs ：この CTIPort がサポートする SRTP アルゴリズム AlgorithmIDs は、CiscoMediaEncryptionAlgorithmType のいずれかにする必要があります。 **例外** CiscoRegistrationException javax.telephony.PrivilegeViolationException

表 6-94　Cisco*MediaTerminal* のメソッド （続き）

インターフェイス	メソッド	説明
void	register(java.net.InetAddress address, int port, CiscoMediaCapability[] capabilities, int[] algorithmIDs, java.net.InetAddress address_v6, int activeAddressingMode)	CiscoMediaTerminal は CiscoTerminal.UNREGISTERED 状態になっている必要があり、プロバイダーは Provider.IN_SERVICE 状態になっている必要があります。 このメソッドが成功した場合は、MediaTerminal が登録されます。activeAddressingMode はアプリケーション IP アドレッシング機能を示します。アプリケーションが activeAddressingMode を CiscoTerminal.IP_ADDRESSING_MODE_IPv4 に指定する場合、address も指定する必要があります。 アプリケーションが activeAddressingMode を CiscoTerminal.IP_ADDRESSING_MODE_IPv6 に指定する場合、address_v6 も指定する必要があります。 アプリケーションが activeAddressingMode を CiscoTerminal.IP_ADDRESSING_MODE_IPv4_6 に指定する場合、address および address_v6 も指定する必要があります。 **メソッドの引数** このメソッドには 4 種類の引数があります。 • 最初の引数は、この端末の RTP メディア ストリームが終端されるインターネット アドレスを指定します。 • 2 番目の引数は RTP パケットが向けられる UDP ポートを指定します。 • 3 番目の引数は、アプリケーションがこの端末に対してサポートする RTP 符号化方式の種類を示します。 • 4 番目の引数は、アプリケーションがサポートする SRTP アルゴリズムを示します。 このメソッドは、アプリケーションが CTIManager で TLS リンクを確立している場合と、アプリケーションがユーザに対して CM の管理ページで SRTP Enabled フラグを有効にしている場合にだけ使用できます。そうでない場合は、PrivilegeViolationException が送出されます。 **メソッドの事後条件** MediaTerminal が登録されると、このメソッドは正常に終了します。

表 6-94　Cisco*MediaTerminal* のメソッド（続き）

インターフェイス	メソッド	説明
		パラメータ • address：この端末の着信 IPv4 RTP ストリームのインターネット アドレス。アプリケーションのアドレッシング モードに応じて、null になる場合もあります。 • port : RTP ストリームの受信に使用されるこの端末の UDP ポート • capabilities：この端末でサポートされている RTP 符号化方式のリスト • AlgorithmIDs ：この CTIPort がサポートする SRTP アルゴリズム AlgorithmIDs は、CiscoMediaEncryptionAlgorithmType のいずれか 1 つだけになります。 • address_v6：この端末の着信 IPv6 RTP ストリームのインターネット アドレス。activeAddressingMode に応じて、null になる場合もあります。 • activeAddressingMode：アプリケーションがこの CiscoMediaTerminal を登録する IP アドレッシング モード 次のいずれかになります。 • CiscoTerminal.IP_ADDRESSING_MODE_IPv4 • CiscoTerminal.IP_ADDRESSING_MODE_IPv6 • CiscoTerminal.IP_ADDRESSING_MODE_IPv4z_v6（**7.0 以降**） **例外** CiscoRegistrationException, javax.telephony.PrivilegeViolationException

表 6-94　Cisco*MediaTerminal* のメソッド （続き）

インターフェイス	メソッド	説明
void	register(CiscoMediaCapability[] capabilities)	このメソッドは、指定された CiscoMediaCapabilities で MediaTerminal を登録します。このメソッドは、各コールに対して IP アドレスおよびポートを動的に指定する場合にだけ使用します。 このメソッドで登録するアプリケーションは、コールごとに CiscoMediaOpenLogicalChannelEv を受信するので、このオブジェクトの setRTPParams メソッドを使用して、IP アドレスおよびポート番号を提示する必要があります。 CiscoMediaTerminal は CiscoTerminal.UNREGISTERED 状態になっている必要があり、プロバイダーは Provider.IN_SERVICE 状態になっている必要があります。 **メソッドの引数** 引数はアプリケーションがこの Terminal に対してサポートする RTP 符号化方式の種類を示します。 **メソッドの事後条件** CiscoMediaTerminal が登録されると、このメソッドは正常に終了します。 **パラメータ** capabilities：この端末でサポートされている RTP 符号化方式のリスト **例外** CiscoRegistrationException

表 6-94　Cisco*MediaTerminal* のメソッド （続き）

インターフェイス	メソッド	説明
void	register(CiscoMediaCapability[] capabilities, int[] algorithmIDs)	このメソッドでは、指定された CiscoMediaCapabilities およびサポートされる SRTP アルゴリズムで MediaTerminal を登録します。 このメソッドは、各コールに対して IP アドレスおよびポートを動的に指定し、SRTP アルゴリズムも指定する場合にだけ使用します。 このメソッドで登録するアプリケーションは、コールごとに CiscoMediaOpenLogicalChannelEv を受信するので、このオブジェクトの setRTPParams メソッドを使用して、IP アドレスおよびポート番号を提示する必要があります。 この形式の register() には、アプリケーションがサポートする SRTP アルゴリズムを指定する 2 番目のパラメータも必要です。 このメソッドでは、アプリケーションが CTIManager で TLS リンクを確立し、Cisco Unified Communications Manager Administration でユーザに対して SRTP Enabled フラグを有効にする必要があります。そうしないと、PrivilegeViolationException が送出されます。 CiscoMediaTerminal が登録されると、このメソッドは正常に終了します。 CiscoMediaTerminal は CiscoTerminal.UNREGISTERED 状態になっている必要があり、プロバイダーは Provider.IN_SERVICE 状態になっている必要があります。 **パラメータ** • capabilities：この端末でサポートされている RTP 符号化方式のリスト • algorithmIDs：この端末でサポートされている RTP アルゴリズムのリスト AlgorithmIDs は、CiscoMediaEncryptionAlgorithmType のいずれかにする必要があります。 **例外** CiscoRegistrationException javax.telephony.PrivilegeViolationException

表 6-94　Cisco*MediaTerminal* のメソッド （続き）

インターフェイス	メソッド	説明
void	register(CiscoMediaCapability[] capabilities, int[] algorithmIDs, int activeAddressingMode)	CiscoMediaTerminal は CiscoTerminal.UNREGISTERED 状態になっている必要があり、プロバイダーは Provider.IN_SERVICE 状態になっている必要があります。このメソッドが成功した場合は、MediaTerminal が登録されます。指定された CiscoMediaCapabilities およびサポートされる SRTP アルゴリズムで端末を登録します。また、これは、アプリケーションが各コールに対して動的に ipAddress およびポートを指定することを示します。 このメソッドを登録するアプリケーションは、各コールに対して CiscoMediaOpenLogicalChannelEv を受信し、このオブジェクトに対する setRTPParams メソッドを使用して ipAddress とポート番号を指定する必要があります。 2 番目の引数は、アプリケーションがサポートする SRTP アルゴリズムを示します。このメソッドは、アプリケーションが CTIManager で TLS リンクを確立している場合と、アプリケーションがユーザに対して Cisco Unified Communications Manager Administration で SRTP Enabled フラグを有効にしている場合にだけ使用できます。そうでない場合は、PrivilegeViolationException が送出されます。 **メソッドの引数** 引数は、アプリケーションがこの端末に対してサポートする RTP 符号化の種類と、アプリケーションまたは CTIManager の障害を示します。 **メソッドの事後条件** CiscoMediaTerminal が登録されると、このメソッドは正常に終了します。 **パラメータ** • capabilities：この端末でサポートされている RTP 符号化方式のリスト • algorithmIDs：この端末でサポートされている RTP アルゴリズムのリスト AlgorithmIDs は、CiscoMediaEncryptionAlgorithmType のいずれか 1 つだけになります。 • activeAddressingMode：アプリケーションがこの CiscoMediaTerminal を登録する IP アドレッシングモード activeAddressingMode は次のいずれかです。 – CiscoTerminal.IP_ADDRESSING_MODE_IPv4 – CiscoTerminal.IP_ADDRESSING_MODE_IPv6 – CiscoTerminal.IP_ADDRESSING_MODE_IPv4_v6 **例外** CiscoRegistrationException javax.telephony.PrivilegeViolationException

表 6-94　Cisco*MediaTerminal* のメソッド （続き）

インターフェイス	メソッド	説明
void	setRTPParams(CiscoRTPHandle rtpHandle, CiscoRTPParams rtpParams)	アプリケーションは、IP アドレスおよび RTP ポート番号を設定して、コールのメディア ストリームを動的に行う場合にだけこのメソッドを使用します。この場合、機能だけを指定して MediaTerminal または CiscoRouteTeminal を登録する必要があります。 アプリケーションでは、terminalObserver で CiscoCallOpenLogicalChannel を受信したときに、このメソッドを起動する必要があります。アプリケーションは、CiscoCallOpenLogicalChannelEv で受信する rtpHandle を渡す必要があります。アプリケーションは、CiscoProvider.getRTPHandle(rtpHandle) メソッドを呼び出して CiscoCall 参照を取得できます。 これは、コール オブザーバが端末に追加されていない場合、このイベントの送信時にコール オブザーバがなかった場合、またはこのハンドルに関連付けられたコールがない場合には、null を返します。 **パラメータ** • rtpHandle：CiscoMediaCallOpenLogicalChannelEv から取得します。 • rtpParams：CiscoRTPParams のタイプで、コールごとにメディア端末の動的 RTP アドレスとポート番号を指定するために使用されます。 **例外** javax.telephony.InvalidStateException javax.telephony.InvalidArgumentException javax.telephony.PrivilegeViolationException
void	unregister()	このメソッドは MediaTerminal を登録解除し、MediaTerminal が登録解除されると、このメソッドは正常に終了します。CiscoMediaTerminal は登録されている必要があり、プロバイダーは Provider.IN_SERVICE 状態になっている必要があります。 **例外** CiscoUnregistrationException
boolean	isRegistered()	CiscoMediaTerminal が登録されている場合は true、そうでない場合は false を返します。MediaTerminal の場合、このメソッドは MediaTerminal が InService 状態の場合は true、OutOfService の場合は false を返します。CTIManager の障害の場合は、false を返します。

表 6-94　Cisco*MediaTerminal* のメソッド （続き）

インターフェイス	メソッド	説明
boolean	isRegisteredByThisApp()	このメソッドは、このアプリケーションが登録要求を発行して成功した場合に true を返します。この登録は、CTIManager の障害によりデバイスがアウトオブサービス状態にある場合でも有効です。これは、このアプリケーションがデバイスの登録を解除するまで true に設定されます。
int	getIPAddressingMode()	アプリケーションはこの API を起動して CiscoMediaTerminal の IP アドレッシング モードを照会できます。アドレッシング モードを表す定数は次のとおりです。 • CiscoTerminal.IP_ADDRESSING_IPv4 • CiscoTerminal.IP_ADDRESSING_IPv6 • CiscoTerminal.IP_ADDRESSING_IPv4_v6

継承したメソッド

インターフェイス com.cisco.jtapi.extensions.CiscoTerminal から
createSnapshot, getAltScript, getDeviceState, getDNDOption, getDNDStatus, getEMLoginUsername, getFilter, getLocale, getProtocol, getRegistrationState, getRTPInputProperties, getRTPOutputProperties, getState, getSupportedEncoding, isRestricted, sendData, sendData, setDNDStatus, setFilter, unPark

インターフェイス javax.telephony.Terminal から
addCallObserver, addObserver, getAddresses, getCallObservers, getCapabilities, getName, getObservers, getProvider, getTerminalCapabilities, getTerminalConnections, removeCallObserver, removeObserver

インターフェイス com.cisco.jtapi.extensions.CiscoObjectContainer から
getObject, setObject

関連資料

CiscoTerminal と CiscoMediaOpenLogicalChannelEv.CiscoRTPParams を参照してください。

CiscoMonitorInitiatorInfo

このインターフェイスは、モニタリングの開始側に関する情報を定義します。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

宣言

public interface CiscoMonitorInitiatorInfo

フィールド

なし

メソッド

表 6-95　CiscoMonitorInitiatorInfo のメソッド

インターフェイス	メソッド	説明
CiscoAddress	getAddress()	モニタリングの開始側のアドレスを返します。
Int	getMonitorInitiatorCallLegHandle()	モニタリングの開始側のコール レッグのハンドルを返します。JTAPI では、provider.getCall (int monitorInitiatorCallLegHandle) を使用してモニタリング ターゲットのコールを取得します。 モニタリングの開始側のコールがこのプロバイダーでアクティブでない場合、このメソッドは null を返します。
java.lang.String	getTerminalName()	モニタリングの開始側の端末名を返します。

関連資料

なし

CiscoMonitorTargetInfo

このインターフェイスは、モニタリングのターゲットに関する情報を提供します。

宣言

public interface CiscoMonitorTargetInfo

フィールド

なし

メソッド

表 6-96　CiscoMonitorTargetInfo のメソッド

インターフェイス	メソッド	説明
CiscoAddress	getAddress()	モニタリングのターゲットのアドレスを返します。
Int	getMonitorTargetCallLegHandle()	モニタリングのターゲットのコール レッグのハンドルを返します。
java.lang.String	getTerminalName()	モニタリングのターゲットの端末名を返します。

関連資料

なし

CiscoObjectContainer

アプリケーションは ApplicationObject インターフェイスを使用して、アプリケーション定義オブジェクトを、このインターフェイスを実装するオブジェクトに関連付けることができます。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

サブインターフェイス

CiscoAddress, CiscoCall, CiscoCallID, CiscoConnection, CiscoConnectionID, CiscoConsultCall, CiscoIntercomAddress, CiscoJtapiPeer, CiscoMediaTerminal, CiscoProvider, CiscoRouteTerminal, CiscoTerminal, CiscoTerminalConnection

宣言

public interface CiscoObjectContainer

フィールド

なし

メソッド

表 6-97　CiscoObjectContainer のメソッド

インターフェイス	メソッド	説明
java.lang.Object	getObject()	アプリケーション定義オブジェクトを取得します。
java.lang.Object	setObject(java.lang.Object reference)	アプリケーション定義オブジェクトを設定します。

関連資料

なし

CiscoOutOfServiceEv

CiscoOutOfServiceEv イベントは、アウトオブサービス イベント CiscoAddrOutOfServiceEv および CiscoTermOutOfServiceEv のスーパー クラスです。このクラスはアウトオブサービス イベントの原因を定義します。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

CiscoEv, javax.telephony.events.Ev

サブインターフェイス

CiscoAddrOutOfServiceEv または CiscoTermOutOfServiceEv

宣言

public interface CiscoOutOfServiceEv extends CiscoEv

フィールド

表 6-98　CiscoOutOfServiceEv のフィールド

インターフェイス	フィールド	説明
static int	CAUSE_CALLMANAGER_FAILURE	このイベントの原因は Cisco Unified Communications Manager の障害です。
static int	CAUSE_CTIMANAGER_FAILURE	このイベントの原因は CTIManager の障害です。
static int	CAUSE_DEVICE_FAILURE	このイベントの原因はデバイスの障害です。
static int	CAUSE_DEVICE_RESTRICTED	このイベントの原因はデバイスが制限されていることです。
static int	CAUSE_DEVICE_UNREGISTERED	このイベントの原因はデバイスが UNREGISTERED 状態であることです。
static int	CAUSE_LINE_RESTRICTED	このイベントの原因は回線が制限されていることです。
static int	CAUSE_NOCALLMANAGER_AVAILABLE	このイベントの原因は Cisco Unified Communications Manager の機能を利用できないことです。
static int	CAUSE_REHOME_TO_HIGHER_PRIORITY_CM	このイベントの原因は、ハイ プライオリティの Cisco Unified Communications Manager ノードへのフェールバックにおけるエラーです。
static int	CAUSE_REHOMING_FAILURE	このイベントの原因は rehome の試行中の障害です。
static int	ID	—

継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

メソッド

なし

関連資料

「定数フィールド値」（P.F-1）を参照してください。

CiscoPartyInfo

このインターフェイスはコールの通話者情報を定義します。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

宣言

public interface CiscoPartyInfo

フィールド

表 6-99　CiscoPartyInfo のフィールド

インターフェイス	フィールド	説明
Static int	ABBREVIATED_NUMBER	この NumberType は 4 と同じで、発信者が Cisco Unified Communications Manager サーバと同じであることを示します。
Static int	INTERNATIONAL_NUMBER	この NumberType は 0 と同じで、何も設定されていないことを示します。
Static int	NATIONAL_NUMBER	この NumberType は 1 と同じで、発信者が INTERNATIONAL であることを示します。
Static int	NET_SPECIFIC_NUMBER	この NumberType は 2 と同じで、発信者が NATIONAL であることを示します。
Static int	RESERVED_FOR_EXTENSION	この NumberType は 6 と同じで、ファースト ダイヤル コールを示します。現在は使用されていません。
Static int	SUBSCRIBER_NUMBER	この NumberType は 3 と同じで、発信側が MGCP/H.323 ゲートウェイであることを示します。
Static int	UNKNOWN_NUMBER	—

メソッド

表 6-100　CiscoPartyInfo のメソッド

インターフェイス	メソッド	説明
javax.telephony.Address	getAddress()	アドレスを返します。
boolean	getAddressPI()	アドレスに関連付けられた Presentation Indicator（PI）を返します。true が返された場合、Application はこの Address Name をエンド ユーザに表示できます。false が返された場合、Application はこの Address Name をエンド ユーザに表示しません。
java.lang.String	getDisplayName()	通話者の表示名を返します。
boolean	getDisplayNamePI()	DisplayName に関連付けられた PI を返します。true が返された場合、Application はこの DisplayName をエンド ユーザに表示できます。false が返された場合、Application はこの DisplayName をエンド ユーザに表示しません。
int	getlocale()	通話者の Unicode 表示名のロケールを返します。
int	getNumberType()	通話者の番号種別を返します。
java.lang.String	getUnicodeDisplayName()	通話者の Unicode 表示名を返します。
CiscoUrlInfo	getUrlInfo()	URL 情報を返します。
java.lang.String	getVoiceMailbox()	通話者のボイス メールボックスを返します。

関連資料

「定数フィールド値」（P.F-1）を参照してください。

CiscoProvCallParkEv

CiscoProvCallParkEv イベントは、クラスタ内のいずれかのデバイスでコールがパークされたとき、またはパーク解除されたときにプロバイダー オブザーバに配信されます。このイベントを受信するには、CiscoProvider.registerFeature() および CiscoProvFeatureID.MONITOR_CALLPARK_DN を使用して登録を行う必要があります。また、このイベントを受信するには、アプリケーションが使用するユーザ プロファイルで Call Park Retrieval Allowed フラグが有効になっている必要があります。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

CiscoEv, CiscoProvEv, CiscoProvFeatureEv, javax.telephony.events.Ev, javax.telephony.events.ProvEv

宣言

public interface CiscoProvCallParkEv extends CiscoProvFeatureEv

フィールド

表 6-101　CiscoProvCallParkEv のフィールド

インターフェイス	フィールド	説明
Static int	ID	—
Static int	PARK_STATE_ACTIVE	コールがパークされたことを示します。
static int	PARK_STATE_IDLE	コールのパークが解除されたことを示します。
static int	REASON_CALLPARK	コールがパークされると、このイベントが発生することを示します。
static int	REASON_CALLPARKREMAINDER	**推奨されません。** このインターフェイスはスペル ミスのために非推奨になりました。新しいインターフェイス REASON_CALLPARKREMINDER を使用してください。
static int	REASON_CALLPARKREMINDER	コール パーク リマインダの後に、パークしている通話者にコールが戻されることを示します。
static int	REASON_CALLUNPARK	コールのパークが解除されたことを示します。

継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

表 6-102　CiscoProvCallParkEv のメソッド

インターフェイス	メソッド	説明
int	getintCallIDValue()	このオブジェクトの整数表現を返します。
java.lang.String	getParkDN()	コールがパークされている場所を返します。
java.lang.String	getParkedParty()	パークされた通話者の DN を返します。
java.lang.String	getParkedPartyPartition()	パークされた通話者のパーティションを返します。
java.lang.String	getParkingParty()	パークしている通話者の DN を返します。
java.lang.String	getParkingPartyPartition()	パークしている通話者のパーティションを返します。
java.lang.String	getParkPartyPartition()	パーク DN のパーティションを返します。
int	getReason()	イベントの原因を返します。
int	getState()	コールの状態を返します。Possible states are CiscoProvCallParkEv.PARK_STATE_IDLE CiscoProvCallParkEv.PARK_STATE_ACTIVE.

継承したメソッド

インターフェイス com.cisco.jtapi.extensions.CiscoProvFeatureEv から
getFeatureID()

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.ProvEv から
getProvider

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

「定数フィールド値」（P.F-1）を参照してください。

CiscoProvEv

JTAPI のコアである javax.telephony.events.ProvEv インターフェイスを拡張する CiscoProvEv インターフェイスは、Cisco によって拡張されたすべての JTAPI Provider イベントの基本インターフェイスになります。このパッケージのプロバイダー関連イベントはすべて、直接的または間接的にこのインターフェイスを拡張します。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

CiscoEv, javax.telephony.events.Ev, javax.telephony.events.ProvEv

サブインターフェイス

CiscoAddrActivatedEv, CiscoAddrActivatedOnTerminalEv, CiscoAddrAddedToTerminalEv, CiscoAddrCreatedEv, CiscoAddrRemovedEv, CiscoAddrRemovedFromTerminalEv, CiscoAddrRestrictedEv, CiscoAddrRestrictedOnTerminalEv, CiscoProvCallParkEv, CiscoProvFeatureEv, CiscoProvTerminalCapabilityChangedEv, CiscoRestrictedEv, CiscoTermActivatedEv, CiscoTermCreatedEv, CiscoTermRemovedEv, CiscoTermRestrictedEv

宣言

public interface CiscoProvEv extends CiscoEv, javax.telephony.events.ProvEv

フィールド

なし

継承したフィールド

インターフェイス javax.telephony.events.Ev から

CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY,

META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

なし

継承したメソッド

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.ProvEv から
getProvider

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

CiscoProvFeatureEv

CiscoProvFeatureEv インターフェイスは、プロバイダー イベントのために com.cisco.jtapi.extensions.CiscoProvEv interface インターフェイスを拡張したものです。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

CiscoEv, CiscoProvEv, javax.telephony.events.Ev, javax.telephony.events.ProvEv

サブインターフェイス

CiscoProvCallParkEv

宣言

public interface CiscoProvFeatureEv extends CiscoProvEv

フィールド

なし

継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

表 6-103　CiscoProvFeatureEv のメソッド

インターフェイス	メソッド	説明
int	getFeatureID()	アプリケーションがイベントを受け取る機能 ID。

継承したメソッド

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.ProvEv から
getProvider

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

CiscoProvEv を参照してください。

ProvEv.

CiscoProvFeatureID

このインターフェイスは、registerFeature インターフェイスでサポートされている機能をリストします。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

宣言

public interface CiscoProvFeatureID

フィールド

表 6-104　CiscoProvFeatureID のフィールド

インターフェイス	フィールド	説明
static int	MONITOR_CALLPARK_DN	クラスタ内のいずれかのデバイスでコールがパークまたはパーク解除されたときに、CiscoProvider の registerFeature インターフェイスで CiscoProvCallParkEv を受信するのに使用します。

メソッド

なし

関連資料

「定数フィールド値」（P.F-1）を参照してください。

CiscoProvider

CiscoProvider インターフェイスは、Cisco 固有の機能を追加することによって CiscoProvider インターフェイスを拡張します。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

CiscoObjectContainer, javax.telephony.Provider

宣言

public interface CiscoProvider extends javax.telephony.Provider, CiscoObjectContainer

フィールド

なし

継承したフィールド

インターフェイス javax.telephony.Provider から
IN_SERVICE, OUT_OF_SERVICE, SHUTDOWN

メソッド

表 6-105　CiscoProvider のメソッド

インターフェイス	メソッド	説明
CiscoTerminal	createTerminal(java.lang.String name)	指定された名前に対応する CiscoTerminal クラスのインスタンスを返します。アプリケーションが十分な機能を備えている必要があります。そうでない場合は、PrivilegeViolationException で CiscoProvider.createTerminal() がスローされます。 **事前条件** this.getState() == Provider.IN_SERVICE **事後条件** Create CiscoTerminal と名前が対応している端末は this.getTerminals() の要素です。 **パラメータ** • name：対象となる CiscoTerminal オブジェクトの名前 **例外** javax.telephony.InvalidArgumentException：指定された名前が、プロバイダーまたはプロバイダーのドメイン内で確認されたどの CiscoMediaTerminal の名前とも対応しません。 javax.telephony.InvalidStateException：プロバイダーがイン サービスではありません。 PreviledgeVoilationException：プロバイダーに十分な機能がありません。CiscoProviderCapabilities.canObserveAnyTerminal() returns false call.getState() == Call.INVALID

表 6-105　CiscoProvider のメソッド （続き）

インターフェイス	メソッド	説明
void	deleteTerminal(CiscoTerminal terminal)	プロバイダー制御から CiscoTerminal オブジェクトを削除します。プロバイダー制御から CiscoTerminal オブジェクトを削除します。アプリケーションが Provider.createTerminal() インターフェイスを使用してこの端末を作成する必要があります。そうでない場合、PreviledgeVoilationException がスローされます。CiscoProvider.deleteTerminal(). **事前条件** this.getState() == Provider.IN_SERVICE **事後条件** 端末のプロバイダー リストから削除された CiscoTerminal オブジェクト。端末は this.getTerminals() の要素ではなくなり、端末に属しているアドレスがが削除されます。 **パラメータ** • terminal：削除する CiscoTerminal オブジェクトへの参照。 **例外** javax.telephony.InvalidArgumentException：提供される端末が this.getTerminals() の要素ではないか、または端末がプロバイダー ドメインではありません。 PrivilegeViolationException：引数で指定された端末が Provider.createTerminal() メソッドを使用して作成された端末ではありません。アプリケーションが削除できるのは、Provider.createTerminal() インターフェイスを使用して作成される端末だけです。
javax.telephony.Address	getAddress(java.lang.String number, java.lang.String partition)	このメソッドで渡される番号とパーティションに対応するアドレス オブジェクトを返します。アドレス オブジェクトは、特定の番号とパーティションの一意の組み合わせです。 **例外** javax.telephony.InvalidArgumentException

表 6-105　CiscoProvider のメソッド （続き）

インターフェイス	メソッド	説明
int	getAppDSCPValue()	CiscoProvider.getAppDSCPValue() を使用して、プロバイダーから DSCP 値を取得します。 **事前条件** this.getState() == Provider.IN_SERVICE **事後条件** このメソッドは、CTI で設定されたアプリケーションの DSCP 値の整数値を返します。
CiscoCall	getCall(CiscoRTPHandle rtpHandle)	特定の端末に関連付けられた RTPHandle のコール オブジェクトを返します。
CiscoCall	getCall(int callleg)	プロバイダー ドメイン内に存在する CiscoCall と、特定の端末に関連付けられた RTPHandle を持つコール オブジェクトを返します。このメソッドは、RTPHandle がどのコールにも関連付けられていない状態になった場合や、このハンドルを含んだ CiscoCallOpenLogicalChannelEv がアプリケーションに送信されたときに端末に callObserver が追加されていなかった場合に、null を返す可能性があります。 **例外** javax.telephony.InvalidStateException
boolean	getCallbackGuardEnabled()	なし
CiscoIntercomAddress[]	getIntercomAddresses()	プロバイダー ドメインに存在している CiscoInterComAddress の配列を返します。

表 6-105　CiscoProvider のメソッド （続き）

インターフェイス	メソッド	説明
CiscoMediaTerminal	getMediaTerminal(java.lang.String name)	指定された名前に対応する CiscoMediaTerminal クラスのインスタンスを返します。各 CiscoMediaTerminal には、JTAPI 実装により固有の名前が割り当てられています。 指定された名前に対する CiscoMediaTerminal がプロバイダーのドメイン内にない場合、このメソッドは、InvalidArgumentException をスローします。 この CiscoMediaTerminal は、Provider.getTerminals() および CiscoProvider.getMediaTerminals() が生成する配列に含まれています。 **事前条件** CiscoMediaTerminal terminal = this.getMediaTerminal(name) を行う。端末は this.getTerminals() の要素です。端末は this.getMediaTerminals() の要素です。 **事後条件** CiscoMediaTerminal terminal = this.getMediaTerminal(name) を行う。端末は this.getTerminals() の要素です。端末は this.getMediaTerminals() の要素です。 **パラメータ** • name：対象となる CiscoMediaTerminal オブジェクトの名前 **例外** javax.telephony.InvalidArgumentException：指定された名前が、プロバイダーまたはプロバイダーのドメイン内で確認されたどの CiscoMediaTerminal の名前とも対応しません。

表 6-105　CiscoProvider のメソッド （続き）

インターフェイス	メソッド	説明
CiscoMediaTerminal[]	getMediaTerminals()	プロバイダーおよびプロバイダーのローカル ドメインに関連付けられた CiscoMediaTerminal の配列を返します。 各 CiscoMediaTerminal には、JTAPI 実装により固有の名前が割り当てられています。 このプロバイダーに関連付けられた CiscoMediaTerminal がない場合、このメソッドは null を返します。 この配列は、Provider.getTerminals() によって返された配列のサブセットです。 **事後条件** CiscoMediaTerminal[] terminals = this.getMediaTerminals() terminals == null または terminals.length >= 1 if terminals != null, terminals is a subset of this.getTerminals () を行う。 **例外** javax.telephony.ResourceUnavailableException : プロバイダーにあるメディア端末の数が、静的配列として返すには大きすぎることを示します。
java.lang.String	getVersion()	なし
void	registerFeature(int featureID)	アプリケーションがプロバイダー イベントを取得する特定の機能を登録します。アプリケーションは、ソフトキーの featureID を渡す必要があります。CiscoProvFeatureID インターフェイスに現在サポートされている機能がリストされています。 **例外** javax.telephony.InvalidStateException javax.telephony.PrivilegeViolationException javax.telephony.InvalidArgumentException

表 6-105　CiscoProvider のメソッド （続き）

インターフェイス	メソッド	説明

Void	setCallbackGuardEnabled(boolean enabled)	オブザーバ コールバックに対して try/catch ロジックを有効または無効にします。オブザーバ コールバックのアプリケーション例外からプロバイダー自体を保護するために、プロバイダーは、通常、アプリケーション インターフェイス（たとえば observer）のすべての呼び出しを、次のコードでガードします。
		`try {`
		`observer.callStateChanged ( ... );`
		`} catch ( Throwable t ) {`
		`// log the exception here`
		`}`
		
		これにより、JTAPI 実装からアプリケーション エラーが隔離されます。JTAPI 実装が処理されない例外を確認し、処理を続行できるため、トラブルシューティングが容易になります。
		
		エラーによっては、回復不能と判断され、JTAPI によって再度スローされます。このため、通常、アプリケーションが終了します。このようなエラーには、ThreadDeath、OutOfMemoryError、および StackOverflowError があります。
		
		JTAPI スレッド内でエラーをトラップする必要があるアプリケーションでは、ThreadGroup のサブクラスを作成し、その ThreadGroup 内のスレッドから JTAPI を初期化する必要があります。
		
		ThreadGroup.uncaughtException () メソッドをオーバーライドすることで、JTAPI スレッドでスローされたすべての回復不能エラーを、アプリケーションで検知可能にできます。場合によっては、JTAPI で積極的にエラーを捕捉することによって、java デバッガでのアプリケーションのトラブルシューティングがより複雑になることがあります。
		
		たとえば、Microsoft Visual J++ バージョン 6.0 では、JTAPI が Throwable を捕捉した場合、アプリケーションのオブザーバ コールバック内のブレークポイントが適正に処理されません。このような場合、JTAPI アプリケーション開発者は、内部の JTAPI try/catch ロジックを無効にすることもできます。

表 6-105　CiscoProvider のメソッド （続き）

インターフェイス	メソッド	説明
		（注） このコールバック ガードを無効にする方法は、アプリケーションのトラブルシューティングを行う場合にだけ使用します。実稼動環境では使用しないでください。デフォルトでは、コールバック ガードは、常に有効にされています。 **パラメータ** • enabled：true の場合、コールバック ガードが有効で、false の場合、コールバック ガードが無効です。
Void	unregisterFeature(int featureID)	特定の機能の登録が解除されます。

継承したメソッド

インターフェイス javax.telephony.Provider から
addObserver, createCall, getAddress, getAddressCapabilities, getAddressCapabilities, getAddresses, getCallCapabilities, getCallCapabilities, getCalls, getCapabilities, getConnectionCapabilities, getConnectionCapabilities, getName, getObservers, getProviderCapabilities, getProviderCapabilities, getState, getTerminal, getTerminalCapabilities, getTerminalCapabilities, getTerminalConnectionCapabilities, getTerminalConnectionCapabilities, getTerminals, removeObserver, shutdown

インターフェイス com.cisco.jtapi.extensions.CiscoObjectContainer から
getObject, setObject

関連資料

なし

CiscoProviderCapabilities

このインターフェイスは、Cisco Unified JTAPI 実装で提供される特定の機能を定義しています。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	メソッド canSupportIPv6()T のサポートを追加します。

スーパーインターフェイス

javax.telephony.capabilities.ProviderCapabilities

宣言

public interface CiscoProviderCapabilities extends javax.telephony.capabilities.ProviderCapabilities

メソッド

表 6-106　CiscoProviderCapabilities のメソッド

インターフェイス	メソッド	説明

boolean	canObserveAnyTerminal()	このメソッドは、ユーザが Cisco Unified Communications Manager でシステムの端末（およびそのアドレス）を監視するための権限をプロビジョニングしたかどうかを確認します。このような端末やアドレスは、JTAPI が起動時に初期化するリストの一部として返されません。このような権限を持つユーザのログインで取得されるプロバイダーは、ProviderCapabilities での canObserverAnyTerminal メソッドから判断できます。ユーザがシステム内の端末を監視できる場合は true を返し、ユーザが制御リスト内の端末およびアドレスだけを監視できる場合は false を返します。
		
		例
		`Provider p = peer.getProvider( loginString );`
		`ProviderCapabilities caps = p.getCapabilities ();`
		`if ( caps instanceof CiscoProviderCapabilities )`
		`{`
		`boolean canObserveAnyTerminal =`
		`((CiscoProviderCapabilities)caps).canObserveAny`
		`Terminal ();`
		`boolean canMonitorParkDN =`
		`((CiscoProviderCapabilities)caps).canMonitorParkDNs`
		`();`
		`boolean canModifyCallingPN=`
		`((CiscoProviderCapabilities)caps).canModifyCalling`
		`Party ();`
		`boolean canRecordCalls =`
		`((CiscoProviderCapabilities)caps).canRecord();`
		`boolean canMonitorCalls =`
		`((CiscoProviderCapabilities)caps).canMonitor();`
		`}`

boolean	canMonitorParkDNs()	このメソッドは、ユーザが Cisco Unified Communications Manager でパーク DN を監視するようにプロビジョニングしたかどうかを確認します。ユーザがパーク DN を監視できる場合は true、そうでない場合は false を返します。

表 6-106　CiscoProviderCapabilities のメソッド （続き）

インターフェイス	メソッド	説明
boolean	canModifyCallingParty()	このメソッドは、ユーザが Cisco Unified Communications Manager でコールの発信側番号を変更するようにプロビジョニングしたかどうかを確認します。ユーザが発信側番号を変更できる場合は true、そうでない場合は false を返します。
boolean	canRecord()	このメソッドは、ユーザが Cisco Unified Communications Manager でコールを録音するようにプロビジョニングしたかどうかを確認します。「Standard CTI Allow Call Recording」ユーザ グループ内のユーザだけがコールを録音できます。ユーザがこのグループに属している場合は True を返します。
boolean	canMonitor()	このメソッドは、ユーザが Cisco Unified Communications Manager でコールをモニタリングするようにプロビジョニングしたかどうかを確認します。「Standard CTI Allow Call Monitoring」ユーザ グループ内のユーザだけがコールのモニタリング要求を開始できます。ユーザがこのグループに属している場合は True を返します。
boolean	canSupportIPv6()	このインターフェイスは、エンタープライズ パラメータ「Enable IPv6」が有効になっている場合は true、そうでない場合は false を返します。

継承したメソッド

インターフェイス javax.telephony.capabilities.ProviderCapabilities から
isObservable

関連資料

canObserveAnyTerminal() を参照してください。

CiscoProviderCapabilityChangedEv

アプリケーション プロバイダー オブザーバは、Cisco Unified Communications Manager でユーザ グループ（capabilitied）にユーザが追加または削除されると、このイベントを受け取ります。このイベントのメソッドでは、変更された機能を確認できます。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	追加された hasIPv6CapabilityChanged() メソッド。

宣言

public interface CiscoProviderCapabilityChangedEv

フィールド

表 6-107　CiscoProviderCapabilityChangedEv のフィールド

インターフェイス	フィールド	説明
static int	ID	なし
static int	MODIFY_CGPN	**推奨されません。** この定数はどのインターフェイスからも返されません。アプリケーションでは使用しないでください。
static int	MONITOR_PARKDN	**推奨されません。** この定数はどのインターフェイスからも返されません。アプリケーションでは使用しないでください。
static int	SUPERPROVIDER	**推奨されません。** この定数はどのインターフェイスからも返されません。アプリケーションでは使用しないでください。

メソッド

表 6-108　CiscoProviderCapabilityChangedEv のメソッド

インターフェイス	メソッド	説明
CiscoProviderCapabilities	getCapability()	このメソッドは、ユーザの現在の CiscoProviderCapabilities オブジェクトを返します。

表 6-108　CiscoProviderCapabilityChangedEv のメソッド （続き）

インターフェイス	メソッド	説明
boolean	hasIPv6CapabilityChanged()	このメソッドは、「Enable IPv6」エンタープライズパラメータが変更されたかどうかを判断するために使用できます。 **事前条件** this.getState() == Provider.IN_SERVICE **事後条件** このメソッドは、「Enable IPv6」エンタープライズパラメータが変更された場合に True を返し、そうでない場合は False を返します。
boolean	hasModifyCallingPartyChanged()	このメソッドは、「modify Calling Party」権限が変更されたかどうかを確認します。 **事前条件** provider.getState() == Provider.IN_SERVICE
boolean	hasMonitorCapabilityChanged()	このメソッドは、ユーザのモニタリング機能が変更されたかどうかを確認します。 **事前条件** provider.getState() == Provider.IN_SERVICE
boolean	hasMonitorParkDNChanged()	このメソッドは、「monitor Park DN」権限が変更されたかどうかを確認します。 **事前条件** provider.getState() == Provider.IN_SERVICE
boolean	hasObserveAnyTerminalChanged()	このメソッドは、「can control any terminal」権限が変更されたかどうかを確認します。 **事前条件** provider.getState() == Provider.IN_SERVICE
boolean	hasRecordingCapabilityChanged()	このメソッドは、録音機能が変更されたかどうかを確認します。 **事前条件** provider.getState() == Provider.IN_SERVICE

関連資料

「定数フィールド値」（P.F-1）を参照してください。

CiscoProviderObserver

Provider.addObserver メソッドを使用してプロバイダーを監視する際に、このインターフェイスを実装して CiscoAddrCreatedEv や CiscoTermCreatedEv などの CiscoProvEv イベントを受信します。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

javax.telephony.ProviderObserver

宣言

public interface CiscoProviderObserver extends javax.telephony.ProviderObserver

メソッド

なし

継承したメソッド

インターフェイス javax.telephony.ProviderObserver から
providerChangedEvent

関連資料

CiscoAddrCreatedEv と CiscoTermCreatedEv を参照してください。

CiscoProvTerminalCapabilityChangedEv

このイベントは、端末機能が変更された場合にプロバイダーに送信されます。このイベントは、アプリケーション オブザーバで提供されます。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.0(1)	イベントが追加されました。
7.0(1)	CiscoMediaTerminals または CiscoRouteTerminals だけが返されるように CiscoTerminal[] が変更されました。

スーパーインターフェイス

CiscoEv, CiscoProvEv, javax.telephony.events.Ev, javax.telephony.events.ProvEv

宣言

public interface CiscoProvTerminalCapabilityChangedEv extends CiscoProvEv

フィールド

表 6-109　CiscoProvTerminalCapabilityChangedEv のフィールド

インターフェイス	フィールド	説明
static int	ID	なし

継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

表 6-110　CiscoProvTerminalCapabilityChangedEv のメソッド

インターフェイス	メソッド	説明
CiscoTerminal[]	getTerminals()	機能が変更された CiscoTerminals の配列を返します。Cisco Unified Communications Manager Release 7.0(1) では、CiscoMediaTerminals または CiscoRouteTerminals だけが返されるように、CiscoTerminal[] インターフェイスがモニタリングされます。

継承したメソッド

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.ProvEv から
getProvider

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

なし

CiscoRecorderInfo

このインターフェイスは、録音セッションにおける録音側に関する情報を提供します。録音セッションがアクティブな場合、このインターフェイスは、録音デバイスに関する情報を提供します。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

宣言

public interface CiscoRecorderInfo

フィールド

なし

メソッド

表 6-111　CiscoRecorderInfo のメソッド

インターフェイス	メソッド	説明
CiscoAddress	getAddress()	録音側のアドレスを返します。
java.lang.String	getTerminalName()	録音デバイスの端末名を返します。

関連資料

なし

CiscoRestrictedEv

CiscoRestrictedEv イベントは、CiscoAddrRestrictedEv および CiscoAddrRestrictedOnTerminalEv イベントの親クラスです。これは制限されたイベントの基底クラスであり、すべての制限されたイベントの原因コードを定義します。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

CiscoEv, CiscoProvEv, javax.telephony.events.Ev, javax.telephony.events.ProvEv

サブインターフェイス

CiscoAddrRestrictedEv, CiscoAddrRestrictedOnTerminalEv

宣言

public interface CiscoRestrictedEv extends CiscoProvEv

フィールド

表 6-112　CiscoRestrictedEv のフィールド

インターフェイス	フィールド	説明
static int	CAUSE_UNKNOWN	不明な理由のために端末が制限されています。
static int	CAUSE_UNSUPPORTED_DEVICE_CONFIGURATION	端末はサポートされない設定（たとえば、ロールオーバー オプション）のために制限されています。
static int	CAUSE_UNSUPPORTED_PROTOCOL	制御リスト内の端末は Cisco Unified JTAPI でサポートされていないプロトコルを使用しています。
static int	CAUSE_USER_RESTRICTED	端末またはアドレスが制限とマークされています。
static int	ID	なし

継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

なし

継承したメソッド

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.ProvEv から
getProvider

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

なし

CiscoRouteAddress

このインターフェイスは推奨されません。また、実装されていません。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

javax.telephony.Address, javax.telephony.callcenter.RouteAddress

宣言

public interface CiscoRouteAddress extends javax.telephony.callcenter.RouteAddress

フィールド

なし

継承したフィールド

インターフェイス javax.telephony.callcenter.RouteAddress から
ALL_ROUTE_ADDRESS

メソッド

表 6-113　CiscoRouteAddress のメソッド

インターフェイス	メソッド	説明
void	registerRouteCallback(javax.telephony.callcenter.RouteCallback routeCallback, boolean disableAutoRehoming)	推奨されません。 **例外** javax.telephony.ResourceUnavailableException javax,telephony.MethodNotSupportedException

継承したメソッド

インターフェイス javax.telephony.callcenter.RouteAddress から
cancelRouteCallback, getActiveRouteSessions, getRouteCallback, registerRouteCallback

インターフェイス javax.telephony.Address から
addCallObserver, addObserver, getAddressCapabilities, getCallObservers, getCapabilities, getConnections, getName, getObservers, getProvider, getTerminals, removeCallObserver, removeObserver

関連資料

なし

CiscoRouteEvent

CiscoRouteEvent インターフェイスは、Cisco 固有の機能を追加することによって RouteEvent インターフェイスを拡張します。getCallingPartyIpAddr メソッドを使用して、発信側デバイスの IP アドレスを取得できます。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	getCallingPartyIpAddr_v6() メソッドが追加されました。

スーパーインターフェイス

javax.telephony.callcenter.events.RouteEvent, javax.telephony.callcenter.events.RouteSessionEvent

宣言

public interface CiscoRouteEvent extends javax.telephony.callcenter.events.RouteEvent

フィールド

なし

継承したフィールド

インターフェイス javax.telephony.callcenter.events.RouteEvent から
SELECT_ACD, SELECT_EMERGENCY, SELECT_LEAST_COST, SELECT_NORMAL, SELECT_USER_DEFINED

メソッド

表 6-114　CiscoRouteEvent のメソッド

インターフェイス	メソッド	説明
java.net.InetAddress	getCallingPartyIpAddr_v6()	発信側の IPv6 アドレスを返します。IP アドレスが取得できない場合、このメソッドは IP アドレス 0::0 の InetAddress と、null のホスト名を返します。このオブジェクトを出力すると、「null/0::0」の文字列表現が出力されます。戻り値：InetAddress。
java.net.InetAddress	getCallingPartyIpAddr()	発信側の IP アドレスを返します。IP アドレスが取得できない場合、このメソッドは IP アドレス 0.0.0.0 の InetAddress と、null の ホスト名を返します。このオブジェクトを出力すると、「null/0.0.0.0」の文字列表現が出力されます。

継承したメソッド

インターフェイス javax.telephony.callcenter.events.RouteEvent から
getCallingAddress, getCallingTerminal, getCurrentRouteAddress, getRouteSelectAlgorithm, getSetupInformation

インターフェイス javax.telephony.callcenter.events.RouteSessionEvent から
getRouteSession

関連資料

なし

CiscoRouteSession

CiscoRouteSession インターフェイスは、RouteSession に関連付けられた、基になるコールにアプリケーションがアクセスすることをサポートします。また、このインターフェイスは RouteEndEvent のさまざまな内部エラーを表示します。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

javax.telephony.callcenter.RouteSession

宣言

public interface CiscoRouteSession extends javax.telephony.callcenter.RouteSession

フィールド

表 6-115　CiscoRouteSession のフィールド

インターフェイス	フィールド	説明
static final int	ERROR_ROUTESELECT_TIMEOUT	各 routeEvent() または reRouteEvent() が送られるたびに、routeSelect() または endRoute() で応答するアプリケーションのタイマーが始動します。このタイマーのデフォルト値は、5 秒です。この時間内にアプリケーションが応答しない場合、このエラーで endRoute が呼び出されます。
static final int	ERROR_NO_CALLBACK	デフォルト ルート メカニズムがないので、このアプリケーションにコールバックが登録されていない場合は、このエラーが設定された endRoute が呼び出されます。
static final int	ERROR_INVALID_STATE	ルーティング中に、内部的な InvalidStateException が発生した場合、または事前条件または事後条件の一部が満たされなかった場合には、このエラーで endRoute が呼び出されます。
static final int	DEFAULT_SEARCH_SPACE	この実装のデフォルトの検索スペースを使用してリダイレクトを実行します。デフォルトでは、発側の検索スペースを使用します。

表 6-115　CiscoRouteSession のフィールド （続き）

インターフェイス	フィールド	説明
static final int	CALLINGADDRESS_SEARCH_SPACE	発信元アドレスの検索スペースを使用してリダイレクトを実行する必要があることを示します。
static final int	ROUTEADDRESS_SEARCH_SPACE	ルート ポイント アドレスの検索スペースを使用して、リダイレクトを実行する必要があることを示します。
static final int	DONOT_RESET_ORIGINALCALLED	これは PreferedOriginalCalled Option のパラメータ値で、OriginalCalled を再設定しないことを指定します。
static final int	RESET_ORIGINALCALLED	これは PreferedOriginalCalled のパラメータ値で、preferredOriginalCalledOption の値がこの値に設定された場合、OriginalCalled は preferredOriginalCalledNumber にリセットされます。
static final int	CAUSE_CTIERR_FAC_CMC_REASON_FAC_NEEDED	RouteSession.getCause() で返されるこの定数は、selectRoute の routeSelectedElement に必要な FAC コードが含まれていないことを示します。
static final int	CAUSE_CTIERR_FAC_CMC_REASON_CMC_NEEDED	RouteSession.getCause() で返されるこの定数は、selectRoute の routeSelectedElement に必要な CMC コードが含まれていないことを示します。
static final int	CAUSE_CTIERR_FAC_CMC_REASON_FAC_CMC_NEEDED	RouteSession.getCause() で返されるこの定数は、selectRoute の routeSelectedElement に必要な FAC コードおよび CMC コードが含まれていないことを示します。
static final int	CAUSE_CTIERR_FAC_CMC_REASON_FAC_INVALID	RouteSession.getCause() で返されるこの定数は、selectRoute の routeSelectedElement に無効な FAC コードが含まれていることを示します。
static final int	CAUSE_CTIERR_FAC_CMC_REASON_CMC_INVALID	RouteSession.getCause() で返されるこの定数は、selectRoute の routeSelectedElement に有効な CMC コードが含まれていることを示します。

継承したフィールド

インターフェイス javax.telephony.callcenter.RouteSession から
CAUSE_INVALID_DESTINATION, CAUSE_NO_ERROR, CAUSE_PARAMETER_NOT_SUPPORTED, CAUSE_ROUTING_TIMER_EXPIRED, CAUSE_STATE_INCOMPATIBLE, CAUSE_UNSPECIFIED_ERROR, ERROR_RESOURCE_BUSY, ERROR_RESOURCE_OUT_OF_SERVICE, ERROR_UNKNOWN, RE_ROUTE, ROUTE, ROUTE_CALLBACK_ENDED, ROUTE_END, ROUTE_USED

メソッド

表 6-116　CiscoRouteSession のメソッド

インターフェイス	メソッド	説明
javax.telephony.Call	getCall()	この RouteSession. に関連付けられたコールを返します。
void	selectRoute (java.lang.String[] routeSelected, int callingSearchSpace)	RouteSession インターフェイスの selectRoute メソッドをオーバーロードして、コールがルートの宛先にリダイレクトされるときに使用する、発信元検索スペースをアプリケーションで指定できるようにします。 **例外** javax.telephony.MethodNotSupportedException **パラメータ** • callingSearchSpace：CiscoRouteSession.DEFAULT_SEARCH_SPACE、CiscoRouteSession.CALLINGADDRESS_SEARCH_SPACE、または CiscoRouteSession.ROUTEADDRESS_SEARCH_SPACE のいずれか。 • routeSelected：コールに対して可能な宛先のリスト。

表 6-116　CiscoRouteSession のメソッド（続き）

インターフェイス	メソッド	説明
void	selectRoute (java.lang.String[] routeSelected, int callingSearchSpace, java.lang.String[] modifyingCallingNumber)	発番号が変更されたコールをルーティング可能な 1 つ以上の宛先を選択します。このメソッドは、宛先の電話のアドレス名と、modifyingCallingNumber の文字列配列を優先順位に従って引数をとります。 最も優先順位の高い宛先が指定配列の最初の要素になり、ルーティングは、この宛先から順に、対応する発信者番号変更の要素を使用して試みられます。 modifiedCallingNumber 要素が null である場合、コールがその特定の routeSelected 要素にルーティングされた際に発信者番号が変更されません。宛先が正常に選択されるまで、指定された宛先アドレスを順に使用するように試行します。コールが宛先にルーティングされると、RouteUsedEvent がアプリケーションに配信されます。 **事前条件** • this.getRouteAddress().getProvider().getState() == • Provider.IN_SERVICE this.getState() == RouteSession.ROUTE or RouteSession.RE_ROUTE **事後条件** • this.getRouteAddress().getProvider().getState() == • Provider.IN_SERVICE this.getState() == • RouteSession.ROUTE_USED（コールが正常にルーティングされた場合）。正常な宛先が選択されると、この RouteSession に RouteUsedEvent が配信されます。 **パラメータ** • routeSelected：callcallingSearchSpace に対して可能な宛先は、CiscoRouteSession.DEFAULT_SEARCH_SPACE、CiscoRouteSession.CALLINGADDRESS_SEARCH_SPACE、または CiscoRouteSession.ROUTEADDRESS_SEARCH_SPACE です。 • modifyingCallingNumber：コールが routeSelected 要素に達したときにアプリケーションが発番号を変更する要素の配列です。 **例外** • com.cisco.jtapi.MethodNotSupportedExceptionImpl（この実装では、ルーティングをサポートしません） • javax.telephony.PrivilegeViolationException（ユーザに、このメソッドを使用するために必要な権限がありません） • javax.telephony.MethodNotSupportedException selectRoute

表 6-116　CiscoRouteSession のメソッド（続き）

インターフェイス	メソッド	説明
void	selectRoute (java.lang.String[] routeSelected, int callingSearchSpace, java.lang.String[] preferedOriginalCalledNumber, int[] preferedOriginalCalledOption	コールをルーティング可能な 1 つ以上の宛先を選択します。このメソッドは、宛先の電話のアドレス名の文字列配列（優先順位順）と、PreferredOriginalCalled 番号の文字列配列を引数にとります。 PreferedOriginalCalled 番号は、宛先電話名配列のインデックスに基づいて選択されます。宛先配列に対応するインデックスが、PreferedOriginalCalled 番号配列にない場合は、preferedOriginalCalled が宛先に設定されます。 最も優先順位の高い宛先が指定配列の最初の要素になり、ルーティングは、この宛先から連続して試みられます。コールが正常にルーティングされるまで、指定された各宛先アドレスが連続して試行されます。選択されたルーティングの宛先が成功し、コールがその宛先にルーティングされると、RouteUsedEvent イベントがアプリケーションに配信されます。 **事前条件** • this.getRouteAddress().getProvider().getState() == • Provider.IN_SERVICE this.getState() == RouteSession.ROUTE or • RouteSession.RE_ROUTE **事後条件** • this.getRouteAddress().getProvider().getState() == • Provider.IN_SERVICE this.getState() == • RouteSession.ROUTE_USED（コールが正常にルーティングされた場合）。正常な宛先が選択されると、この RouteSession に RouteUsedEvent が配信されます。 **パラメータ** • routeSelected：コールに対して可能な宛先。 • preferedOriginalCalledNumber：routeSelected リストの一致する配列インデックスのルートに対応する各項目のリスト。 • preferedOriginalCalledOption：routeSelected リストに対応する各オプションのリスト。このオプションは、OriginalCalled を preferedOriginalCalledNumber に設定するかどうかを指定します。オプションの値は CiscoRouteSession.DONOT_RESET_ORIGINALCALLED および CiscoRouteSession.RESET_ORIGINALCALLED です。値が指定されていないか null の場合、デフォルト値は CiscoRouteSession.DONOT_RESET_ORIGINALCALLED です。

表 6-116　CiscoRouteSession のメソッド （続き）

インターフェイス	メソッド	説明
		例外 com.cisco.jtapi.MethodNotSupportedExceptionImpl（この実装では、ルーティングをサポートしません） javax.telephony.PrivilegeViolationException（ユーザに、このメソッドを使用するために必要な権限がありません） javax.telephony.MethodNotSupportedException selectRoute
void	selectRoute (java.lang.String[] routeSelected, int callingSearchSpace, java.lang.String[] modifyingCallingNumber, java.lang.String[] preferedOriginalCalledNumber, int[] preferedOriginalCalledOption, java.lang.String[] facCode, java.lang.String[] cmcCode)	ルーティング可能な 1 つ以上の宛先を選択します。次の文字列配列を引数にとります。 • 宛先の電話のアドレス名（優先順位順） • PreferredOriginalCalled 番号 • FAC • CMC PreferedOriginalCalled 番号は、宛先電話名配列のインデックスに対応して選択されます。インデックスが referedOriginalCalled 番号配列にない場合は、preferedOriginalCalled が宛先に設定されます。 最も優先順位の高い宛先が指定配列の最初の要素になり、ルーティングは、この宛先から順に試みられます。コールが正常にルーティングされるまで、指定された宛先アドレスを順に使用するように試行します。選択されたルーティングの宛先が成功し、コールがその宛先にルーティングされると、RouteUsedEvent イベントがアプリケーションに配信されます。 **事前条件** • this.getRouteAddress().getProvider().getState() == • Provider.IN_SERVICE this.getState() == RouteSession.ROUTE • RouteSession.RE_ROUTE **事後条件** • this.getRouteAddress().getProvider().getState() == • Provider.IN_SERVICE this.getState() == • RouteSession.ROUTE_USED（コールが正常にルーティングされた場合）。正常な宛先が選択されると、この RouteSession に RouteUsedEvent が配信されます。 **パラメータ** • routeSelected：可能な宛先のリスト。 • preferedOriginalCalledNumber：routeSelected リスト内の同じ配列インデックスのルートに対応する各番号のリスト。

表 6-116　CiscoRouteSession のメソッド（続き）

インターフェイス	メソッド	説明
		• list.preferedOriginalCalledOption：RouteList に対応する各オプションのリスト。このオプションは、OriginalCalled を preferedOriginalCalledNumber に設定するかどうかを指定します。オプションの値は iscoRouteSession.DONOT_RESET_ORIGINALCALLED および CiscoRouteSession.RESET_ORIGINALCALLED です。値が指定されていないか null の場合、デフォルト値は CiscoRouteSession.DONOT_RESET_ORIGINALCALLED です。 • modifyingCallingNumber：コールがルート選択要素に達したときにアプリケーションが発番号を変更する要素の配列です。アプリケーションが発番号を変更しない場合は、アプリケーションによってこのパラメータに null 値が渡される必要があります。 • facCode (Forced Authorization Code [FAC])：routeSelected 要素に FAC が必要な場合、対応する facCode 要素にそのコードを含める必要があります。routeSelected 要素にコードが不要な場合、アプリケーションはこのパラメータに null 値を渡す必要があります。 • cmcCode：（Client Matter Code [CMC]）routeSelected 要素に CMC が必要な場合、対応する cmcCode 要素にそのコードを含める必要があります。routeSelected 要素にコードが不要な場合、アプリケーションはこのパラメータに null 値を渡す必要があります。 **例外** • com.cisco.jtapi.MethodNotSupportedExceptionImpl（この実装では、ルーティングをサポートしません） • javax.telephony.PrivilegeViolationException（ユーザに、このメソッドを使用するために必要な権限がありません） • javax.telephony.MethodNotSupportedException selectRoute

表 6-116　CiscoRouteSession のメソッド （続き）

インターフェイス	メソッド	説明
void	selectRoute (java.lang.String[] routeSelected, int callingSearchSpace, java.lang.String[] modifyingCallingNumber, java.lang.String[] prefered OriginalCalledNumber, int[] preferedOriginal CalledOption, java.lang.String[] facCode, java.lang.String[] cmcCode, int featurePriority)	ルーティング可能な 1 つ以上の宛先を選択します。次の文字列配列を引数にとります。 • 宛先の電話のアドレス名（優先順位順） • PreferredOriginalCalled 番号 • FAC • CMC • 整数の優先順位 PreferedOriginalCalled 番号は、宛先電話名配列のインデックスに基づいて選択されます。インデックスが referedOriginalCalled 番号配列にない場合は、preferedOriginalCalled が宛先に設定されます。 最も優先順位の高い宛先が指定配列の最初の要素になり、ルーティングは、この宛先から連続して試みられます。コールが正常にルーティングされるまで、指定された宛先アドレスを使用するように連続して試行します。選択されたルーティングの宛先が成功し、コールがその宛先にルーティングされると、RouteUsedEvent イベントがアプリケーションに配信されます。 **事前条件** • this.getRouteAddress().getProvider().getState() == • Provider.IN_SERVICE this.getState() == RouteSession.ROUTE • RouteSession.RE_ROUTE **事後条件** • this.getRouteAddress().getProvider().getState() == • Provider.IN_SERVICE this.getState() == • RouteSession.ROUTE_USED（コールが正常にルーティングされた場合）。 **パラメータ** routeSelected：コールに対して可能な宛先。 preferedOriginalCalledNumber：routeSelected リスト内の同じ配列インデックスのルートに対応する各要素のリスト。 preferedOriginalCalledOption：RouteList に対応する各オプションのリスト。このオプションは、OriginalCalled を preferedOriginalCalledNumber に設定するかどうかを指定します。オプションの値は CiscoRouteSession.DONOT_RESET_ORIGINALCALLED および CiscoRouteSession.RESET_ORIGINALCALLED です。値が指定されていないか null の場合、デフォルト値は CiscoRouteSession.DONOT_RESET_ORIGINALCALLED です。

表 6-116　CiscoRouteSession のメソッド（続き）

インターフェイス	メソッド	説明
		• modifyingCallingNumber：コールが routeselected 要素に達したときにアプリケーションが発番号を変更する要素の配列です。アプリケーションが発番号を変更しない場合は、アプリケーションによってこのパラメータに null 値が渡される必要があります。 • facCode (Forced Authorization Code [FAC])：routeSelected 要素に FAC が必要な場合、対応する facCode 要素にそのコードを含める必要があります。routeSelected 要素にコードが不要な場合、アプリケーションはこのパラメータに null 値を渡す必要があります。 • cmcCode (Client Matter Code [CMC])：routeSelected 要素に CMC が必要な場合、対応する cmcCode 要素にそのコードを含める必要があります。routeSelected 要素にコードが不要な場合、アプリケーションはこのパラメータに null 値を渡す必要があります。 • featurePriority：コールの機能の優先順位を設定します。特定の優先順位を設定しない場合、CiscoCall.FEATUREPRIORITY_NORMAL が設定されます。featurePriority パラメータは、次のいずれかになります。 – CiscoCall.FEATUREPRIORITY_NORMAL – CiscoCall.FEATUREPRIORITY_URGENT – CiscoCall.FEATUREPRIORITY_EMERGENCY **例外** • javax.telephony.PrivilegeViolationException（ユーザに、このメソッドを使用するために必要な権限がありません） • com.cisco.jtapi.MethodNotSupportedExceptionImpl（この実装では、ルーティングをサポートしません） • javax.telephony.MethodNotSupportedException

表 6-116　CiscoRouteSession のメソッド （続き）

インターフェイス	メソッド	説明
void	selectRoute(java.lang.String[] routeSelected, int[] callingSearchSpace, java.lang.String[] modifyingCallingNumber, java.lang.String[] preferedOriginalCalledNumber, int[] preferedOriginalCalledOption, java.lang.String[] facCode, java.lang.String[] cmcCode, int[] featurePriority)	ルーティング可能な 1 つ以上の宛先を選択します。次の文字列配列を引数にとります。 • 宛先の電話のアドレス名（優先順位順） • コーリング サーチ スペース • modifyingCallingNumbers • PreferredOriginalCalled 番号 • FAC • CMC • 機能プライオリティ PreferedOriginalCalled 番号は、宛先電話名配列のインデックスに基づいて選択されます。インデックスが referedOriginalCalled 番号配列にない場合は、preferedOriginalCalled が宛先に設定されます。 最も優先順位の高い宛先が指定配列の最初の要素になり、ルーティングは、この宛先から連続して試みられます。コールが正常にルーティングされるまで、指定された宛先アドレスを使用するように連続して試行します。 選択されたルーティングの宛先が成功し、コールがその宛先にルーティングされると、RouteUsedEvent イベントがアプリケーションに配信されます。 **事前条件** • this.getRouteAddress().getProvider().getState() == • Provider.IN_SERVICE this.getState() == RouteSession.ROUTE • RouteSession.RE_ROUTE **事後条件** • this.getRouteAddress().getProvider().getState() == • Provider.IN_SERVICE this.getState() == • RouteSession.ROUTE_USED（コールが正常にルーティングされた場合）。 **パラメータ** • routeSelected：コールに対して可能な宛先のリスト。 • callingSearchSpace：選択したルートごとに、CiscoRouteSession.DEFAULT_SEARCH_SPACE、CiscoRouteSession.CALLINGADDRESS_SEARCH_SPACE、または CiscoRouteSession.ROUTEADDRESS_SEARCH_SPACE です。

表 6-116　CiscoRouteSession のメソッド（続き）

インターフェイス	メソッド	説明
		• preferedOriginalCalledNumber：routeSelected リスト内の同じ配列インデックスのルートに対応する各要素のリスト。 • preferedOriginalCalledOption：RouteList に対応する各オプションのリスト。このオプションは、OriginalCalled を preferedOriginalCalledNumber に設定するかどうかを指定します。オプションの値は CiscoRouteSession.DONOT_RESET_ORIGINALCALLED および CiscoRouteSession.RESET_ORIGINALCALLED です。値が指定されていないか null の場合、デフォルト値は CiscoRouteSession.DONOT_RESET_ORIGINALCALLED です。 • modifyingCallingNumber：コールが routeselected 要素に達したときにアプリケーションが発番号を変更する要素の配列です。アプリケーションが発番号を変更しない場合は、アプリケーションによってこのパラメータに null 値が渡される必要があります。 • facCode (Forced Authorization Code [FAC])：routeSelected 要素に FAC が必要な場合、対応する facCode 要素にそのコードを含める必要があります。routeSelected 要素にコードが不要な場合、アプリケーションはこのパラメータに null 値を渡す必要があります。 • cmcCode (Client Matter Code [CMC])：routeSelected 要素に CMC が必要な場合、対応する cmcCode 要素にそのコードを含める必要があります。routeSelected 要素にコードが不要な場合、アプリケーションはこのパラメータに null 値を渡す必要があります。 • featurePriority：選択したルートごとに、機能プライオリティを設定できます。特定の優先順位を設定しない場合、CiscoCall.FEATUREPRIORITY_NORMAL が設定されます。featurePriority パラメータは、CiscoCall.FEATUREPRIORITY_NORMAL、CiscoCall.FEATUREPRIORITY_URGENT、または CiscoCall.FEATUREPRIORITY_EMERGENCY のいずれかになります。 **例外** • javax.telephony.PrivilegeViolationException（ユーザに、このメソッドを使用するために必要な権限がありません） • com.cisco.jtapi.MethodNotSupportedExceptionImpl（この実装では、ルーティングをサポートしません） • javax.telephony.MethodNotSupportedException

継承したメソッド

インターフェイス javax.telephony.callcenter.RouteSession から
endRoute, getCause, getRouteAddress, getState, selectRoute

関連資料

「定数フィールド値」（P.F-1）を参照してください。

CiscoRouteTerminal

CiscoRouteTerminal は、アプリケーションによる RTP メディア ストリームの終端を可能にする、特殊な CiscoTerminal です。CiscoRouteTerminal は、通常の CiscoTerminal と異なり、物理的なテレフォニー エンドポイントではないため、サードパーティ方式で監視、制御することができます。CiscoRouteTerminal は論理的なテレフォニー エンドポイントなので、コールをルーティングしメディアを終端する必要があるアプリケーションに関連付けることができます。CiscoMediaTerminal とは異なり、CiscoRouteTerminal では複数のアクティブ コールを同時に保持できます。通常、アプリケーションは、エージェントが次の発信者に応対できるようになるまでの間、コールをキューに格納するために使用されます。

（注）　CiscoRouteTerminal は Cisco Unified Communications Manager の CTI Route Point です。

メディアの終端プロセスには次の 3 つの段階があります。

ステップ 1　アプリケーションが、CiscoRouteTerminal.register メソッドを使用して、メディア機能をこの端末に登録します。

ステップ 2　CiscoTerminalObserver インターフェイスを実装するオブザーバを、アプリケーションが Terminal.addObserver メソッドを使用して追加します。

ステップ 3　アプリケーションは CiscoRouteTerminal または CiscoRouteAddress で addCallObserver を呼び出して、コールを着信して応答する必要があります。

アプリケーションはコールごとに CiscoMediaOpenLogicalChannelEv を受信します。メディアが停止した場合は、再確立する必要があります。アプリケーションは、CiscoRouteTerminal の setRTPParams メソッドを使用して、IP アドレスおよびポート番号を提示する必要があります。

（注）　**重要**：CiscoJtapiClient 1.4 リリース以前で作成されたすべてのアプリケーションは、メディア終端を必要としない場合、修正して CiscoRouteTerminal.NO_MEDIA_TERMINATION タイプに登録する必要があります。

登録されるメディア機能および登録タイプが同じであれば、複数のアプリケーションを同じ RoutePoint に登録できます。CiscoRouteTerminal.DYNAMIC_MEDIA_REGISTRATION に登録された場合は、callObserver が追加されたときに CiscoMediaOpenLogicalChannelEv を受信します。ただし、1 つのアプリケーションだけが setRTPParams を呼び出すことができます。

メディア終端を行うアプリケーションは、RouteAddress または CiscoRouteTerminals に CallObserver を追加する必要があります。アプリケーションは動的タイプで登録せず、registerRouteCallBack を追加する必要があります。メディア終端を行わない場合、アプリケーションは、registerRouteCallBack だけを使用する必要があります。アプリケーションは registerRouteCallBack と callObserver を同時に追加できません。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	activeAddressingMode に次のモードが追加されました。 • CiscoTerminal.IP_ADDRESSING_MODE_IPv6 • CiscoTerminal.IP_ADDRESSING_MODE_IPv4_v6

スーパーインターフェイス

CiscoObjectContainer, CiscoTerminal, javax.telephony.Terminal

宣言

public interface CiscoRouteTerminal extends CiscoTerminal

フィールド

表 6-117　CiscoRouteTerminal のフィールド

インターフェイス	フィールド	説明
static int	DYNAMIC_MEDIA_REGISTRATION	メディア終端を行うアプリケーションは、登録要求でこのタイプを登録し、サポートしている機能を渡す必要があります。
static int	NO_MEDIA_REGISTRATION	メディア終端を行わないアプリケーションは、登録要求でこのタイプを登録し、CiscoMediaCapability に null 値を渡す必要があります。 registrationType が CiscoRouteTerminal.NO_MEDIA_REGISTRATION である場合、アプリケーションはメディアを終端できず、CiscoRouteTerminal をコールのルーティング用に使用できます。

継承したフィールド

インターフェイス com.cisco.jtapi.extensions.CiscoTerminal から
ASCII_ENCODING, DEVICESTATE_ACTIVE, DEVICESTATE_ALERTING, DEVICESTATE_HELD, DEVICESTATE_IDLE, DEVICESTATE_UNKNOWN, DEVICESTATE_WHISPER, DND_OPTION_CALL_REJECT, DND_OPTION_NONE,

DND_OPTION_RINGER_OFF, IN_SERVICE, IP_ADDRESSING_MODE_IPV4, IP_ADDRESSING_MODE_IPV4_V6, IP_ADDRESSING_MODE_IPV6, IP_ADDRESSING_MODE_UNKNOWN, IP_ADDRESSING_MODE_UNKNOWN_ANATRED, NOT_APPLICABLE, OUT_OF_SERVICE, UCS2UNICODE_ENCODING, UNKNOWN_ENCODING

メソッド

表 6-118　CiscoRouteTerminal のメソッド

インターフェイス	メソッド	説明
void	register (CiscoMediaCapability[] capabilities, int registrationType)	指定した CiscoMediaCapabilities と登録タイプで端末を登録します。プロバイダーは Provider.IN_SERVICE 状態になっている必要があります。CiscoRouteTerminal が登録されると、正常に終了します。 **パラメータ** • capabilities：アプリケーションがこの端末に対してサポートする RTP 符号化方式のリスト。アプリケーションでメディア終端に関する処理をしない場合は null 値を渡すことができます。 • registrationType：CiscoRouteTerminal.DYNAMIC_MEDIA_REGISTRATION または CiscoRouteTerminal.NO_MEDIA_REGISTRATION registrationType が CiscoRouteTerminal.NO_MEDIA_REGISTRATION である場合、アプリケーションはメディアを終端できず、CiscoRouteTerminal をコールのルーティング用に使用できます。 registrationType が CiscoRouteTerminal.DYNAMIC_MEDIA_REGISTRATION である場合、アプリケーションはメディアを終端でき、アクティブ コールを複数持つことができます。この registrationType は、アプリケーションが各コールに対して動的に IP アドレスおよびポートを指定することを示します。この種類で登録するアプリケーションは、コールごとに CiscoMediaOpenLogicalChannelEv を受信するので、CiscoRouteTerminal の setRTPParams メソッドを使用して、IP アドレスおよびポート番号を提示する必要があります。 **例外** • CiscoRegistrationException

表 6-118　CiscoRouteTerminal のメソッド （続き）

インターフェイス	メソッド	説明
void	register (CiscoMediaCapability[] capabilities, int registrationType, int[] algorithmIDs)	指定された CiscoMediaCapabilities、registrationType およびサポートされる SRTP アルゴリズムで端末を登録します。プロバイダーは Provider.IN_SERVICE 状態になっている必要があります。CiscoRouteTerminal が登録されると、このメソッドは正常に終了します。 パラメータ • capabilities：アプリケーションがこの端末に対してサポートする RTP 符号化方式のリスト。アプリケーションでメディア終端に関する処理をしない場合は null 値を渡すことができます。 • registrationType：CiscoRouteTerminal.DYNAMIC_MEDIA_REGISTRATION または CiscoRouteTerminal.NO_MEDIA_REGISTRATION registrationType が CiscoRouteTerminal.NO_MEDIA_REGISTRATION である場合、アプリケーションはメディアを終端できず、CiscoRouteTerminal をコールのルーティング用に使用できます。このメソッドのその他のパラメータは無視されます。 registrationType が CiscoRouteTerminal.DYNAMIC_MEDIA_REGISTRATION である場合、アプリケーションはメディアを終端でき、アクティブ コールを複数持つことができます。この registrationType は、アプリケーションが各コールに対して動的に IP アドレスおよびポートを指定することを示します。この種類で登録するアプリケーションは、コールごとに CiscoMediaOpenLogicalChannelEv を受信するので、setRTPParams メソッドを使用して、IP アドレスおよびポート番号を提示する必要があります。 • algorithmIDs：アプリケーションがこの端末に対してサポートする SRTP アルゴリズムのリスト。これを使用するには、アプリケーションで TLS Link および SRTP Enabled フラグを有効にする必要があります。AlgorithmID は、CiscoMediaEncryptionSupportedAlgorithms のいずれかにする必要があります。 **例外** • javax.telephony.PrivilegeViolationException （アプリケーションはこのメソッドを使用しようとしますが、使用するための権限がありません）。 • CiscoRegistrationException

表 6-118　CiscoRouteTerminal のメソッド （続き）

インターフェイス	メソッド	説明
void	register (CiscoMediaCapability[] capabilities, int registrationType, int[] algorithmIDs, int activeAddressingMode)	CiscoRouteTerminal は CiscoTerminal.UNREGISTERED 状態になっている必要があり、プロバイダーは Provider.IN_SERVICE 状態になっている必要があります。このメソッドが成功した場合は、RouteTerminal が登録されます。指定された CiscoMediaCapabilities 、登録タイプおよびサポートされる SRTP アルゴリズムで端末を登録します。 registrationType が CiscoRouteTerminal.NO_MEDIA_ REGISTRATION である場合、アプリケーションはメディアを終端できず、ルート ポイントをコールのルーティング用に使用できます。このメソッドのその他のパラメータは無視されます。 registrationType が CiscoRouteTerminal. DYNAMIC_MEDIA_REGISTRATION である場合、アプリケーションはメディアを終端でき、アクティブ コールを複数持つことができます。これは、アプリケーションが各コールに対して動的に ipAddress およびポートを指定することを示します。 このタイプに登録するアプリケーションは、各コールに対して CiscoMediaOpenLogicalChannelEv を受信し、CiscoRouteTerminal に対する setRTPParams メソッドを使用して ipAddress とポート番号を指定する必要があります。 **メソッドの引数** アプリケーションがこの端末に対してサポートする RTP 符号化方式の種類を示します。アプリケーションでメディア終端に関する処理をしない場合は null 値を渡すことができます。registrationType は CiscoRouteTerminal.NO_MEDIA_ REGISTRATION または CiscoRouteTerminal. DYNAMIC_MEDIA_REGISTRATION です。 サポートされるアルゴリズムはアプリケーションがこの端末に対してサポートする SRTP アルゴリズムです。これを使用するには、アプリケーションで TLS Link および SRTP Enabled フラグを有効にする必要があります。アプリケーションにこのメソッドを使用する権限がない場合、PrivilegeViolationException がスローされます。 **事後条件** CiscoRouteTerminal が登録されると、このメソッドは正常に終了します。 **パラメータ** • capabilities：この端末でサポートされている RTP 符号化方式のリスト。 • registrationType : CiscoRouteTerminal.DYNAMIC_MEDIA_REGISTRATION または CiscoRouteTerminal. NO_MEDIA_REGISTRATION • algorithmIDs：サポートされる SRTP アルゴリズムのリスト。AlgorithmID は、CiscoMediaEncryption SupportedAlgorithms のいずれか 1 つだけになります。

表 6-118　CiscoRouteTerminal のメソッド （続き）

インターフェイス	メソッド	説明
		• activeAddressingMode：アプリケーションがこの CiscoRouteTerminal を登録する IP アドレッシング モード モードは次のいずれかです。 – CiscoTerminal.IP_ADDRESSING_MODE_IPv4 – CiscoTerminal.IP_ADDRESSING_MODE_IPv6 – CiscoTerminal.IP_ADDRESSING_MODE_IPv4_v6 **例外** • CiscoRegistrationException • javax.telephony.PrivilegeViolationException
void	unregister()	CiscoRouteTerminal は登録されている必要があり、プロバイダーは Provider.IN_SERVICE 状態になっている必要があります。このメソッドが成功した場合は、CiscoRouteTerminal の登録が解除されます。 **事後条件** • MediaTerminal の登録が解除されると、このメソッドは正常に終了します。 **例外** • CiscoUnregistrationException
void	setRTPParams(CiscoRTPHandle rtpHandle, CiscoRTPParams rtpParams)	アプリケーションは、IP アドレスおよび RTP ポート番号を設定して、コールのメディア ストリームを動的に行います。これを行うには、機能だけを指定して MediaTerminal または CiscoRouteTeminal を登録する必要があります。 アプリケーションでは、TerminalObserver で CiscoCallOpenLogicalChannelEv を受信したときに、このメソッドを起動する必要があります。 **パラメータ** rtpHandle：アプリケーションが CiscoCallOpenLogicalChannelEvrtpParams で受け取るハンドル。CiscoRTPParams を参照してください。 **例外** • javax.telephony.InvalidStateException • javax.telephony.InvalidArgumentException • javax.telephony.PrivilegeViolationException
boolean	isRegistered()	CiscoMediaTerminal が登録されている場合は true、そうでない場合は false を返します。CiscoRouteTerminal が OutOfService である場合は false、InService である場合は true を返します。CTIManager の障害の場合は、false を返します。

表 6-118　CiscoRouteTerminal のメソッド （続き）

インターフェイス	メソッド	説明
boolean	isRegisteredByThisApp()	このメソッドは、このアプリケーションが登録要求を発行して成功した場合に true を返します。この登録は、CTIManager の障害によりデバイスがアウトオブサービス状態にある場合でも有効です。これは、このアプリケーションがデバイスの登録を解除するまで true を返します。
int	getIPAddressingMode()	アプリケーションはこの API を起動して、CiscoRouteTerminal の IP アドレッシング モードを照会できます。アドレッシング モードは次のいずれかの定数になります。 • CiscoTerminal.IP_ADDRESSING_IPv4 • CiscoTerminal.IP_ADDRESSING_IPv6 • CiscoTerminal.IP_ADDRESSING_IPv4_v6

継承したメソッド

インターフェイス com.cisco.jtapi.extensions.CiscoTerminal から

createSnapshot, getAltScript, getDeviceState, getDNDOption, getDNDStatus, getEMLoginUsername, getFilter, getLocale, getProtocol, getRegistrationState, getRTPInputProperties, getRTPOutputProperties, getState, getSupportedEncoding, isRestricted, sendData, sendData, setDNDStatus, setFilter, unPark

インターフェイス javax.telephony.Terminal から

addCallObserver, addObserver, getAddresses, getCallObservers, getCapabilities, getName, getObservers, getProvider, getTerminalCapabilities, getTerminalConnections, removeCallObserver, removeObserver

インターフェイス com.cisco.jtapi.extensions.CiscoObjectContainer から

getObject, setObject

関連資料

CiscoTerminal と 「定数フィールド値」（P.F-1）を参照してください。
CiscoMediaOpenLogicalChannelEv

CiscoRouteUsedEvent

CiscoRouteUsedEvent イベントは、RouteSession が RouteSession.ROUTE_USED 状態に移行し、アプリケーションによるルーティングの結果、コールが宛先で終端したことを示します。このインターフェイスは RouteUsedEvent インターフェイスを拡張し、RouteCallback インターフェイスによって報告されます。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

javax.telephony.callcenter.events.RouteSessionEvent,
javax.telephony.callcenter.events.RouteUsedEvent

宣言

public interface CiscoRouteUsedEvent extends javax.telephony.callcenter.events.RouteUsedEvent

フィールド

なし

メソッド

表 6-119　CiscoRouteUsedEvent のメソッド

インターフェイス	メソッド	説明
Int	getRouteSelectedIndex()	コールのルーティング先となるルートの配列インデックスを返します。

継承したメソッド

インターフェイス javax.telephony.callcenter.events.RouteUsedEvent から
getCallingAddress, getCallingTerminal, getDomain, getRouteUsed

インターフェイス javax.telephony.callcenter.events.RouteSessionEvent から
getRouteSession

関連資料

RouteSession、RouteCallback および RouteSessionEvent を参照してください。

CiscoRTPBitRate

RTPBitRate インターフェイスには、G.723 RTP ビット レートを記述する定数が入ります。CiscoRTPInputProperties.getBitRate と CiscoRTPOutputProperties.getBitRate はこれらの定数を返します。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

宣言

public interface CiscoRTPBitRate

フィールド

表 6-120　CiscoRTPBitRate のフィールド

インターフェイス	フィールド	説明
static int	R5_3	この定数は 5.3k G.723 ビット レートです。
static int	R6_4	この定数は 6.4k G.723 ビット レートです。

メソッド

なし

関連資料

CiscoRTPInputProperties.getBitRate()、CiscoRTPOutputProperties.getBitRate() を参照してください。

CiscoRTPHandle

CiscoProvider.getCall(CiscoRTPHandle) を使用してコール参照を取得するために、CiscoRTPHandle オブジェクトを使用します。このオブジェクトは CiscoMediaCallOpenLogicalChannelEv で返されます。CiscoMediaCallOpenLogicalChannelEv event をどこで受信したかに応じて、CiscoMediaTerminal または CiscoRouteTerminal の setRTPParams パラメータでこのハンドルを渡します。

コール オブザーバが追加されていない場合や、CiscoMediaCallOpen LogicalChannelEv が送信されたときにコール オブザーバが追加されていなかった場合、CiscoProvider.getCall(CiscoRTPHandle) はメソッドとして null を返します。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

宣言

public interface CiscoRTPHandle

フィールド

なし

メソッド

表 6-121　CiscoRTPHandle のメソッド

インターフェイス	メソッド	説明
Int	getHandle()	コールの Cisco Unified Communications Manager CallLeg ID を整数の形式で返します。

関連資料

なし

CiscoRTPInputKeyEv

CiscoRTPInputKeyEv イベント インターフェイスは、暗号化された着信メディア ストリームの鍵情報を提供します。アプリケーションは CiscoTermEvFilter.setRTPKeyEventsEnabled(true) を使用してフィルタを設定し、TerminalObserver.terminalChangedEvent() を介してこのイベントを取得する必要があります。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

CiscoEv, CiscoTermEv, javax.telephony.events.Ev, javax.telephony.events.TermEv

宣言

public interface CiscoRTPInputKeyEv extends CiscoTermEv

フィールド

表 6-122　CiscoRTPInputKeyEv のフィールド

インターフェイス	フィールド	説明
static int	ID	なし

継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

表 6-123　CiscoRTPInputKeyEv のメソッド

インターフェイス	メソッド	説明
int	getCiscoMediaEncryptionKeyInfo()	プロバイダーが TLS リンクで開かれており、Cisco Unified Communications Manager administration でアプリケーションの SRTP を有効にするオプションが設定されている場合にだけ、CiscoMediaEncryptionKeyInfo を返します。そうでない場合は null を返します。

表 6-123　CiscoRTPInputKeyEv のメソッド

インターフェイス	メソッド	説明
int	getCiscoMediaSecurityIndicator()	次の定数の 1 つのメディアのセキュリティ インジケータを返します。 • CiscoMediaSecurityIndicator.MEDIA_ENCRYPTED_KEYS_AVAILABLE • CiscoMediaSecurityIndicator.MEDIA_ENCRYPT_USER_NOT_AUTHORIZED • CiscoMediaSecurityIndicator.MEDIA_ENCRYPTED_KEYS_UNAVAILABLE.
CiscoCallID	getCallID()	このイベントの送信時に CiscoCall がすでに存在している場合、CiscoCallID オブジェクトを返します。CiscoCall がない場合、このメソッドは null を返します。getCallID().getCall() は この鍵が適用されるコールを指定します。
int	getCiscoRTPHandle()	CiscoRTPHandle オブジェクトを返します。アプリケーションは、CiscoProvider.getCall を使用してコール参照を取得できます。コール オブザーバがない場合や、このイベントの配信時にコール オブザーバがなかった場合、CiscoProvider.getCall は null を返す可能性があります。戻り値：CiscoRTPHandle。

継承したメソッド

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.TermEv から
getTerminal

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

CiscoRTPParams と CiscoMediaSecurityIndicator を参照してください。

CiscoRTPInputProperties

CiscoRTPInputProperties インターフェイスは、端末が受信したメディアのプロパティを返します（受信メディア ストリーム）。CiscoRTPInputStartedEv はメディアが起動していることを示します。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

宣言

public interface CiscoRTPInputProperties

フィールド

なし

メソッド

表 6-124　CiscoRTPInputProperties のメソッド

インターフェイス	メソッド	説明
int	getBitRate()	CiscoRTPBitRate.R5_3 または CiscoRTPBitRate.R6_4 のメディア ビット レートを返します。
boolean	getEchoCancellation()	アプリケーションでエコー キャンセルを使用する必要がある場合は、true を返します。
java.net.InetAddress	getLocalAddress()	メディアが向けられるアドレスを返します。
int	getLocalPort()	メディアが向けられるポートを返します。
int	getPacketSize()	パケット サイズ（ミリ秒単位）を返します。

表 6-124　CiscoRTPInputProperties のメソッド （続き）

インターフェイス	メソッド	説明
int	getPayloadType()	次の定数の 1 つのペイロード形式を返します。 • CiscoRTPPayload.G711ALAW64K • CiscoRTPPayload.G711ALAW56K • CiscoRTPPayload.G711ULAW64K • CiscoRTPPayload.G711ULAW56K • CiscoRTPPayload.G722_64K • CiscoRTPPayload.G722_56K • CiscoRTPPayload.G722_48K • CiscoRTPPayload.G7231 • CiscoRTPPayload.G728 • CiscoRTPPayload.G729 • CiscoRTPPayload.G729ANNEXA • CiscoRTPPayload.ACY_G729AASSN • CiscoRTPPayload.DATA64 • CiscoRTPPayload.DATA56 • CiscoRTPPayload.GSM • CiscoRTPPayload.WIDEBAND_256K

関連資料

CiscoRTPPayload と CiscoRTPBitRate を参照してください。

CiscoRTPInputStartedEv

CiscoRTPInputStartedEv イベントは着信メディアの起動を示します。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

CiscoEv, CiscoTermEv, javax.telephony.events.Ev, javax.telephony.events.TermEv

宣言

public interface CiscoRTPInputStartedEv extends CiscoTermEv

フィールド

表 6-125　CiscoRTPInputStartedEv のフィールド

インターフェイス	フィールド	説明
static int	ID	なし

継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

表 6-126　CiscoRTPInputStartedEv のメソッド

インターフェイス	メソッド	説明
CiscoCallID	getCallID()	CiscoCallID を返します。
CiscoRTPHandle	getCiscoRTPHandle()	CiscoRTPHandle オブジェクトを返します。
int	getMediaConnectionMode()	CiscoMediaConnectionMode を返します。

表 6-126　CiscoRTPInputStartedEv のメソッド

インターフェイス	メソッド	説明
int	getRTPInputProperties()	着信メディアの特性を示す CiscoRTPInputProperties を返します。

継承したメソッド

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.TermEv から
getTerminal

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

「定数フィールド値」（P.F-1）、CiscoRTPInputProperties、CiscoCallID、CiscoRTPParams、および CiscoMediaConnectionMode を参照してください。

CiscoRTPInputStoppedEv

CiscoRTPInputStoppedEv イベントは着信メディア ストリームが停止したことを示します。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

CiscoEv, CiscoTermEv, javax.telephony.events.Ev, javax.telephony.events.TermEv

宣言

public interface CiscoRTPInputStoppedEv extends CiscoTermEv

フィールド

表 6-127　CiscoRTPInputStoppedEv のフィールド

インターフェイス	フィールド	説明
static int	ID	なし

継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

表 6-128　CiscoRTPInputStoppedEv のメソッド

インターフェイス	メソッド	説明
CiscoCallID	getCallID()	CiscoCallID を返します。CiscoRTPInputStartedEv は CiscoCallID.getCall() に適用されます。
CiscoRTPHandle	getCiscoRTPHandle()	CiscoRTPHandle オブジェクトを返します。アプリケーションは、CiscoProvider.getCall を使用してコール参照を取得できます。コール オブザーバがない場合や、このイベントの配信時にコール オブザーバがなかった場合、CiscoProvider.getCall は null を返す可能性があります。

表 6-128　CiscoRTPInputStoppedEv のメソッド

インターフェイス	メソッド	説明
int	getMediaConnectionMode()	次のいずれかの値の mediaMode とともに CiscoMediaConnectionMode を返します。 • CiscoMediaConnectionMode.RECEIVE_ONLY（一方向メディア、受信だけ） • CiscoMediaConnectionMode.TRANSMIT_AND_RECEIVE：（双方向メディア） 通常、モード NONE のイベントを受け取ることはありません。受け取った場合は、そのイベントを無視してエラーを記録してください。

継承したメソッド

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.TermEv から
getTerminal

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

「定数フィールド値」（P.F-1）、CiscoMediaConnectionMode、CiscoCallID および CiscoRTPParams を参照してください。

CiscoRTPOutputKeyEv

CiscoRTPOutputKeyEv イベントは、暗号化された発信メディア（転送）ストリームの鍵情報を提供します。アプリケーションは CiscoTermEvFilter.setRTPKeyEventsEnabled(true) を使用してフィルタを設定し、TerminalObserver.terminalChangedEvent() を介してこのイベントを取得します。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

CiscoEv, CiscoTermEv, javax.telephony.events.Ev, javax.telephony.events.TermEv

宣言

public interface CiscoRTPOutputKeyEv extends CiscoTermEv

フィールド

表 6-129　CiscoRTPOutputKeyEv のフィールド

インターフェイス	フィールド	説明
static int	ID	なし

継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

表 6-130　CisctoRTPOutputKeyEv のメソッド

インターフェイス	メソッド	説明
CiscoCallID	getCallID()	このイベントの送信時に CiscoCall がすでに存在している場合、CiscoCallID オブジェクトを返します。CiscoCall がない場合、このメソッドは null を返します。

表 6-130　CisctoRTPOutputKeyEv のメソッド （続き）

インターフェイス	メソッド	説明
CiscoMedia EncryptionKeyInfo	getCiscoMediaEncryptionKeyInfo()	プロバイダーが TLS リンクで開かれており、Cisco Unified Communications Manager administration でアプリケーションの SRTP を有効にするオプションが設定されている場合にだけ、CiscoMediaEncryptionKeyInfo を返します。そうでない場合は null を返します。
int	getCiscoMediaSecurityIndicator()	次の定数の 1 つのメディアのセキュリティ インジケータを返します。 • CiscoMediaSecurityIndicator.MEDIA_ENCRYPTED_KEYS_AVAILABLE • CiscoMediaSecurityIndicator.MEDIA_ENCRYPT_USER_NOT_AUTHORIZED • CiscoMediaSecurityIndicator.MEDIA_ENCRYPTED_KEYS_UNAVAILABLE
CiscoRTPHandle	getCiscoRTPHandle()	CiscoRTPHandle オブジェクトを返します。アプリケーションは、CiscoProvider.getCall を使用してコール参照を取得できます。コール オブザーバがない場合や、このイベントの配信時にコール オブザーバがなかった場合、CiscoProvider.getCall は null を返す可能性があります。

継承したメソッド

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.TermEv から
getTerminal

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

「定数フィールド値」（P.F-1）、CiscoRTPParams、および CiscoMediaSecurityIndicator を参照してください。

CiscoRTPOutputProperties

CiscoRTPPutputProperties インターフェイスは、端末によって転送されたメディアのプロパティを提供します。CiscoRTPOutPutStartedEv はメディアが起動していることを示します。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

宣言

public interface CiscoRTPOutputProperties

フィールド

なし

メソッド

表 6-131　CiscoRTPOutputProperties のメソッド

インターフェイス	メソッド	説明
int	getBitRate()	次の定数の 1 つのメディア ビット レートを返します。 • CiscoRTPBitRate.R5_3 • CiscoRTPBitRate.R6_4
int	getMaxFramesPerPacket()	パケットあたりの最大送信フレーム数を返します。
int	getPacketSize()	パケット サイズ（ミリ秒単位）を返します。
int	getPayloadType()	次の定数の 1 つのペイロード形式を返します。 • CiscoRTPPayload.NONSTANDARD • CiscoRTPPayload.G711ALAW64K • CiscoRTPPayload.G711ALAW56K • CiscoRTPPayload.G711ULAW64K • CiscoRTPPayload.G711ULAW56K • CiscoRTPPayload.G722_64K • CiscoRTPPayload.G722_56K • CiscoRTPPayload.G722_48K • CiscoRTPPayload.G7231 • CiscoRTPPayload.G728 • CiscoRTPPayload.G729 • CiscoRTPPayload.G729ANNEXA • CiscoRTPPayload.IS11172AUDIOCAP • CiscoRTPPayload.IS13818AUDIOCAP • CiscoRTPPayload.ACY_G729AASSN • CiscoRTPPayload.DATA64 • CiscoRTPPayload.DATA56 • CiscoRTPPayload.GSM • CiscoRTPPayload.ACTIVEVOICE • CiscoRTPPayload.WIDEBAND_256K
int	getPrecedenceValue()	優先順位の値を返します。
java.net.InetAddress	getRemoteAddress()	メディアの送信先アドレスを返します。
int	getRemotePort()	メディアの送信先ポートを返します。
boolean	getSilenceSuppression()	Cisco Unified Communication Manager サービス パラメータ「無音抑止」が False に設定されている場合は false を返し、そうでない場合は True を返します。

関連資料

CiscoRTPBitRate を参照してください。

CiscoRTPOutputStartedEv

CiscoRTPOutputStartedEv イベント インターフェイスはメディア転送の開始を示します。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

CiscoEv, CiscoTermEv, javax.telephony.events.Ev, javax.telephony.events.TermEv

宣言

public interface CiscoRTPOutputStartedEv extends CiscoTermEv

フィールド

表 6-132　CiscoRTPOutputStartedEv のフィールド

インターフェイス	フィールド	説明
static int	ID	なし

継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

表 6-133　CiscoRTPOutputStartedEv のメソッド

インターフェイス	メソッド	説明
CiscoRTPOutputProperties	getRTPOutputProperties()	RTP 出力プロパティを返します。
CiscoCallID	getCallID()	CiscoCallID を返します。CiscoRTPOutputStartedEv は CiscoCallID.getCall() に適用されます。
CiscoRTPHandle	getCiscoRTPHandle()	CiscoRTPHandle オブジェクトを返します。アプリケーションは、CiscoProvider.getCall を使用してコール参照を取得できます。コール オブザーバがない場合や、このイベントの配信時にコール オブザーバがなかった場合、CiscoProvider.getCall は null を返す可能性があります。
int	getMediaConnectionMode()	次のいずれかの値の mediaMode とともに CiscoMediaConnectionMode を返します。 • CiscoMediaConnectionMode.TRANSMIT_ONLY（一方向メディア、送信だけ） • CiscoMediaConnectionMode.TRANSMIT_AND_RECEIVE：（双方向メディア） **（注）** 通常、モード NONE のイベントを受け取ることはありません。受け取った場合は、そのイベントを無視してエラーを記録してください。

継承したメソッド

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.TermEv から
getTerminal

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

「定数フィールド値」（P.F-1）、CiscoCallID、および CiscoRTPParams を参照してください。

CiscoRTPOutputStoppedEv

CiscoRTPOutputStoppedEv イベントはメディア転送が停止したことを示します。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

CiscoEv, CiscoTermEv, javax.telephony.events.Ev, javax.telephony.events.TermEv

宣言

public interface CiscoRTPOutputStoppedEv extends CiscoTermEv

フィールド

表 6-134　CiscoRTPOutputStoppedEv のフィールド

インターフェイス	フィールド	説明
static int	ID	なし

継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

表 6-135　CiscoRTPOutputStoppedEv のメソッド

インターフェイス	メソッド	説明
CiscoCallID	getCallID()	CiscoCallID を返します。CiscoRTPOutputStoppedEv は CiscoCallID.getCall() に適用されます。
CiscoRTPHandle	getCiscoRTPHandle()	CiscoRTPHandle オブジェクトを返します。アプリケーションは、CiscoProvider.getCall を使用してコール参照を取得できます。コール オブザーバがない場合や、このイベントの配信時にコール オブザーバがなかった場合、CiscoProvider.getCall は null を返す可能性があります。
int	getMediaConnectionMode()	• 次のいずれかの値で CiscoMediaConnectionMode を返します。 • CiscoMediaConnectionMode.TRANSMIT_ONLY（一方向メディア、送信） • onlyCiscoMediaConnectionMode.TRANSMIT_AND_RECEIVE（双方向メディア） **（注）** 通常、モード NONE のイベントを受け取ることはありません。受け取った場合は、そのイベントを無視してエラーを記録してください。

継承したメソッド

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.TermEv から
getTerminal

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

「定数フィールド値」（P.F-1）、CiscoCallID、CiscoRTPParams、および CiscoMediaConnectionMode を参照してください。

CiscoRTPOutputKeyEv

CiscoRTPOutputKeyEv イベントは、暗号化された発信メディア（転送）ストリームの鍵情報を提供します。アプリケーションは CiscoTermEvFilter.setRTPKeyEventsEnabled(true) を使用してフィルタを設定し、TerminalObserver.terminalChangedEvent() を介してこのイベントを取得します。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

CiscoEv, CiscoTermEv, javax.telephony.events.Ev, javax.telephony.events.TermEv

宣言

public interface CiscoRTPOutputKeyEv extends CiscoTermEv

フィールド

表 6-136　CiscoRTPOutputKeyEv のフィールド

インターフェイス	フィールド	説明
static int	ID	なし

継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

表 6-137　CisctoRTPOutputKeyEv のメソッド

インターフェイス	メソッド	説明
CiscoCallID	getCallID()	このイベントの送信時に CiscoCall がすでに存在している場合、CiscoCallID オブジェクトを返します。CiscoCall がない場合、このメソッドは null を返します。
CiscoMedia EncryptionKeyInfo	getCiscoMediaEncryptionKeyInfo()	プロバイダーが TLS リンクで開かれており、Cisco Unified Communications Manager administration でアプリケーションの SRTP を有効にするオプションが設定されている場合にだけ、CiscoMediaEncryptionKeyInfo を返します。そうでない場合は null を返します。
int	getCiscoMediaSecurityIndicator()	次の定数の 1 つのメディアのセキュリティ インジケータを返します。 • CiscoMediaSecurityIndicator.MEDIA_ENCRYPTED_KEYS_AVAILABLE • CiscoMediaSecurityIndicator.MEDIA_ENCRYPT_USER_NOT_AUTHORIZED • CiscoMediaSecurityIndicator.MEDIA_ENCRYPTED_KEYS_UNAVAILABLE
CiscoRTPHandle	getCiscoRTPHandle()	CiscoRTPHandle オブジェクトを返します。アプリケーションは、CiscoProvider.getCall を使用してコール参照を取得できます。コール オブザーバがない場合や、このイベントの配信時にコール オブザーバがなかった場合、CiscoProvider.getCall は null を返す可能性があります。

継承したメソッド

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.TermEv から
getTerminal

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

「定数フィールド値」（P.F-1）、CiscoRTPParams、および CiscoMediaSecurityIndicator を参照してください。

CiscoRTPOutputProperties

CiscoRTPPutputProperties インターフェイスは、端末によって転送されたメディアのプロパティを提供します。CiscoRTPOutPutStartedEv はメディアが起動していることを示します。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

宣言

public interface CiscoRTPOutputProperties

フィールド

なし

メソッド

表 6-138　CiscoRTPOutputProperties のメソッド

インターフェイス	メソッド	説明
int	getBitRate()	次の定数の 1 つのメディア ビット レートを返します。 • CiscoRTPBitRate.R5_3 • CiscoRTPBitRate.R6_4
int	getMaxFramesPerPacket()	パケットあたりの最大送信フレーム数を返します。
int	getPacketSize()	パケット サイズ（ミリ秒単位）を返します。
int	getPayloadType()	次の定数の 1 つのペイロード形式を返します。 • CiscoRTPPayload.NONSTANDARD • CiscoRTPPayload.G711ALAW64K • CiscoRTPPayload.G711ALAW56K • CiscoRTPPayload.G711ULAW64K • CiscoRTPPayload.G711ULAW56K • CiscoRTPPayload.G722_64K • CiscoRTPPayload.G722_56K • CiscoRTPPayload.G722_48K • CiscoRTPPayload.G7231 • CiscoRTPPayload.G728 • CiscoRTPPayload.G729 • CiscoRTPPayload.G729ANNEXA • CiscoRTPPayload.IS11172AUDIOCAP • CiscoRTPPayload.IS13818AUDIOCAP • CiscoRTPPayload.ACY_G729AASSN • CiscoRTPPayload.DATA64 • CiscoRTPPayload.DATA56 • CiscoRTPPayload.GSM • CiscoRTPPayload.ACTIVEVOICE • CiscoRTPPayload.WIDEBAND_256K
int	getPrecedenceValue()	優先順位の値を返します。
java.net.InetAddress	getRemoteAddress()	メディアの送信先アドレスを返します。
int	getRemotePort()	メディアの送信先ポートを返します。
boolean	getSilenceSuppression()	Cisco Unified Communication Manager サービス パラメータ「無音抑止」が False に設定されている場合は false を返し、そうでない場合は True を返します。

関連資料

CiscoRTPBitRate を参照してください。

CiscoRTPOutputStartedEv

CiscoRTPOutputStartedEv イベント インターフェイスはメディア転送の開始を示します。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

CiscoEv, CiscoTermEv, javax.telephony.events.Ev, javax.telephony.events.TermEv

宣言

public interface CiscoRTPOutputStartedEv extends CiscoTermEv

フィールド

表 6-139　CiscoRTPOutputStartedEv のフィールド

インターフェイス	フィールド	説明
static int	ID	なし

継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

表 6-140　CiscoRTPOutputStartedEv のメソッド

インターフェイス	メソッド	説明
CiscoRTPOutputProperties	getRTPOutputProperties()	RTP 出力プロパティを返します。
CiscoCallID	getCallID()	CiscoCallID を返します。CiscoRTPOutputStartedEv は CiscoCallID.getCall() に適用されます。
CiscoRTPHandle	getCiscoRTPHandle()	CiscoRTPHandle オブジェクトを返します。アプリケーションは、CiscoProvider.getCall を使用してコール参照を取得できます。コール オブザーバがない場合や、このイベントの配信時にコール オブザーバがなかった場合、CiscoProvider.getCall は null を返す可能性があります。
int	getMediaConnectionMode()	次のいずれかの値の mediaMode とともに CiscoMediaConnectionMode を返します。 • CiscoMediaConnectionMode.TRANSMIT_ONLY（一方向メディア、送信だけ） • CiscoMediaConnectionMode.TRANSMIT_AND_RECEIVE（双方向メディア） **（注）** 通常、モード NONE のイベントを受け取ることはありません。受け取った場合は、そのイベントを無視してエラーを記録してください。

継承したメソッド

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.TermEv から
getTerminal

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

「定数フィールド値」（P.F-1）、CiscoCallID、および CiscoRTPParams を参照してください。

CiscoRTPOutputStoppedEv

CiscoRTPOutputStoppedEv イベントはメディア転送が停止したことを示します。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

CiscoEv, CiscoTermEv, javax.telephony.events.Ev, javax.telephony.events.TermEv

宣言

public interface CiscoRTPOutputStoppedEv extends CiscoTermEv

フィールド

表 6-141　CiscoRTPOutputStoppedEv のフィールド

インターフェイス	フィールド	説明
static int	ID	なし

継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

表 6-142　CiscoRTPOutputStoppedEv のメソッド

インターフェイス	メソッド	説明
CiscoCallID	getCallID()	CiscoCallID を返します。CiscoRTPOutputStoppedEv は CiscoCallID.getCall() に適用されます。
CiscoRTPHandle	getCiscoRTPHandle()	CiscoRTPHandle オブジェクトを返します。アプリケーションは、CiscoProvider.getCall を使用してコール参照を取得できます。コール オブザーバがない場合や、このイベントの配信時にコール オブザーバがなかった場合、CiscoProvider.getCall は null を返す可能性があります。
int	getMediaConnectionMode()	• 次のいずれかの値で CiscoMediaConnectionMode を返します。 • CiscoMediaConnectionMode.TRANSMIT_ONLY（一方向メディア、送信） • onlyCiscoMediaConnectionMode.TRANSMIT_AND_RECEIVE（双方向メディア） **（注）** 通常、モード NONE のイベントを受け取ることはありません。受け取った場合は、そのイベントを無視してエラーを記録してください。

継承したメソッド

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.TermEv から
getTerminal

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

「定数フィールド値」（P.F-1）、CiscoCallID、CiscoRTPParams、および CiscoMediaConnectionMode を参照してください。

CiscoRTPPayload

RTPPayload インターフェイスには、RTP 形式を記述する定数が入ります。CiscoRTPInputProperties.getPayloadType と CiscoRTPOutputProperties.getPayloadType メソッドはこれらの定数を返します。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

宣言

public interface CiscoRTPPayload

フィールド

表 6-143　CiscoRTPPayload のフィールド

インターフェイス	フィールド	説明
static final int	NONSTANDARD	標準外の RTP ペイロード
static final int	G711ALAW64K	G.711 64K a-law ペイロード
static final int	G711ALAW56K	G.711 56K a-law ペイロ [ド
static final int	G711ULAW64K	G.711 64K u-law ペイロード
static final int	G711ULAW56K	G.711 56K u-law ペイロード
static final int	G722_64K	G.722 64K ペイロード
static final int	G722_56K	G.722 56K ペイロード
static final int	G722_48K	G.722 48K ペイロード
static final int	G7231	G.723.1 ペイロード
static final int	G728	G.728 ペイロード
static final int	G729	G.729 ペイロード

表 6-143　CiscoRTPPayload のフィールド （続き）

インターフェイス	フィールド	説明
static final int	G729ANNEXA	G.729a ペイロード
static final int	IS11172AUDIOCAP	IS11172AUDIOCAP ペイロード
static final int	IS13818AUDIOCAP	IS13818AUDIOCAP ペイロード
static final int	ACY_G729AASSN	ACY_G729AASSN ペイロード
static final int	DATA64	DATA64 ペイロード
static final int	DATA56	DATA56 ペイロード
static final int	GSM	GSM ペイロード
static final int	ACTIVEVOICE	ACTIVEVOICE ペイロード
static final int	WIDEBAND_256K	Wide_Band_256k ペイロード

メソッド

なし

関連資料

CiscoRTPInputProperties.getPayloadType()、CiscoRTPOutputProperties.getPayloadType() および 「定数フィールド値」（P.F-1）を参照してください。

CiscoSynchronousObserver

Cisco JTAPI 実装は、アプリケーションによって Call.connect() と TerminalConnection.answer() などのブロッキング用 JTAPI メソッドを、そのオブザーバのコールバック内部から呼び出せる設計になっています。これは、オブザーバのコールバック内部から JTAPI メソッドを使用しないように警告する JTAPI 仕様の制限に、アプリケーションが制約されないことを意味します。

通常、アプリケーションが JTAPI オブジェクトに新規のオブザーバを追加すると、Cisco JTAPI 実装は、新規のオブザーバへのサービスを行うためのイベント キューとそれに付随する作業スレッドを作成します。同じオブザーバが別のオブジェクトに追加された場合、そのキューとスレッドが再利用されます。実質、オブザーバごとに、単一のキューと作業スレッドがあります。前述したとおり、この方式の利点は、アプリケーションがオブザーバ コールバック内からブロッキング用 JTAPI メソッドを呼び出すことが可能である点です。ただし、微妙な欠点は、Call.getConnections() および Connection.getState() などのアクセス用メソッドが、オブザーバ コールバック内から呼び出されたときに、イベントと一貫する結果を返さないことがある点です。

たとえば、アプリケーションが、アドレス「A」からアドレス「B」へのコールを作成し接続したとします。アプリケーションがアドレス「A」を監視している場合、CallActiveEv を受け取ったときには、そのコールの状態が当然 Call.ACTIVE になると考えられます。しかし、必ずしも真であるとは限りません。これは、アプリケーションにイベントを配送する作業スレッドが、オブジェクトの状態を更新し

ている内部 JTAPI スレッドから切り離されているためです。事実、「B」が「A」からのコールを拒絶した場合、作業スレッドが CallActiveEv を配送する時点に応じて、コール オブジェクトは、Call.ACTIVE または Call.INVALID のいずれかの状態にあります。

多くのアプリケーションは、この非同期動作の悪影響を受けることはありません。ただし、オブザーバ コールバック中にコヒーレント コール モデルの機能を使用するアプリケーションでは、Cisco JTAPI 実装のキューイング ロジックを選択的に無効にできます。対象のオブザーバ オブジェクト上に CiscoSynchronousObserver インターフェイスを実装するアプリケーションは、そのオブザーバへのイベント配送の同期を取ることを宣言します。同期オブザーバに配送されるイベントは、オブザーバ コールバックの内部から照会されるコール モデル オブジェクトの状態と一致します。

CiscoSynchronousObserver インターフェイスの実装されたオブジェクトでは、そのイベント コールバック内部からブロッキング用 JTAPI メソッドを呼び出さないでください。これを行った場合の影響は予測不能であり、JTAPI 実装が停止する可能性もあります。一方、Call.getConnections() または Connection.getState() などのすべての JTAPI オブジェクトのアクセス用メソッドは安全に使用できます。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

宣言

public interface CiscoSynchronousObserver

フィールド

なし

メソッド

なし

関連資料

なし

CiscoTermActivatedEv

端末が監視され、Restriction 状態がアクティブに変更された場合、このイベントがアプリケーションに送信されます。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

CiscoEv, CiscoProvEv, javax.telephony.events.Ev, javax.telephony.events.ProvEv

宣言

public interface CiscoTermActivatedEv extends CiscoProvEv

フィールド

表 6-144　CiscoTermActivatedEv のフィールド

インターフェイス	フィールド	説明
static int	ID	なし

継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

表 6-145　CiscoTermActivatedEv のメソッド

インターフェイス	メソッド	説明
javax.telephony.Terminal	getTerminal()	有効化された端末を返します。

継承したメソッド

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.ProvEv から
getProvider

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

「定数フィールド値」（P.F-1）を参照してください。

CiscoTermButtonPressedEv

CiscoTermButtonPressedEv イベントは、Terminal でボタンが押されたときに TerminalObserver に対して配信されます。アプリケーションがイベントを受信するには、ciscoTermEvFilter.setButtonPressedEnabled(true) を使用してフィルタを設定する必要があります。ボタン押下イベントは、このインターフェイスの定数（0 ～ 9、*、#、A、B、C、D など）に挙げられている数値キーパッド ボタンが Terminal で押されたときにだけ応答します。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

CiscoEv, CiscoTermEv, javax.telephony.events.Ev, javax.telephony.events.TermEv

宣言

public interface CiscoTermButtonPressedEv extends CiscoTermEv

フィールド

表 6-146　CiscoTermButtonPressedEv のフィールド

インターフェイス	フィールド
static int	CHARA
static int	CHARB
static int	CHARC
static int	CHARD
static int	EIGHT
static int	FIVE
static int	FOUR
static int	ID
static int	NINE
static int	ONE
static int	POUND
static int	SEVEN
static int	SIX
static int	STAR
static int	THREE
static int	TWO
static int	ZERO

継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

表 6-147　CiscoTermButtonPressedEv のメソッド

インターフェイス	メソッド	説明
int	getButtonPressed ()	端末上で押されたボタン

継承したメソッド

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.TermEv から
getTerminal

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

CiscoTermEvFilter と 「定数フィールド値」（P.F-1）を参照してください。

CiscoTermConnMonitoringEndEv

コールに対する監視が停止した場合、またはコールが接続解除された場合に、システムは CiscoTermConnMonitoringEndEv イベントをコール オブザーバに配信します。

インターフェイス履歴

Cisco Unified Communications Manager リリース	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

javax.telephony.events.CallEv, javax.telephony.events.Ev, javax.telephony.events.TermConnEv

宣言

public interface CiscoTermConnMonitoringEndEv extends javax.telephony.events.TermConnEv

フィールド

表 6-148　CiscoTermConnMonitoringEndEv のフィールド

インターフェイス	フィールド
static int	ID

継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

表 6-149　CiscoTermConnMonitoringEndEv のメソッド

インターフェイス	メソッド	説明
Int	getMonitorType()	監視のタイプを返します。返される値は常に CiscoCall.SILENT_MONITOR です。

継承したメソッド

インターフェイス javax.telephony.events.TermConnEv から
getTerminalConnection

インターフェイス javax.telephony.events.CallEv から
getCall

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

詳細については、「定数フィールド値」（P.F-1）を参照してください。

CiscoTermConnMonitoringStartEv

コールに対する監視が開始されたとき、システムは CiscoTermConnMonitoringStartEv イベントをコール オブザーバに配信します。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

javax.telephony.events.CallEv, javax.telephony.events.Ev, javax.telephony.events.TermConnEv

宣言

public interface CiscoTermConnMonitoringStartEv extends javax.telephony.events.TermConnEv

フィールド

表 6-150　CiscoTermConnMonitoringStartEv のフィールド

インターフェイス	フィールド
static final int	ID

継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE,

CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

表 6-151　CiscoTermConnMonitoringStartEv のメソッド

インターフェイス	メソッド	説明
int	getMonitorType()	監視のタイプを返します。返される値は常に CiscoCall.Silent_Monitor です。

継承したメソッド

インターフェイス javax.telephony.events.TermConnEv から
getTerminalConnection

インターフェイス javax.telephony.events.CallEv から
getCall

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

詳細については、「定数フィールド値」（P.F-1）を参照してください。

CiscoTermConnMonitorInitiatorInfoEv

CiscoTermConnMonitorInitiatorInfoEv イベント インターフェイスは TermConnEv インターフェイスを拡張し、モニタリングのターゲット（エージェント）上の CallObserver によって報告されます。このインターフェイスは、モニタリング セッションが確立されたときに、モニタリングの開始側（スーパーバイザ）に関する情報を提供します。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

javax.telephony.events.CallEv, javax.telephony.events.Ev, javax.telephony.events.TermConnEv

宣言

public interface CiscoTermConnMonitorInitiatorInfoEv extends javax.telephony.events.TermConnEv

フィールド

表 6-152　CiscoTermConnMonitorInitiatorInfoEv のフィールド

インターフェイス	フィールド
static int	ID

継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

表 6-153　CiscoTermConnMonitorInitiatorInfoEv のメソッド

インターフェイス	メソッド	説明
CiscoMonitorInitiatorInfo	getCiscoMonitorInitiatorInfo()	モニタリングの開始側の端末名とアドレスを返します。

継承したメソッド

インターフェイス javax.telephony.events.TermConnEv から
getTerminalConnection

インターフェイス javax.telephony.events.CallEv から
getCall

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

「定数フィールド値」（P.F-1）を参照してください。

CiscoTermConnMonitorTargetInfoEv

CiscoTermConnMonitorTargetInfoEv イベント インターフェイスは TermConnEv インターフェイスを拡張し、モニタリングの開始側の CallObserver によって報告されます。このインターフェイスは、モニタリング セッションが確立されたときに、モニタリング ターゲット（エージェント）に関する情報を提供します。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

javax.telephony.events.CallEv, javax.telephony.events.Ev, javax.telephony.events.TermConnEv

宣言

public interface CiscoTermConnMonitorTargetInfoEv extends javax.telephony.events.TermConnEv

フィールド

表 6-154　CiscoTermConnMonitorTargetInfoEv のフィールド

インターフェイス	フィールド
static int	ID

継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY,

META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

表 6-155　CiscoTermConnMonitorTargetInfoEv のメソッド

インターフェイス	メソッド	説明
CiscoMonitorTargetInfo	getCiscoMonitorTargetInfo()	モニタリング ターゲットの端末名とアドレスを返します。

継承したメソッド

インターフェイス javax.telephony.events.TermConnEv から
getTerminalConnection

インターフェイス javax.telephony.events.CallEv から
getCall

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

「定数フィールド値」（P.F-1）を参照してください。

CiscoTermConnPrivacyChangedEv

TeminalConnection のプライバシー ステータスが変更されると、CiscoTermConnPrivacyChangedEv イベントが送信されます。プライバシーがオンの場合は、ユーザはコールに割り込みできません。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

宣言

public interface CiscoTermConnPrivacyChangedEv

フィールド

表 6-156　CiscoTermConnPrivacyChangedEv のフィールド

インターフェイス	フィールド
static int	ID

メソッド

表 6-157　CiscoTermConnPrivacyChangedEv のメソッド

インターフェイス	メソッド	説明
javax.telephony.TerminalConnection	getTerminalConnection()	プライバシーが変更された TerminalConnection を返します。TerminalConnection で getPrivacyStatus を呼び出して、プライバシー ステータスを確認することができます。

関連資料

「定数フィールド値」（P.F-1）と CiscoTerminalConnection.getPrivacyStatus() を参照してください。

CiscoTermConnRecordingEndEv

コール録音が停止したとき、JTAPI は CiscoTermConnRecordingEndEv をコール オブザーバに配信します。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

javax.telephony.events.CallEv, javax.telephony.events.Ev, javax.telephony.events.TermConnEv

宣言

public interface CiscoTermConnRecordingEndEv extends javax.telephony.events.TermConnEv

フィールド

static intID

継承したフィールド

インターフェイス javax.telephony.events.Ev から継承したフィールド

CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

なし

継承したメソッド

インターフェイス javax.telephony.events.TermConnEv から
getTerminalConnection

インターフェイス javax.telephony.events.CallEv から
getCall

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

「定数フィールド値」（P.F-1）を参照してください。

CiscoTermConnRecordingStartEv

コール録音が開始されたとき、JTAPI は CiscoTermConnRecordingStartEv をコール オブザーバに配信します。

スーパーインターフェイス

javax.telephony.events.CallEv, javax.telephony.events.Ev, javax.telephony.events.TermConnEv

宣言

public interface CiscoTermConnRecordingStartEv extends javax.telephony.events.TermConnEv

フィールド

表 6-158　CiscoTermConnRecordingStartEv のフィールド

インターフェイス	フィールド
static int	ID

継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

なし

継承したメソッド

インターフェイス javax.telephony.events.TermConnEv から
getTerminalConnection

インターフェイス javax.telephony.events.CallEv から
getCall

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

「定数フィールド値」（P.F-1）を参照してください。

CiscoTermConnRecordingTargetInfoEv

JTAPI は CiscoTermConnRecordingTargetInfoEv を録音元のコール オブザーバに配信します。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

javax.telephony.events.CallEv, javax.telephony.events.Ev, javax.telephony.events.TermConnEv

宣言

public interface CiscoTermConnRecordingTargetInfoEv extends javax.telephony.events.TermConnEv

フィールド

表 6-159　CiscoTermConnRecordingTargetInfoEv のフィールド

インターフェイス	フィールド
static int	ID

継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

表 6-160　CiscoTermConnRecordingTargetInfoEv のメソッド

インターフェイス	メソッド	説明
CiscoRecorderInfo	getCiscoRecorderInfo()	録音デバイスの端末名およびアドレスを提供する CiscoRecorderInfo を返します。

インターフェイス javax.telephony.events.TermConnEv から
getTerminalConnection

インターフェイス javax.telephony.events.CallEv から
getCall

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

「定数フィールド値」（P.F-1）および CiscoRecorderInfo を参照してください。

CiscoTermConnSelectChangedEv

TerminalConnection のコール選択ステータスが変更されると、JTAPI は CiscoTermConnSelectChangedEv を、機能の呼び出しまたは手動によって送信します。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

javax.telephony.events.CallEv, javax.telephony.events.Ev, javax.telephony.events.TermConnEv

宣言

public interface CiscoTermConnSelectChangedEv extends javax.telephony.events.TermConnEv

フィールド

表 6-161　CiscoTermConnSelectChangedEv のフィールド

インターフェイス	フィールド
static int	ID

継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

なし

継承したメソッド

インターフェイス javax.telephony.events.TermConnEv から
getTerminalConnection

インターフェイス javax.telephony.events.CallEv から
getCall

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

「定数フィールド値」（P.F-1）を参照してください。

CiscoTermCreatedEv

CiscoTerminal がプロバイダー ドメインに追加されると、JTAPI は CiscoTermCreatedEv イベントをアプリケーションのプロバイダー オブザーバに送信します。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

CiscoEv, CiscoProvEv, javax.telephony.events.Ev, javax.telephony.events.ProvEv

宣言

public interface CiscoTermCreatedEv extends CiscoProvEv

フィールド

表 6-162　CiscoTermEv のフィールド

インターフェイス	フィールド
static int	ID

継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

表 6-163　CiscoTermCreatedEv のメソッド

インターフェイス	メソッド	説明
javax.telephony.Terminal	getTerminal()	このイベントが送信された端末オブジェクトを返します。

継承したメソッド

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.ProvEv から
getProvider

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

「定数フィールド値」（P.F-1）を参照してください。

CiscoTermDataEv

電話機が XSI データ（XML オブジェクト）を受信すると、JTAPI は CiscoTermDataEv イベントを端末オブザーバに送信します。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

CiscoEv, CiscoTermEv, javax.telephony.events.Ev, javax.telephony.events.TermEv

宣言

public interface CiscoTermDataEv extends CiscoTermEv

フィールド

表 6-164　CiscoTermDataEv のフィールド

インターフェイス	フィールド
static int	ID

継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

表 6-165　CiscoTermDataEv のメソッド

インターフェイス	メソッド	説明
java.lang.String	getData()	推奨されません。byte[] getTermData を使用してください。
byte[]	getTermData()	電話機が受信した XSI データに対応する XML で符号化されたバイト配列を返します。

継承したメソッド

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.TermEv から
getTerminal

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

「定数フィールド値」（P.F-1）を参照してください。

CiscoTermDeviceStateActiveEv

端末のいずれかのアドレスにあらゆる状態の発信コールがある場合、または TerminalConnection が ACTIVE 状態で CallCtlTerminalConnection が TALKING 状態の着信コールがある場合には、CiscoTermDeviceStateActiveEv イベントがアプリケーションに送信されます。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

CiscoEv, CiscoTermEv, javax.telephony.events.Ev, javax.telephony.events.TermEv

宣言

public interface CiscoTermDeviceStateActiveEv extends CiscoTermEv

フィールド

表 6-166　CiscoTermDeviceStateActiveEv のフィールド

インターフェイス	フィールド
static int	ID

継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

なし

継承したメソッド

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.TermEv から
getTerminal

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

「定数フィールド値」（P.F-1）を参照してください。

CiscoTermDeviceStateAlertingEv

端末上に CallCtlConnection.Established 状態の接続で発信コールまたは着信コールがあるアドレスがなく、端末上の少なくとも 1 つのアドレスに CallCtlConnection.OFFERED 状態または CallCtlConnection.ALERTING 状態の接続で着信コールがある場合、CiscoTermDeviceStateAlertingEv イベントが 端末オブザーバに送信されます。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

CiscoEv, CiscoTermEv, javax.telephony.events.Ev, javax.telephony.events.TermEv

宣言

public interface CiscoTermDeviceStateAlertingEv extends CiscoTermEv

フィールド

表 6-167　CiscoTermDeviceStateAlertingEv のフィールド

インターフェイス	フィールド
static int	ID

継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE,

CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

なし

継承したメソッド

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.TermEv から
getTerminal

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

「定数フィールド値」（P.F-1）を参照してください。

CiscoTermDeviceStateHeldEv

端末のアドレスのすべてのコールが CallCtlTerminalConnection.HELD 状態の TerminalConnection の場合は、CiscoTermDeviceStateHeldEv イベントが端末オブザーバに送信されます。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

CiscoEv, CiscoTermEv, javax.telephony.events.Ev, javax.telephony.events.TermEv

宣言

public interface CiscoTermDeviceStateHeldEv extends CiscoTermEv

フィールド

表 6-168　CiscoTermDeviceStateHeldEv のフィールド

インターフェイス	フィールド
static int	ID

継承したフィールド

インターフェイス javax.telephony.events.Ev から継承したフィールド

CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から

CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

なし

継承したメソッド

インターフェイス javax.telephony.events.Ev から

getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.TermEv から

getTerminal

インターフェイス javax.telephony.events.Ev から

getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

「定数フィールド値」（P.F-1）を参照してください。

CiscoTermDeviceStateIdleEv

端末のどのアドレスにもコールがない場合は、CiscoTermDeviceStateIdleEv が端末オブザーバに送信されます。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

CiscoEv, CiscoTermEv, javax.telephony.events.Ev, javax.telephony.events.TermEv

宣言

public interface CiscoTermDeviceStateIdleEv extends CiscoTermEv

フィールド

表 6-169　CiscoTermDeviceStateIdleEv のフィールド

インターフェイス	フィールド
static int	ID

継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

なし

継承したメソッド

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.TermEv から
getTerminal

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

「定数フィールド値」（P.F-1）を参照してください。

CiscoTermDeviceStateWhisperEv

CiscoTermDeviceStateActiveEv イベントは、端末上の少なくとも 1 つのアドレスがインターコム ターゲットで、TerminalConnection または CallCtlTerminalConneciton の状態が Passive または Bridged であるインターコム コールが存在する場合に端末オブザーバ送信されます。この状態では、ユーザは新しい発信コールを開始したり、新しい着信コールを受けられます。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

CiscoEv, CiscoTermEv, javax.telephony.events.Ev, javax.telephony.events.TermEv

宣言

public interface CiscoTermDeviceStateWhisperEv extends CiscoTermEv

フィールド

表 6-170　CiscoTermDeviceStateWhisperEv のフィールド

インターフェイス	フィールド
static int	ID

継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

なし

継承したメソッド

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.TermEv から
getTerminal

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

「定数フィールド値」（P.F-1）を参照してください。

CiscoTermDNDOptionChangedEv

CiscoTermDNDOptionChangedEv イベントは、DND（サイレント）オプションが変更された場合に端末オブザーバに送信されます。このイベントは、アプリケーションでイベントを受信するフィルタが有効になっている場合にだけ送信されます。このイベントは、アプリケーション オブザーバで提供されます。

履歴

Cisco Unified Communications Manager リリース	説明
7.0(1)	拡張が追加されました。

スーパーインターフェイス

CiscoEv, CiscoTermEv, javax.telephony.events.Ev, javax.telephony.events.TermEv

public interface CiscoTermDNDOptionChangedEv

extends CiscoTermEv

フィールド

表 6-171　CiscoTermDNDOptionChangedEv のフィールド

インターフェイス	フィールド
static final int	ID

表 6-172　継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

表 6-173　継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

表 6-174　CiscoTermDNDOptionChangedEv のメソッド

インターフェイス	メソッド	説明
Int	getDNDOption()	このインターフェイスは、アプリケーションに現在の DND（サイレント）オプションを返します。int dndOption を返します。

表 6-175　継承したメソッド

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

表 6-176　継承したメソッド

インターフェイス javax.telephony.events.TermEv から
getTerminal

表 6-177　継承したメソッド

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

「定数フィールド値」（P.F-1）と CiscoTermEv を参照してください。

CiscoTermDNDStatusChangedEv

CiscoTermDNDStatusChangedEv イベントは、DND（サイレント）のステータスが変更された場合に端末オブザーバに送信されます。ただし、アプリケーションでイベントを受信するフィルタが有効になっている必要があります。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

CiscoEv, CiscoTermEv, javax.telephony.events.Ev, javax.telephony.events.TermEv

宣言

public interface CiscoTermDNDStatusChangedEv extends CiscoTermEv

フィールド

表 6-178　CiscoTermDNDStatusChangedEv のフィールド

インターフェイス	フィールド
static int	ID

継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

表 6-179　CiscoTermDNDStatusChangedEv のメソッド

インターフェイス	メソッド	説明
boolean	getDNDStatus()	アプリケーションに現在の DND（サイレント）ステータスを返します。

継承したメソッド

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.TermEv から
getTerminal

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

CiscoTermEvFilter、CiscoTermEv および「定数フィールド値」（P.F-1）を参照してください。

CiscoTermEv

JTAPI のコアである javax.telephony.events.TermEv インターフェイスを拡張する CiscoTermEv インターフェイスは、Cisco によって拡張されたすべての JTAPI Terminal イベントの基本インターフェイスになります。このパッケージのコール関連イベントはすべて、直接的または間接的にこのインターフェイスを拡張します。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

CiscoEv, javax.telephony.events.Ev, javax.telephony.events.TermEv

サブインターフェイス

CiscoMediaOpenLogicalChannelEv, CiscoRTPInputKeyEv, CiscoRTPInputStartedEv, CiscoRTPInputStoppedEv, CiscoRTPOutputKeyEv, CiscoRTPOutputStartedEv, CiscoRTPOutputStoppedEv, CiscoTermButtonPressedEv, CiscoTermDataEv, CiscoTermDeviceStateActiveEv, CiscoTermDeviceStateAlertingEv, CiscoTermDeviceStateHeldEv, CiscoTermDeviceStateIdleEv, CiscoTermDeviceStateWhisperEv, CiscoTermDNDOptionChangedEv, CiscoTermDNDStatusChangedEv, CiscoTermInServiceEv, CiscoTermOutOfServiceEv, CiscoTermRegistrationFailedEv, CiscoTermSnapshotCompletedEv, CiscoTermSnapshotEv

宣言

public interface CiscoTermEv extends CiscoEv, javax.telephony.events.TermEv

フィールド

なし

継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

なし

継承したメソッド

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.TermEv から
getTerminal

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

CallEv を参照してください。

CiscoTermEvFilter

アプリケーションは、CiscoTermEvFilter インターフェイスを使用して、関係のない Terminal イベントを選択的に制限できます。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

宣言

public interface CiscoTermEvFilter

フィールド

なし

メソッド

表 6-180　CiscoTermEvFilter のメソッド

インターフェイス	メソッド	説明
boolean	getDeviceDataEnabled()	端末の CiscoTermDataEv イベントの CiscoTermSnapshotCompletedEv のイベントフィルタのステータスを返します。デフォルトでは、無効に設定されています。イベント フィルタが有効な場合は true を返します。イベントフィルタが無効な場合は false を返します。
void	setDeviceDataEnabled(boolean enabled)	端末の CiscoTermDataEv イベントを有効または無効に設定します。
boolean	getButtonPressedEnabled()	端末の CiscoTermButtonPressedEv イベントの CiscoTermSnapshotCompletedEv のイベントフィルタのステータスを返します。デフォルトでは、無効に設定されています。イベント フィルタが有効な場合は true を返します。イベントフィルタが無効な場合は false を返します。
void	setButtonPressedEnabled(boolean enabled)	端末の CiscoTermButtonPressedEv イベントを有効または無効に設定します。
boolean	getRTPEventsEnabled()	端末の CiscoRTPInputStartedEv、CiscoRTPOutputStartedEv、CiscoRTPInputStoppedEv、および CiscoRTPOutputStoppedEv イベントのイベントフィルタのステータスを返します。デフォルトでは、無効に設定されています。イベント フィルタが有効な場合は true を返します。イベントフィルタが無効な場合は false を返します。
void	setRTPEventsEnabled(boolean enabled)	端末の CiscoRTPInputStartedEv、CiscoRTPOutputStartedEv、CiscoRTPInputStoppedEv、および CiscoRTPOutputStoppedEv イベントを有効または無効に設定します。

表 6-180　CiscoTermEvFilter のメソッド （続き）

インターフェイス	メソッド	説明
boolean	getSnapshotEnabled()	端末の CiscoTermSnapshotEv および CiscoTermSnapshotCompletedEv イベントのイベント フィルタのステータスを返します。無効な場合、CiscoTermSnapshotEv および CiscoTermSnapshotCompletedEv はアプリケーションに送信されません。イベント フィルタが有効な場合は true を返します。イベント フィルタが無効な場合は false を返します。
void	setSnapshotEnabled(boolean enabled)	端末の CiscoTermSnapshotEv および CiscoTermSnapshotCompletedEv を有効または無効に設定します。
boolean	getRTPKeyEventsEnabled()	端末の CiscoRTPInputKeyEv および CiscoRTPOutputKeyEv イベントのイベント フィルタのステータスを返します。イベント フィルタが有効な場合は true を返します。イベント フィルタが無効な場合は false を返します。
void	setRTPKeyEventsEnabled(boolean enabled)	端末の CiscoRTPInputKeyEv および CiscoRTPOutputKeyEv イベントを有効または無効に設定します。
boolean	getDeviceStateActiveEvFilter()	端末の CiscoTermDeviceStateActiveEv イベントの CiscoTermSnapshotCompletedEv のイベント フィルタのステータスを返します。イベント フィルタが有効な場合は true を返します。イベント フィルタが無効な場合は false を返します。
boolean	getDeviceStateHeldEvFilter()	端末の CiscoTermDeviceStateHeldEv イベントの CiscoTermSnapshotCompletedEv のイベント フィルタのステータスを返します。イベント フィルタが有効な場合は true を返します。イベント フィルタが無効な場合は false を返します。
boolean	getDeviceStateAlertingEvFilter()	端末の CiscoTermDeviceStateAlerting イベントの CiscoTermSnapshotCompletedEv のイベント フィルタのステータスを返します。イベント フィルタが有効な場合は true を返します。イベント フィルタが無効な場合は false を返します。
boolean	getDeviceStateIdleEvFilter()	CiscoTermDeviceStateIdleEv フィルタの状態を返します。イベント フィルタが有効な場合は true を返します。イベント フィルタが無効な場合は false を返します。
void	setDeviceStateActiveEvFilter(boolean filterValue)	端末に対して CiscoTermDeviceStateActiveEv フィルタを有効または無効にします。デフォルトでは、無効に設定されています。

表 6-180　CiscoTermEvFilter のメソッド （続き）

インターフェイス	メソッド	説明
void	setDeviceStateHeldEvFilter(boolean filterValue)	端末に対して CiscoTermDeviceStateHeldEv フィルタを有効または無効にします。デフォルトでは、無効に設定されています。
void	setDeviceStateAlertingEvFilter(boolean filter Value)	端末に対して CiscoTermDeviceStateAlertingEv フィルタを有効または無効にします。デフォルトでは、無効に設定されています。
void	setDeviceStateIdleEvFilter(boolean filterValue)	端末に対して CiscoTermDeviceStateIdleEv フィルタを有効または無効にします。デフォルトでは、無効に設定されています。
boolean	getDeviceStateWhisperEvFilter()	端末の CiscoTermDeviceStateWhisperEv フィルタのステータスを返します。
boolean	getDNDChangedEvFilter()	端末の CiscoTermDNDStatusChangedEv フィルタのステータスを返します。
void	setDNDChangedEvFilter(boolean filterValue)	端末に対して CiscoTermDNDStatusChangedEv フィルタを有効または無効にします。パラメータ：filterValue - void setDeviceStateWhisperEvFilter(boolean filterValue)。端末に対して CiscoTermDeviceStateWhisperEv フィルタを有効または無効にします。デフォルトでは、無効に設定されています。
boolean	getDNDOptionChangedEvFilter()	このインターフェイスは CiscoTermDNDOptionChangedEv フィルタのステータスを送信するために使用できます。端末の CiscoTermDNDOptionChangedEv フィルタのステータスを返します。
void	setDNDOptionChangedEvFilter(boolean filter Value)	このインターフェイスは、端末に対して CiscoTermDNDOptionChangedEv フィルタを有効または無効にするために使用します。パラメータ：filterValue

関連資料

なし

CiscoTerminal

標準の JTAPI では動的な端末登録の概念をサポートしていません。CiscoTerminal インターフェイスは、これをサポートするために、標準の端末インターフェイスを拡張します。すべての Cisco Unified Communications Manager デバイスは CiscoTerminal で表します。また、すべての CiscoTerminal について、現在 IN_SERVICE であるか OUT_OF_SERVICE であるかを照会できます。

CiscoTerminal が表す Cisco Unified Communications Manager デバイスが、たとえば IP フォンである場合、そのネットワーク接続が失われると、電話は OUT_OF_SERVICE になります。CiscoMediaTerminal などのその他のデバイスは、アプリケーションの要求に応じて登録され、IN_SERVICE または OUT_OF_SERVICE になります。

CiscoTerminal には API getIPAddressingMode() が含まれています。このインターフェイスは、CiscoTerminal に設定されている IP アドレッシング モードを返します。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

CiscoObjectContainer, javax.telephony.Terminal

サブインターフェイス

CiscoMediaTerminal, CiscoRouteTerminal

宣言

public interface CiscoTerminal extends javax.telephony.Terminal, CiscoObjectContainer

フィールド

表 6-181　CiscoTerminal のフィールド

インターフェイス	フィールド	説明
static final int	OUT_OF_SERVICE	CiscoTerminal の getState() インターフェイスによって返されるこの定数フィールド値は、CiscoTerminal がアウトオブサービスであることを示します。
static final int	IN_SERVICE	CiscoTerminal の getState() インターフェイスによって返されるこの定数値は、CiscoTerminal がインサービスであることを示します。
static final int	DEVICESTATE_IDLE	CiscoTerminal の getDeviceState() インターフェイスによって返されるこの定数値は、現在、CiscoTerminal のどのアドレスにもコールがないことを示します。
static final int	DEVICESTATE_ACTIVE	CiscoTerminal の getDeviceState() インターフェイスによって返されるこの定数値は、CiscoTerminal のアドレスに少なくとも 1 つのアクティブなコールがあることを示します。

表 6-181　CiscoTerminal のフィールド （続き）

インターフェイス	フィールド	説明
static final int	DEVICESTATE_ALERTING	CiscoTerminal の getDeviceState() インターフェイスによって返されるこの定数値は、CiscoTerminal のアドレスに少なくとも 1 つがアラート状態になっているコールがあり、アクティブなコールがないことを示します。
static final int	DEVICESTATE_HELD	CiscoTerminal の getDeviceState() インターフェイスによって返されるこの定数値は、CiscoTerminal のアドレスに少なくとも 1 つのコールが保留中であり、アラート状態になっているコールやアクティブなコールがないことを示します。
static final int	DEVICESTATE_UNKNOWN	CiscoTerminal の getDeviceState() インターフェイスによって返されるこの定数値は、CiscoTerminal DeviceState が不明であることを示します。この状態は、デバイス ステート フィルタが無効な場合に返されます。
static final int	DEVICESTATE_WHISPER	CiscoTerminal の getDeviceState() インターフェイスによって返されるこの定数値は、CiscoTerminal のアドレスに少なくとも 1 つの一方向メディアでのインターコム コールコールがあり、保留中やアラート状態になっているコールやアクティブなコールがないことを示します。
static final int	UNKNOWN_ENCODING	この端末の CiscoTerminal.getSupportedEncoding () が UNKNOWN であることを示します。
static final int	NOT_APPLICABLE	この CiscoMediaTerminal または RoutePoint の CiscoTerminal.getSupportedEncoding () が NOT_APPLICABLE であることを示します。
static final int	ASCII_ENCODING	この端末の CiscoTerminal.getSupportedEncoding () が ASCII であり、この端末が ASCII_ENCODING だけをサポートしていることを示します。
static final int	UCS2UNICODE_ENCODING	この端末の CiscoTerminal.getSupportedEncoding () が UCS2UNICODE_ENCODING であることを示します。
static final int	DND_OPTION_NONE	CiscoTerminal の getDNDOption() インターフェイスによって返されるこの定数値は、設定されている DND（サイレント）オプションが None であることを示します。
static final int	DND_OPTION_RINGER_OFF	CiscoTerminal の getDNDOption() インターフェイスによって返されるこの定数値は、設定されている DND（サイレント）オプションが Ringer Off であることを示します。電話機で DND が有効になっている場合、その電話機に着信コールがあっても、呼出音が鳴りません。
static final int	DND_OPTION_CALL_REJECT	CiscoTerminal の getDNDOptions() インターフェイスによって返されるこの定数値は、設定されている DND（サイレント）オプションが Call Reject であることを示します。電話機で DND が有効になっている場合 A 共用回線を除いて、その電話機へのすべてのコールが拒否されます。

表 6-181　CiscoTerminal のフィールド （続き）

インターフェイス	フィールド	説明
static final int	IP_ADDRESSING_MODE_ UNKNOWN	これは、IPAddressing モードが不明であることを示します。
static final int	IP_ADDRESSING_MODE_IPV4	これは、IPAddressing モードが IPv4 であることを示します。
static final int	IP_ADDRESSING_MODE_IPV6	これは、IPAddressing モードが IPv6 であることを示します。
static final int	IP_ADDRESSING_MODE_IPV4 _V6	これは、IPAddressing モードがデュアル スタック（IPv4 と IPv6 の両方）であることを示します。
static final int	IP_ADDRESSING_MODE_ UNKNOWN_ANATRED	これは Cisco Unified CM の ANAT のために予約された IPAddressing 定数です。JTAPI では使用されません。

メソッド

表 6-182　CiscoTerminal のメソッド

インターフェイス	メソッド	説明
int	getRegistrationState()	**推奨されません。** このメソッドは getState() メソッドに置き換えられました。この端末の状態を返します。状態を表す定数は次のとおりです。 • CiscoTerminal.OUT_OF_SERVICE • CiscoTerminal.IN_SERVICE
int	getState()	この端末の状態を返します。状態を表す定数は次のとおりです。 • CiscoTerminal.OUT_OF_SERVICE • CiscoTerminal.IN_SERVICE
CiscoRTPInputProperties	getRTPInputProperties()	この端末の ACTIVE TerminalConnection に関連付けられた、RTP 入力ストリームに使用されるプロパティ。CiscoTerminal が CiscoTerminal.REGISTERED 状態であり、そのプロバイダーが Provider.IN_SERVICE 状態である必要があります。さらに、Terminal.getTerminalConnections () が、TerminalConnection.ACTIVE 状態にある端末接続を少なくとも 1 つ返す必要があります。 **例外** javax.telephony.InvalidStateException

表 6-182　CiscoTerminal のメソッド （続き）

インターフェイス	メソッド	説明
CiscoRTPOutputProperties	getRTPOutputProperties()	この端末の ACTIVE TerminalConnection に関連付けられた、RTP 出力ストリームに使用されるプロパティ。CiscoTerminal が CiscoTerminal.REGISTERED 状態であり、そのプロバイダーが Provider.IN_SERVICE 状態である必要があります。さらに、Terminal.getTerminalConnections () が、TerminalConnection.ACTIVE 状態にある端末接続を少なくとも 1 つ返す必要があります。 **例外** javax.telephony.InvalidStateException
java.lang.String	sendData(java.lang.String terminalData)	**推奨されません。** CiscoTerminal.sendData (byte[]) を使用します。 **例外** javax.telephony.InvalidStateException javax.telephony.MethodNotSupportedException
byte[]	sendData(byte[] terminalData)	CiscoTerminal は CiscoTerminal.IN_SERVICE 状態になっている必要があり、プロバイダーは Provider.IN_SERVICE 状態になっている必要があります。アプリケーションでは、このインターフェイスを使用して、バイト形式の XSI オブジェクトをプッシュできます。電話機がこのデータを受信すると、このメソッドは正常に終了します。アプリケーションは、このインターフェイスを使用して最大 2000 バイトのデータを送信します。それ以上のサイズのデータ送信要求は拒否されます。 **例外** PlatformException (The data did not get sent successfully), javax.telephony.InvalidStateException, and javax.telephony.MethodNotSupportedException
CiscoTermEvFilter	getFilter()	この端末に関連付けられたフィルタ オブジェクトを取得します。

表 6-182　CiscoTerminal のメソッド （続き）

インターフェイス	メソッド	説明
void	setFilter(CiscoTermEvFilter terminalEvFilter)	TerminalObserver に配信されるイベントのフィルタを設定します。このメソッドはコードの実行中いつでも起動できますが、通常は初期化の一部として Terminal イベントのセットアップに使用するか、CiscTermCreatedEv によって新規端末の作成が通知されたときに使用します。 • 例 1：使用方法の 1 つとして、通常は配信されないボタン押下イベントをオンにします。Terminal term = provider.getTerminal (name); if (term instanceof CiscoTerm) { CiscoTerm ciscoTerm = (CiscoTerm)term; CiscoTermEvFilter filter = ciscoTerm.getFilter (); filter.setButtonPressedEnabled (true); } term.addObserver (terminalObserver) • 例 2：別の使用方法として、アプリケーションに関係のないいくつかのイベントをオフにします。たとえば、純粋にコール制御だけを行っているアプリケーションでは、次のようにメディア（RTP）イベントをオフにできます。Terminal term = provider.getTerminal (name); if (term instanceof CiscoTerm) { CiscoTerm ciscoTerm = (CiscoTerm)term; CiscoTermEvFilter filter = ciscoTerm.getFilter (); filter.setRTPEventsEnabled (false); ciscoTerm.setFilter (filter); } term.addObserver (terminalObserver); term.getAddresses () [0].addCallObserver (callObserver) **（注）** フィルタを明示的に設定せずに CallObserver を追加すると、RTP イベントがオンになります。Cisco JTAPI Release 1.4 以前のこの動作は、現在も維持されています。setFilter が明示的に呼び出されると、フィルタの設定が有効になります。RTP イベントは、上記のようなコードでは配信されませんが、次の例では配信されます。Terminal term = provider.getTerminal (name); term.addObserver (terminalObserver); term.getAddresses () [0].addCallObserver (callObserver).

表 6-182　CiscoTerminal のメソッド （続き）

インターフェイス	メソッド	説明
javax.telephony. TerminalConnection	unPark(javax.telephony. Address UnParkAddress, java.lang.String ParkedAt)	terminalConnection を返します。CiscoTerminal は CiscoTerminal.IN_SERVICE 状態になっている必要があり、プロバイダーは Provider.IN_SERVICE 状態になっている必要があります。このメソッドには、アドレスと文字列を入力できます。 **パラメータ** • UnParkAddress：端末上のいずれかのアドレス。 • ParkedAt：指定される文字列は、コールが以前にパークされたシステム パーク ポートです。コールがパークされると、この文字列が返されます。 **例外** • javax.telephony.InvalidStateException（CiscoTerminal.getState() が IN_SERVICE ではありません） • PlatformException：その他のエラーはパークが解除されたときに発生します（たとえば、パーク解除番号が無効）。 • javax.telephony.InvalidArgumentException • javax.telephony.ResourceUnavailableException
int	getDeviceState()	• この端末の DeviceState を返します。DeviceState には端末のすべてのアドレスのコール状態が累積して反映されています。状態を表す定数は次のとおりです。 • CiscoTerminal.DEVICESTATE_ILDE • CiscoTerminal.DEVICESTATE_ACTIVE • CiscoTerminal.DEVICESTATE_ALERTING • CiscoTerminal.DEVICESTATE_HELD • CiscoTerminal.DEVICESTATE_UNKNOWN • CiscoTerminal.DEVICESTATE_WHISPER **例外** • javax.telephony.InvalidStateException : CiscoTerminal.getState() が IN_SERVICE ではありません。

表 6-182　CiscoTerminal のメソッド （続き）

インターフェイス	メソッド	説明
int	getSupportedEncoding ()	この端末でサポートされる符号化方式の種類を返します。このメソッドを使用して、端末が Unicode をサポートしているかどうかを確認します。この情報にアクセスするには、端末が CiscoTerminal.IN_SERVICE 状態になっている必要があります。supportedEncoding は次の定数のいずれかです。 • CiscoTerminal.UNKNOWN_ENCODING • CiscoTerminal.ASCII_ENCODING • CiscoTerminal.UCS2UNICODE_ENCODING • CiscoTerminal.NOT_APPLICABLE **例外** • javax.telephony.InvalidStateException
int	getLocale()	この端末がサポートしているロケールを返します。このメソッドにアクセスするには、端末が CiscoTerminal.IN_SERVICE 状態になっている必要があります。 **例外** • javax.telephony.InvalidStateException : CiscoTerminal. getState() が IN_SERVICE ではありません。
boolean	isRestricted()	この端末の制限状態を返します。端末が制限されている場合、その端末に関連付けられたすべてのアドレスも制限されます。戻り値：端末が制限されている場合は true を返します。そうでない場合は false を返します。
void	createSnapshot ()	端末における現在のアクティブ コールのセキュリティ状態を格納した CiscoTermSnapshotEv イベントを生成します。このメソッドにアクセスするには、端末が CiscoTerminal.IN_SERVICE 状態になっており、CiscoTermEvFilter.setSnapshotEnabled() が True に設定されている必要があります。 **例外** • javax.telephony.InvalidStateException : CiscoTerminal.getState() が IN_SERVICE ではありません。

表 6-182　CiscoTerminal のメソッド （続き）

インターフェイス	メソッド	説明
java.lang.String	getAltScript()	この端末がサポートしているロケール代替スクリプトを返します。空の値が戻り値は、この端末がサポートしていないか、または代替スクリプトで設定されていないことを示します。このメソッドにアクセスするには、端末が CiscoTerminal.IN_SERVICE 状態になっている必要があります。 **例外** • javax.telephony.InvalidStateException : CiscoTerminal.getState() が IN_SERVICE ではありません。
int	getProtocol()	端末のプロトコル（SCCP、SIP、またはなし）を報告し、次の定数のいずれかとしてこの端末のプロトコルを返します。 • CiscoTerminalProtocol.PROTOCOL_NONE • CiscoTerminalProtocol.PROTOCOL_SCCP • CiscoTerminalProtocol.PROTOCOL_SIP
void	setDNDStatus(boolean dndStatus)	DND（サイレント）ステータスを設定します。これにより、DND 機能が有効または無効になります。この機能は、ルート ポイントには適用されません。 **パラメータ** • dndStatus **例外** • javax.telephony.InvalidStateException : CiscoTerminal.getState() が IN_SERVICE ではありません。
boolean	getDNDStatus()	DND（サイレント）ステータスを報告し、dndStatus を返します。 **例外** • javax.telephony.InvalidStateException : CiscoTerminal.getState() が IN_SERVICE ではありません。

表 6-182　CiscoTerminal のメソッド （続き）

インターフェイス	メソッド	説明
int	getDNDOption()	DND（サイレント）オプションの値を返します。DND 機能は実際の電話機だけに適用されるため、この値は CiscoMediaTerminal または CiscoRouteTerminal にとっては重要ではありません。DND オプションを表す定数は次のとおりです。 • CiscoTerminal.DND_OPTION_NONE • CiscoTerminal.DND_OPTION_RINGER_OFF • CiscoTerminal.DND_OPTION_CALL_REJECT **例外** • javax.telephony.InvalidStateException : CiscoTerminal.getState() が IN_SERVICE ではありません。
java.lang.String	getEMLoginUsername	エクステンション モビリティ（EM）ログイン ユーザ名 EM のユーザが端末にログインしていない場合、このインターフェイスは null または空文字列を返します。 **例外** • javax.telephony.InvalidStateException : CiscoCiscoTerminal.getState() が IN_SERVICE ではない場合。

継承したフィールド

インターフェイス javax.telephony.Terminal から

addCallObserver, addObserver, getAddresses, getCallObservers, getCapabilities, getName, getObservers, getProvider, getTerminalCapabilities, getTerminalConnections, removeCallObserver, removeObserver

インターフェイス com.cisco.jtapi.extensions.CiscoObjectContainer から

関連資料

Terminal と CiscoMediaTerminal、「定数フィールド値」（P.F-1）および CiscoTermEvFilter を参照してください。

CiscoTerminalConnection

CiscoTerminalConnection インターフェイスは、CallControlTerminalConnection インターフェイスを拡張し、機能を追加します。アプリケーションは getReason メソッドを使用して、接続が確立された原因を取得できます。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

javax.telephony.callcontrol.CallControlTerminalConnection, CiscoObjectContainer, javax.telephony.TerminalConnection

宣言

public interface CiscoTerminalConnection extends javax.telephony.callcontrol.CallControlTerminalConnection, CiscoObjectContainer

フィールド

表 6-183　CiscoTerminalConnection のフィールド

インターフェイス	フィールド	説明
static final int	CISCO_SELECTEDNONE	コールが選択されていません。
static final int	CISCO_SELECTEDLOCAL	コールが選択されています。
static final int	CISCO_SELECTEDREMOTE	コールがその共用回線で選択されている場合、Passive TerminalConnection がこの選択状態を受け取ります。

継承したフィールド

インターフェイス javax.telephony.callcontrol.CallControlTerminalConnection から
BRIDGED, DROPPED, HELD, IDLE, INUSE, RINGING, TALKING, UNKNOWN

インターフェイス javax.telephony.TerminalConnection から
ACTIVE, PASSIVE

メソッド

表 6-184　CiscoTerminalConnection のメソッド

インターフェイス	メソッド	説明
boolean	getPrivacyStatus()	端末上のコールのプライバシー ステータスを返します。このインターフェイスは、プライバシーが有効になっている場合は true、そうでない場合は false を返します。アプリケーションの端末の実装でコールに関する情報を表示する前に、常に TerminalConnection のプライバシー ステータスを確認する必要があります。
int	getSelectStatus()	端末上のコールの選択ステータスを返します。コールのコール処理オペレーションを実行する前に、常に TerminalConnection の選択ステータスを確認する必要があります。次のいずれかになります。 • CiscoTerminalConnection.CISCO_SELECTED NONE • CiscoTerminalConnection.CISCO_SELECTED LOCAL • CiscoTerminalConnection.CISCO_SELECTED REMOTE

表 6-184　CiscoTerminalConnection のメソッド （続き）

インターフェイス	メソッド	説明
void	startRecording(int playTone Direction)	コールの録音を開始します。このメソッドが成功した場合は、システムは CiscoTermConnRecordingStartEv および CiscoTermConnRecordingTargetInfoEv をコール オブザーバに配信します。 **事前条件** • ((this.getTerminal()).getProvider()).getState() == Provider.IN_SERVICE • this.getCallControlState() == CallControlTerminalConnection.TALKING • ((CiscoProviderCapabilities)(this.getTerminal(). getProvider().getProviderCapabilities()).canRecord() == TRUE • this.getConnection().getAddress().getRecordingConfig (this.getTerminal()) == CiscoAddress.APPLICATION_CONTROLLED__ RECORDING **パラメータ** • playToneDirection：トーンを再生するかどうかを指定します。有効な値は次のとおりです。 – CiscoCall.PLAYTONE_NOLOCAL_OR_REMOTE – CiscoCall.PLAYTONE_LOCALONLY – CiscoCall.PLAYTONE_REMOTEONLY – CiscoCall.PLAYTONE_ BOTHLOCALANDREMOTE **例外** • javax.telephony.InvalidStateException：プロバイダーが「インサービス」状態でないか、または TerminalConnection が「TALKING」状態ではありません。 • javax.telephony.PrivilegeViolationException：アプリケーションに、このメソッドを起動する適切な権限がありません。 • javax.telephony.ResourceUnavailableException：このメソッドに必要な内部リソースがないことを意味します。 • javax.telephony.InvalidArgumentException：playToneDirection の値が有効ではありません。

表 6-184　CiscoTerminalConnection のメソッド （続き）

インターフェイス	メソッド	説明
CiscoRecorderInfo	getCiscoRecorderInfo()	録音側の端末名およびアドレスを公開する CiscoRecorderInfo を返します。コールが録音されていない場合は、null を返します。コール制御端末接続は、通話中状態である必要があります。
CiscoMonitorInitiatorInfo	getCiscoMonitorInitiatorInfo()	CiscoMonitorInitiatorInfo を返します。コールがモニタリングされていない場合は、null を返します。アプリケーションはこのメソッドをモニタリング ターゲットの端末接続に対して使用して、モニタリングの開始側に関する情報を取得するか、またはモニタリング セッションが存在しないと判断できます。
CiscoMonitorTargetInfo	getCiscoMonitorTargetInfo()	CiscoMonitorTargetInfo または null を返します。アプリケーションはモニタリングの開始側の端末接続に対してこのメソッドを使用して、モニタリング ターゲットに関する情報を取得できます。このメソッドは、モニタリング ターゲットの端末接続に対して呼び出された場合や、モニタリング セッションが存在しない場合には、null を返します。

継承したメソッド

インターフェイス javax.telephony.callcontrol.CallControlTerminalConnection から
getCallControlState, hold, join, leave, unhold

インターフェイス javax.telephony.TerminalConnection から
answer, getCapabilities, getConnection, getState, getTerminal, getTerminalConnectionCapabilities, getObject, setObject

インターフェイス com.cisco.jtapi.extensions.CiscoObjectContainer から

関連資料

「定数フィールド値」（P.F-1）を参照してください。

CiscoTerminalObserver

アプリケーションは、Terminal.addObserver メソッドを使用して Terminal を監視する際に、このインターフェイスを実装して CiscoRTPInputStartedEv や CiscoRTPInputStoppedEv などの CiscoTermEv イベントを受信します。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

javax.telephony.TerminalObserver

宣言

public interface CiscoTerminalObserver extends javax.telephony.TerminalObserver

フィールド

なし

メソッド

なし

継承したメソッド

インターフェイス javax.telephony.TerminalObserver から
terminalChangedEvent

関連資料

CiscoTermInServiceEv、CiscoTermOutOfServiceEv、CiscoRTPInputStartedEv、CiscoRTPInputStoppedEv、CiscoRTPOutputStartedEv、および CiscoRTPOutputStoppedEv を参照してください。

CiscoTerminalProtocol

CiscoTerminalProtocol イベントは、プロトコルのタイプを定義する定数のためのコンテナです。

インターフェイス履歴

Cisco Unified Communications Manager リリース	説明
3.x	拡張が追加されました。

スーパーインターフェイス

public interface CiscoTerminalProtocol

フィールド

表 6-185　CiscoTerminalProtocol のフィールド

インターフェイス	フィールド	説明
static int	PROTOCOL_NONE	CiscoTerminal の getProtocol() インターフェイスによって返されるこの定数値は、CiscoTerminal のプロトコルのタイプが不明であるか、または利用できないことを示します。
static int	PROTOCOL_SCCP	CiscoTerminal の getProtocol() インターフェイスによって返されるこの定数値は、CiscoTerminal のプロトコルのタイプが SCCP であることを示します。
static int	PROTOCOL_SIP	CiscoTerminal の getProtocol() インターフェイスによって返されるこの定数値は、CiscoTerminal のプロトコルのタイプが SIP であることを示します。

関連資料

詳細については、CiscoTerminal と「定数フィールド値」（P.F-1）を参照してください。

CiscoTermInServiceEv

CiscoTermInServiceEv イベントは、CiscoTerminal が動作可能であることを伝えるために、アプリケーションの TerminalObservers に送信されます。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

CiscoEv, CiscoTermEv, javax.telephony.events.Ev, javax.telephony.events.TermEv

宣言

public interface CiscoTermInServiceEv extends CiscoTermEv

フィールド

表 6-186　CiscoTermInServiceEv のフィールド

インターフェイス	フィールド
static int	ID

継承したフィールド

インターフェイス javax.telephony.events.Ev から

CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から

CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

表 6-187　CiscoTermInServiceEv のメソッド

インターフェイス	メソッド	説明
int	getSupportedEncoding ()	サポートされる符号化方式を返すことによって、端末が UNICODE に対応しているかどうかを報告します。サポートされる符号化方式の値は、次の定数のいずれかです。 • CiscoTerminal.UNKNOWN_ENCODING • CiscoTerminal.ASCII_ENCODING • CiscoTerminal.UCS2UNICODE_ENCODING • CiscoTerminal.NOT_APPLICABLE 戻り値：この端末でサポートされる符号化方式の整数値。
int	getLocale()	この端末がサポートしているロケールを返します。CiscoLocales インターフェイスで定義された int 値を返します。
boolean	getDNDStatus()	アプリケーションに現在の DND（サイレント）ステータスを返します。ブール値の dndStatus を返します。
int	getDNDOption()	アプリケーションに現在の DND（サイレント）オプションを返します。DND オプションを表す定数は次のとおりです。 • CiscoTerminal.DND_OPTION_NONE • CiscoTerminal.DND_OPTION_RINGER_OFF • CiscoTerminal.DND_OPTION_CALL_REJECT int dndOption を返します。

継承したメソッド

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.TermEv から
getTerminal

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

「定数フィールド値」（P.F-1）および CiscoLocales を参照してください。

CiscoTermOutOfServiceEv

CiscoTermOutOfServiceEv イベントは、CiscoTerminal がアウトオブサービスであることを伝えるために、アプリケーションの TerminalObservers に送信されます。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

CiscoEv, CiscoOutOfServiceEv, CiscoTermEv, javax.telephony.events.Ev, javax.telephony.events.TermEv

宣言

public interface CiscoTermOutOfServiceEv extends CiscoTermEv, CiscoOutOfServiceEv

フィールド

表 6-188　CiscoTermOutOfServiceEv のフィールド

インターフェイス	フィールド
static int	ID

継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE,

CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN, CAUSE_CALLMANAGER_FAILURE, CAUSE_CTIMANAGER_FAILURE, CAUSE_DEVICE_FAILURE, CAUSE_DEVICE_RESTRICTED, CAUSE_DEVICE_UNREGISTERED, CAUSE_LINE_RESTRICTED, CAUSE_NOCALLMANAGER_AVAILABLE, CAUSE_REHOME_TO_HIGHER_PRIORITY_CM, CAUSE_REHOMING_FAILURE

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス com.cisco.jtapi.extensions.CiscoOutOfServiceEv から

メソッド

なし

継承したメソッド

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.TermEv から
getTerminal

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

「定数フィールド値」（P.F-1）を参照してください。

CiscoTermRegistrationFailedEv

プロバイダーで TerminalRegistration が失敗すると、アプリケーションはこのイベントを受信します。getErrorCode() が返すエラーは、問題について説明しています。このイベントを受け取った場合、アプリケーションは端末の再登録を試みる必要があります。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

CiscoEv, CiscoTermEv, javax.telephony.events.Ev, javax.telephony.events.TermEv

宣言

public interface CiscoTermRegistrationFailedEv extends CiscoTermEv

フィールド

表 6-189　CiscoTermRegistrationFailedEv のフィールド

インターフェイス	フィールド	説明
static final int	ID	なし
static final int	MEDIA_CAPABILITY_MISMATCH	端末が別のメディア機能ですでに登録されているため、登録に失敗しました。同じ機能で再登録してください。
static final int	MEDIA_ALREADY_TERMINATED_NONE	端末がメディア終端タイプ none ですでに登録されているため、登録に失敗しました。メディア終端タイプ none で再登録してください。
static final int	MEDIA_ALREADY_TERMINATED_STATIC	端末が静的メディア終端ですでに登録されているため、登録に失敗しました。静的登録では、2 回目の登録ができません。端末が登録解除されるまで待機してください。
static final int	MEDIA_ALREADY_TERMINATED_DYNAMIC	端末が動的メディア終端ですでに登録されているため、登録に失敗しました。動的メディア終端で再登録してください。
static final int	OWNER_NOT_ALIVE	端末の登録中に、登録は競合状態でした。端末を登録してください。

表 6-189　CiscoTermRegistrationFailedEv のフィールド （続き）

インターフェイス	フィールド	説明
static final int	DB_INITIALIZATION_ERROR	端末の登録中にアドレスの初期化エラーが発生しました。端末を登録してください。
static final int	UNKNOWN	不明な内部理由のために登録に失敗しました。端末を再登録してください。
static final int	IP_ADDRESSING_MODE_MISMATCH	サポートされていない IP アドレッシング モードのために登録に失敗しました。正しい IP アドレッシング モードで端末の登録を試行してください。

継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

表 6-190　CiscoTermRegistrationFailedEv のメソッド

インターフェイス	メソッド	説明
int	getErrorCode()	この例外のエラー コードを整数で返します。

継承したメソッド

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.TermEv から
getTerminal

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

「定数フィールド値」（P.F-1）を参照してください。

CiscoTermRemovedEv

CiscoTerminal がプロバイダー ドメインから削除されると、CiscoTermRemovedEv イベントがアプリケーションのプロバイダー オブザーバに送信されます。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

CiscoEv, CiscoProvEv, javax.telephony.events.Ev, javax.telephony.events.ProvEv

宣言

public interface CiscoTermRemovedEv extends CiscoProvEv

フィールド

表 6-191　CiscoTermRemovedEv のフィールド

インターフェイス	フィールド
static final int	ID

継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

表 6-192　CiscoTermRemovedEv のメソッド

インターフェイス	メソッド	説明
javax.telephony.Terminal	getTerminal()	プロバイダー ドメインから削除された端末を返します。

継承したメソッド

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.ProvEv から
getProvider

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

「定数フィールド値」（P.F-1）を参照してください。

CiscoTermRestrictedEv

アプリケーションの実行開始後に、ユーザが Cisco Unified Communications Manager Administration から端末を制限すると、アプリケーションは CiscoTermRestrictedEv イベントを受信します。アプリケーションは、制限された端末やその端末のアドレスに関するイベントは受信できません。インサービス状態にある端末が制限された場合、アプリケーションはこのイベントを受信し、端末およびそれに対応するアドレスはアウトオブサービス状態になります。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

CiscoEv, CiscoProvEv, javax.telephony.events.Ev, javax.telephony.events.ProvEv

宣言

public interface CiscoTermRestrictedEv extends CiscoProvEv

フィールド

表 6-193　CiscoTermRestrictedEv のフィールド

インターフェイス	フィールド
static int	ID

継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

表 6-194　CiscoTermRestrictedEv のメソッド

インターフェイス	メソッド	説明
javax.telephony.Terminal	getTerminal	制限された端末を返します。

継承したメソッド

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.ProvEv から
getProvider

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

「定数フィールド値」（P.F-1）を参照してください。

CiscoTermSnapshotCompletedEv

2 つのエンド ポイント間でコールが確立された後にアプリケーションが起動された場合、コール中のモニタリングに関して、アプリケーションは Terminal.createSnapshot() を照会する必要があります。端末上のすべてのアドレスのコール イベントが配信された後で、アプリケーションは CiscoTermSnapshotCompletedEv を取得します。下位互換性を維持するため、これらのイベントはアプリケーションで CiscoTermEvFilter の snapShotRTPEnabled フィルタが有効にされた場合にのみ生成されます。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

CiscoEv, CiscoTermEv, javax.telephony.events.Ev, javax.telephony.events.TermEv

宣言

public interface CiscoTermSnapshotCompletedEv extends CiscoTermEv

フィールド

表 6-195　CiscoTermSnapshotCompletedEv のフィールド

インターフェイス	フィールド
static int	ID

継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

なし

継承したメソッド

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.TermEv から
getTerminal

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

CiscoTermEvFilter と 「定数フィールド値」（P.F-1）を参照してください。

CiscoTermSnapshotEv

2 つのエンド ポイント間でコールが確立された後にアプリケーションが起動された場合、コール中のモニタリングに関して、アプリケーションは Terminal.createSnapshot() を照会する必要があります。スナップショット イベント CiscoTermSnapshotEv が送信され、エンドポイント間の現在のメディアがセキュアかどうかが示されます。また、アプリケーションは CiscoMediaCallSecurityIndicator を照会してコールのセキュリティ インジケータを取得することもできますが、このイベントに鍵情報が含まれません。端末のどの回線にもコールが存在しない場合、アプリケーションは CiscoTermSnapshotCompletedEv のみを取得します。下位互換性を維持するため、これらのイベントはアプリケーションで CiscoTermEvFilter の snapShotRTPEnabled フィルタが有効にされた場合にのみ生成されます。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

CiscoEv, CiscoTermEv, javax.telephony.events.Ev, javax.telephony.events.TermEv

宣言

public interface CiscoTermSnapshotEv extends CiscoTermEv

フィールド

表 6-196　CiscoTermSnapshotEv のフィールド

インターフェイス	フィールド
static int	ID

継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

表 6-197　CiscoTermSnapshotEv のメソッド

インターフェイス	メソッド	説明
CiscoMediaCallSecurity Indicator[]	getCiscoMediaCallSecurityIndicator()	このデバイス上の各アクティブ コールにおけるメディアのセキュリティ状態を返します。

継承したメソッド

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.TermEv から
getTerminal

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

CiscoTermEvFilter と 「定数フィールド値」（P.F-1）を参照してください。

CiscoTone

CiscoTone インターフェイスは CTI Tone 定数コードを定義します。CiscoToneChangedEv は、これらの定数のいずれかを返す getTone() メソッドを提供します。ZIPZIP トーン タイプがアプリケーションに公開されます。

履歴

Cisco Unified Communications Manager リリース	説明
7.0(1)	拡張が追加されました。

スーパーインターフェイス

public interface CiscoTone

フィールド

表 6-198　CiscoTone のフィールド

インターフェイス	フィールド	説明
Static int	ZIPZIP	このインターフェイスは ZIPZIP tone の整数値を定義します。CiscoToneChangedEv.getTone() インターフェイスはトーンの整数値を返します。

CiscoToneChangedEv、「定数フィールド値」（P.F-1）も参照してください。

CiscoToneChangedEv

CiscoToneChangedEv イベントは、このコールにトーンが生成されたことを示します。このイベントは CallControlCallObserver インターフェイスによって報告されます。現在、このトーンは Forced Authorization Code （FAC）機能または Client Matter Code（CMC）機能だけによって生成されます。FAC_CMC によってトーンが作成される場合、CiscoToneChangedEv.getCiscoCause() は CiscoCallEv.CAUSE_FAC_CMC_FEATURE を返します。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

javax.telephony.events.CallEv, CiscoCallEv, CiscoEv, javax.telephony.events.Ev

宣言

public interface CiscoToneChangedEv extends CiscoCallEv

フィールド

表 6-199　CiscoToneChangedEv のフィールド

インターフェイス	フィールド	説明
static final int	ID	なし
static final int	FAC_REQUIRED	Connection.addToAddress(String) を使用して FAC を入力する必要があることを示しています。これは FAC_CMC_FEATURE_ID だけに適用されます。
static final int	CMC_REQUIRED	Connection.addToAddress(String) を使用して CMC を入力する必要があることを示しています。これは FAC_CMC_FEATURE_ID だけに適用されます。
static final int	FAC_CMC_REQUIRED	Connection.addToAddress(String) を使用して FAC と CMC の両方を入力する必要があることを示しています。アプリケーションは、一度に 1 つずつ文字列コードを入力することも、同時に両方を入力して #（シャープ記号）文字で区切ることもできます。これは FAC_CMC_FEATURE_ID だけに適用されます。

CAUSE_ACCESSINFORMATIONDISCARDED

, CAUSE_BARGE, CAUSE_BCBPRESENTLYAVAIL, CAUSE_BCNAUTHORIZED, CAUSE_BEARERCAPNIMPL, CAUSE_CALLBEINGDELIVERED, CAUSE_CALLIDINUSE, CAUSE_CALLMANAGER_FAILURE, CAUSE_CALLREJECTED, CAUSE_CALLSPLIT, CAUSE_CHANTYPENIMPL, CAUSE_CHANUNACCEPTABLE, CAUSE_CTICCMSIP400BADREQUEST, CAUSE_CTICCMSIP401UNAUTHORIZED, CAUSE_CTICCMSIP402PAYMENTREQUIRED, CAUSE_CTICCMSIP403FORBIDDEN, CAUSE_CTICCMSIP404NOTFOUND, CAUSE_CTICCMSIP405METHODNOTALLOWED, CAUSE_CTICCMSIP406NOTACCEPTABLE, CAUSE_CTICCMSIP407PROXYAUTHENTICATIONREQUIRED, CAUSE_CTICCMSIP408REQUESTTIMEOUT, CAUSE_CTICCMSIP410GONE, CAUSE_CTICCMSIP411LENGTHREQUIRED, CAUSE_CTICCMSIP413REQUESTENTITYTOOLONG, CAUSE_CTICCMSIP414REQUESTURITOOLONG,

CAUSE_CTICCMSIP415UNSUPPORTEDMEDIATYPE,
CAUSE_CTICCMSIP416UNSUPPORTEDURISCHEME,
CAUSE_CTICCMSIP420BADEXTENSION, CAUSE_CTICCMSIP421EXTENSTIONREQUIRED,
CAUSE_CTICCMSIP423INTERVALTOOBRIEF,
CAUSE_CTICCMSIP480TEMPORARILYUNAVAILABLE,
CAUSE_CTICCMSIP481CALLLEGDOESNOTEXIST, CAUSE_CTICCMSIP482LOOPDETECTED,
CAUSE_CTICCMSIP483TOOMANYHOOPS, CAUSE_CTICCMSIP484ADDRESSINCOMPLETE,
CAUSE_CTICCMSIP485AMBIGUOUS, CAUSE_CTICCMSIP486BUSYHERE,
CAUSE_CTICCMSIP487REQUESTTERMINATED,
CAUSE_CTICCMSIP488NOTACCEPTABLEHERE, CAUSE_CTICCMSIP491REQUESTPENDING,
CAUSE_CTICCMSIP493UNDECIPHERABLE,
CAUSE_CTICCMSIP500SERVERINTERNALERROR,
CAUSE_CTICCMSIP501NOTIMPLEMENTED, CAUSE_CTICCMSIP502BADGATEWAY,
CAUSE_CTICCMSIP503SERVICEUNAVAILABLE, CAUSE_CTICCMSIP504SERVERTIMEOUT,
CAUSE_CTICCMSIP505SIPVERSIONNOTSUPPORTED,
CAUSE_CTICCMSIP513MESSAGETOOLARGE, CAUSE_CTICCMSIP600BUSYEVERYWHERE,
CAUSE_CTICCMSIP603DECLINE, CAUSE_CTICCMSIP604DOESNOTEXISTANYWHERE,
CAUSE_CTICCMSIP606NOTACCEPTABLE, CAUSE_CTICONFERENCEFULL,
CAUSE_CTIDEVICENOTPREEMPTABLE, CAUSE_CTIDROPCONFEREE,
CAUSE_CTIMANAGER_FAILURE, CAUSE_CTIPRECEDENCECALLBLOCKED,
CAUSE_CTIPRECEDENCELEVELEXCEEDED,
CAUSE_CTIPRECEDENCEOUTOFBANDWIDTH, CAUSE_CTIPREEMPTFORREUSE,
CAUSE_CTIPREEMPTNOREUSE, CAUSE_DESTINATIONOUTOFORDER,
CAUSE_DESTNUMMISSANDDCNOTSUB, CAUSE_DPARK, CAUSE_DPARK_REMINDER,
CAUSE_DPARK_UNPARK, CAUSE_EXCHANGEROUTINGERROR, CAUSE_FAC_CMC,
CAUSE_FACILITYREJECTED, CAUSE_IDENTIFIEDCHANDOESNOTEXIST, CAUSE_IENIMPL,
CAUSE_INBOUNDBLINDTRANSFER, CAUSE_INBOUNDCONFERENCE,
CAUSE_INBOUNDTRANSFER, CAUSE_INCOMINGCALLBARRED,
CAUSE_INCOMPATABLEDDESTINATION, CAUSE_INTERWORKINGUNSPECIFIED,
CAUSE_INVALIDCALLREFVALUE, CAUSE_INVALIDIECONTENTS,
CAUSE_INVALIDMESSAGEUNSPECIFIED, CAUSE_INVALIDNUMBERFORMAT,
CAUSE_INVALIDTRANSITNETSEL, CAUSE_MANDATORYIEMISSING,
CAUSE_MSGNCOMPATABLEWCS, CAUSE_MSGTYPENCOMPATWCS,
CAUSE_MSGTYPENIMPL, CAUSE_NETOUTOFORDER, CAUSE_NOANSWERFROMUSER,
CAUSE_NOCALLSUSPENDED, CAUSE_NOCIRCAVAIL, CAUSE_NOERROR,
CAUSE_NONSELECTEDUSERCLEARING, CAUSE_NORMALCALLCLEARING,
CAUSE_NORMALUNSPECIFIED, CAUSE_NOROUTETODDESTINATION,
CAUSE_NOROUTETOTRANSITNET, CAUSE_NOUSERRESPONDING,
CAUSE_NUMBERCHANGED, CAUSE_ONLYRDIVEARERCAPAVAIL,
CAUSE_OUTBOUNDCONFERENCE, CAUSE_OUTBOUNDTRANSFER,
CAUSE_OUTOFBANDWIDTH, CAUSE_PROTOCOLERRORUNSPECIFIED, CAUSE_QSIG_PR,
CAUSE_QUALOFSERVNAVAIL, CAUSE_QUIET_CLEAR,
CAUSE_RECOVERYONTIMEREXPIRY, CAUSE_REDIRECTED,
CAUSE_REQCALLIDHASBEENCLEARED, CAUSE_REQCIRCNAVIL,
CAUSE_REQFACILITYNIMPL, CAUSE_REQFACILITYNOTSUBSCRIBED,
CAUSE_RESOURCESNAVAIL, CAUSE_RESPONSETOSTATUSENQUIRY,
CAUSE_SERVNOTAVAILUNSPECIFIED, CAUSE_SERVOPERATIONVIOLATED,
CAUSE_SERVOROPTNAVAILORIMPL, CAUSE_SUBSCRIBERABSENT,
CAUSE_SUSPCALLBUTNOTTHISONE, CAUSE_SWITCHINGEQUIPMENTCONGESTION,
CAUSE_TEMPORARYFAILURE, CAUSE_UNALLOCATEDNUMBER, CAUSE_USERBUSY

継承したフィールド

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から
CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス com.cisco.jtapi.extensions.CiscoCallEv から

メソッド

表 6-200　CiscoToneChangedEv のメソッド

インターフェイス	メソッド	説明
int	getTone()	CiscoTone から生成されたトーン タイプを返します。
int	getWhichCodeRequired()	ダイヤルされた DN にどのコードが必要かを返します。odeRequired は次のいずれかです。 • CiscoToneChangedEv.FAC_REQUIRED • CiscoToneChangedEv.CMC_REQUIRED • CiscoToneChangedEv.FAC_CMC_REQUIRED

継承したメソッド

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.CallEv から
getCall

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス com.cisco.jtapi.extensions.CiscoCallEv から

関連資料

「定数フィールド値」（P.F-1）を参照してください。

CiscoTransferEndEv

CiscoTransferEndEv イベントは、転送動作が完了したことを示します。このイベントは CallControlCallObserver インターフェイスによって報告されます。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

javax.telephony.events.CallEv, CiscoCallEv, CiscoEv, javax.telephony.events.Ev

宣言

public interface CiscoTransferEndEv extends CiscoCallEv

フィールド

なし

継承したフィールド

インターフェイス com.cisco.jtapi.extensions.CiscoCallEv から
CAUSE_ACCESSINFORMATIONDISCARDED, CAUSE_BARGE, CAUSE_BCBPRESENTLYAVAIL, CAUSE_BCNAUTHORIZED, CAUSE_BEARERCAPNIMPL, CAUSE_CALLBEINGDELIVERED, CAUSE_CALLIDINUSE, CAUSE_CALLMANAGER_FAILURE, CAUSE_CALLREJECTED, CAUSE_CALLSPLIT, CAUSE_CHANTYPENIMPL, CAUSE_CHANUNACCEPTABLE, CAUSE_CTICCMSIP400BADREQUEST, CAUSE_CTICCMSIP401UNAUTHORIZED, CAUSE_CTICCMSIP402PAYMENTREQUIRED, CAUSE_CTICCMSIP403FORBIDDEN, CAUSE_CTICCMSIP404NOTFOUND, CAUSE_CTICCMSIP405METHODNOTALLOWED, CAUSE_CTICCMSIP406NOTACCEPTABLE, CAUSE_CTICCMSIP407PROXYAUTHENTICATIONREQUIRED,

CAUSE_CTICCMSIP408REQUESTTIMEOUT, CAUSE_CTICCMSIP410GONE,
CAUSE_CTICCMSIP411LENGTHREQUIRED,
CAUSE_CTICCMSIP413REQUESTENTITYTOOLONG,
CAUSE_CTICCMSIP414REQUESTURITOOLONG,
CAUSE_CTICCMSIP415UNSUPPORTEDMEDIATYPE,
CAUSE_CTICCMSIP416UNSUPPORTEDURISCHEME,
CAUSE_CTICCMSIP420BADEXTENSION, CAUSE_CTICCMSIP421EXTENSTIONREQUIRED,
CAUSE_CTICCMSIP423INTERVALTOOBRIEF,
CAUSE_CTICCMSIP480TEMPORARILYUNAVAILABLE,
CAUSE_CTICCMSIP481CALLLEGDOESNOTEXIST, CAUSE_CTICCMSIP482LOOPDETECTED,
CAUSE_CTICCMSIP483TOOMANYHOOPS, CAUSE_CTICCMSIP484ADDRESSINCOMPLETE,
CAUSE_CTICCMSIP485AMBIGUOUS, CAUSE_CTICCMSIP486BUSYHERE,
CAUSE_CTICCMSIP487REQUESTTERMINATED,
CAUSE_CTICCMSIP488NOTACCEPTABLEHERE, CAUSE_CTICCMSIP491REQUESTPENDING,
CAUSE_CTICCMSIP493UNDECIPHERABLE,
CAUSE_CTICCMSIP500SERVERINTERNALERROR,
CAUSE_CTICCMSIP501NOTIMPLEMENTED, CAUSE_CTICCMSIP502BADGATEWAY,
CAUSE_CTICCMSIP503SERVICEUNAVAILABLE, CAUSE_CTICCMSIP504SERVERTIMEOUT,
CAUSE_CTICCMSIP505SIPVERSIONNOTSUPPORTED,
CAUSE_CTICCMSIP513MESSAGETOOLARGE, CAUSE_CTICCMSIP600BUSYEVERYWHERE,
CAUSE_CTICCMSIP603DECLINE, CAUSE_CTICCMSIP604DOESNOTEXISTANYWHERE,
CAUSE_CTICCMSIP606NOTACCEPTABLE, CAUSE_CTICONFERENCEFULL,
CAUSE_CTIDEVICENOTPREEMPTABLE, CAUSE_CTIDROPCONFEREE,
CAUSE_CTIMANAGER_FAILURE, CAUSE_CTIPRECEDENCECALLBLOCKED,
CAUSE_CTIPRECEDENCELEVELEXCEEDED,
CAUSE_CTIPRECEDENCEOUTOFBANDWIDTH, CAUSE_CTIPREEMPTFORREUSE,
CAUSE_CTIPREEMPTNOREUSE, CAUSE_DESTINATIONOUTOFORDER,
CAUSE_DESTNUMMISSANDDCNOTSUB, CAUSE_DPARK, CAUSE_DPARK_REMINDER,
CAUSE_DPARK_UNPARK, CAUSE_EXCHANGEROUTINGERROR, CAUSE_FAC_CMC,
CAUSE_FACILITYREJECTED, CAUSE_IDENTIFIEDCHANDOESNOTEXIST, CAUSE_IENIMPL,
CAUSE_INBOUNDBLINDTRANSFER, CAUSE_INBOUNDCONFERENCE,
CAUSE_INBOUNDTRANSFER, CAUSE_INCOMINGCALLBARRED,
CAUSE_INCOMPATABLEDDESTINATION, CAUSE_INTERWORKINGUNSPECIFIED,
CAUSE_INVALIDCALLREFVALUE, CAUSE_INVALIDIECONTENTS,
CAUSE_INVALIDMESSAGEUNSPECIFIED, CAUSE_INVALIDNUMBERFORMAT,
CAUSE_INVALIDTRANSITNETSEL, CAUSE_MANDATORYIEMISSING,
CAUSE_MSGNCOMPATABLEWCS, CAUSE_MSGTYPENCOMPATWCS,
CAUSE_MSGTYPENIMPL, CAUSE_NETOUTOFORDER, CAUSE_NOANSWERFROMUSER,
CAUSE_NOCALLSUSPENDED, CAUSE_NOCIRCAVAIL, CAUSE_NOERROR,
CAUSE_NONSELECTEDUSERCLEARING, CAUSE_NORMALCALLCLEARING,
CAUSE_NORMALUNSPECIFIED, CAUSE_NOROUTETODDESTINATION,
CAUSE_NOROUTETOTRANSITNET, CAUSE_NOUSERRESPONDING,
CAUSE_NUMBERCHANGED, CAUSE_ONLYRDIVEARERCAPAVAIL,
CAUSE_OUTBOUNDCONFERENCE, CAUSE_OUTBOUNDTRANSFER,
CAUSE_OUTOFBANDWIDTH, CAUSE_PROTOCOLERRORUNSPECIFIED, CAUSE_QSIG_PR,
CAUSE_QUALOFSERVNAVAIL, CAUSE_QUIET_CLEAR,
CAUSE_RECOVERYONTIMEREXPIRY, CAUSE_REDIRECTED,
CAUSE_REQCALLIDHASBEENCLEARED, CAUSE_REQCIRCNAVIL,
CAUSE_REQFACILITYNIMPL, CAUSE_REQFACILITYNOTSUBSCRIBED,
CAUSE_RESOURCESNAVAIL, CAUSE_RESPONSETOSTATUSENQUIRY,
CAUSE_SERVNOTAVAILUNSPECIFIED, CAUSE_SERVOPERATIONVIOLATED,

CAUSE_SERVOROPTNAVAILORIMPL, CAUSE_SUBSCRIBERABSENT, CAUSE_SUSPCALLBUTNOTTHISONE, CAUSE_SWITCHINGEQUIPMENTCONGESTION, CAUSE_TEMPORARYFAILURE, CAUSE_UNALLOCATEDNUMBER, CAUSE_USERBUSY

インターフェイス javax.telephony.events.Ev から

CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から

CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

表 6-201　CiscoTransferEndEv のメソッド

インターフェイス	メソッド	説明
javax.telephony.Call	getTransferredCall()	転送を実行したコールを返します。このコールは Call.INVALID の状態です。
javax.telephony.Call	getFinalCall()	転送の終了後にアクティブになっているコールを返します。
javax.telephony.TerminalConnection	getTransferController()	現在、最後のコールの転送コントローラとして機能している ACTIVE TerminalConnection を返します。transferController が SharedLine の場合、複数の TerminalConnection オブジェクトがあります。このメソッドは ACTIVE TerminalConnection を返します。ただし、アプリケーションが ACTIVE TerminalConnection を監視していない場合、このメソッドは PASSIVE TerminalConnection オブジェクトのうち 1 つを返します。
javax.telephony.TerminalConnection[]	getTransferControllers()	現在、最後のコールの転送コントローラとして機能している TerminalConnection オブジェクトのリストを返します。transferController が SharedLine ではないときは、1 つの TerminalConnection だけがリストに含まれます。転送コントローラにオブザーバがない場合、このメソッドは null を返します。

表 6-201　CiscoTransferEndEv のメソッド

インターフェイス	メソッド	説明
javax.telephony.Address	getTransferControllerAddress()	現在、最後のコールの転送コントローラとして機能しているアドレスを返します。
boolean	isSuccess()	転送が正常に実行された場合は true、そうでない場合は false を返します。

継承したメソッド

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.CallEv から
getCall

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス com.cisco.jtapi.extensions.CiscoCallEv から
getCiscoCause, getCiscoFeatureReason

関連資料

「定数フィールド値」（P.F-1）および getTransferControllers() を参照してください。

CiscoTransferStartEv

CiscoTransferStartEv イベントは、転送操作が開始したことを示します。このイベントは CallControlCallObserver インターフェイスによって報告されます。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

スーパーインターフェイス

javax.telephony.events.CallEv, CiscoCallEv, CiscoEv, javax.telephony.events.Ev

宣言

public interface CiscoTransferStartEv extends CiscoCallEv

フィールド

表 6-202　CiscoTransferStartEv のフィールド

インターフェイス	フィールド
static final int	ID

継承したフィールド

インターフェイス com.cisco.jtapi.extensions.CiscoCallEv から
CAUSE_ACCESSINFORMATIONDISCARDED, CAUSE_BARGE,
CAUSE_BCBPRESENTLYAVAIL, CAUSE_BCNAUTHORIZED, CAUSE_BEARERCAPNIMPL,
CAUSE_CALLBEINGDELIVERED, CAUSE_CALLIDINUSE,
CAUSE_CALLMANAGER_FAILURE, CAUSE_CALLREJECTED, CAUSE_CALLSPLIT,
CAUSE_CHANTYPENIMPL, CAUSE_CHANUNACCEPTABLE,
CAUSE_CTICCMSIP400BADREQUEST, CAUSE_CTICCMSIP401UNAUTHORIZED,
CAUSE_CTICCMSIP402PAYMENTREQUIRED, CAUSE_CTICCMSIP403FORBIDDEN,
CAUSE_CTICCMSIP404NOTFOUND, CAUSE_CTICCMSIP405METHODNOTALLOWED,
CAUSE_CTICCMSIP406NOTACCEPTABLE,
CAUSE_CTICCMSIP407PROXYAUTHENTICATIONREQUIRED,
CAUSE_CTICCMSIP408REQUESTTIMEOUT, CAUSE_CTICCMSIP410GONE,
CAUSE_CTICCMSIP411LENGTHREQUIRED,
CAUSE_CTICCMSIP413REQUESTENTITYTOOLONG,
CAUSE_CTICCMSIP414REQUESTURITOOLONG,
CAUSE_CTICCMSIP415UNSUPPORTEDMEDIATYPE,
CAUSE_CTICCMSIP416UNSUPPORTEDURISCHEME,
CAUSE_CTICCMSIP420BADEXTENSION, CAUSE_CTICCMSIP421EXTENSTIONREQUIRED,
CAUSE_CTICCMSIP423INTERVALTOOBRIEF,
CAUSE_CTICCMSIP480TEMPORARILYUNAVAILABLE,
CAUSE_CTICCMSIP481CALLLEGDOESNOTEXIST, CAUSE_CTICCMSIP482LOOPDETECTED,
CAUSE_CTICCMSIP483TOOMANYHOOPS, CAUSE_CTICCMSIP484ADDRESSINCOMPLETE,
CAUSE_CTICCMSIP485AMBIGUOUS, CAUSE_CTICCMSIP486BUSYHERE,
CAUSE_CTICCMSIP487REQUESTTERMINATED,
CAUSE_CTICCMSIP488NOTACCEPTABLEHERE, CAUSE_CTICCMSIP491REQUESTPENDING,
CAUSE_CTICCMSIP493UNDECIPHERABLE,
CAUSE_CTICCMSIP500SERVERINTERNALERROR,
CAUSE_CTICCMSIP501NOTIMPLEMENTED, CAUSE_CTICCMSIP502BADGATEWAY,
CAUSE_CTICCMSIP503SERVICEUNAVAILABLE, CAUSE_CTICCMSIP504SERVERTIMEOUT,
CAUSE_CTICCMSIP505SIPVERSIONNOTSUPPORTED,
CAUSE_CTICCMSIP513MESSAGETOOLARGE, CAUSE_CTICCMSIP600BUSYEVERYWHERE,
CAUSE_CTICCMSIP603DECLINE, CAUSE_CTICCMSIP604DOESNOTEXISTANYWHERE,
CAUSE_CTICCMSIP606NOTACCEPTABLE, CAUSE_CTICONFERENCEFULL,
CAUSE_CTIDEVICENOTPREEMPTABLE, CAUSE_CTIDROPCONFEREE,
CAUSE_CTIMANAGER_FAILURE, CAUSE_CTIPRECEDENCECALLBLOCKED,
CAUSE_CTIPRECEDENCELEVELEXCEEDED,
CAUSE_CTIPRECEDENCEOUTOFBANDWIDTH, CAUSE_CTIPREEMPTFORREUSE,
CAUSE_CTIPREEMPTNOREUSE, CAUSE_DESTINATIONOUTOFORDER,
CAUSE_DESTNUMMISSANDDCNOTSUB, CAUSE_DPARK, CAUSE_DPARK_REMINDER,

CAUSE_DPARK_UNPARK, CAUSE_EXCHANGEROUTINGERROR, CAUSE_FAC_CMC, CAUSE_FACILITYREJECTED, CAUSE_IDENTIFIEDCHANDOESNOTEXIST, CAUSE_IENIMPL, CAUSE_INBOUNDBLINDTRANSFER, CAUSE_INBOUNDCONFERENCE, CAUSE_INBOUNDTRANSFER, CAUSE_INCOMINGCALLBARRED, CAUSE_INCOMPATABLEDDESTINATION, CAUSE_INTERWORKINGUNSPECIFIED, CAUSE_INVALIDCALLREFVALUE, CAUSE_INVALIDIECONTENTS, CAUSE_INVALIDMESSAGEUNSPECIFIED, CAUSE_INVALIDNUMBERFORMAT, CAUSE_INVALIDTRANSITNETSEL, CAUSE_MANDATORYIEMISSING, CAUSE_MSGNCOMPATABLEWCS, CAUSE_MSGTYPENCOMPATWCS, CAUSE_MSGTYPENIMPL, CAUSE_NETOUTOFORDER, CAUSE_NOANSWERFROMUSER, CAUSE_NOCALLSUSPENDED, CAUSE_NOCIRCAVAIL, CAUSE_NOERROR, CAUSE_NONSELECTEDUSERCLEARING, CAUSE_NORMALCALLCLEARING, CAUSE_NORMALUNSPECIFIED, CAUSE_NOROUTETODDESTINATION, CAUSE_NOROUTETOTRANSITNET, CAUSE_NOUSERRESPONDING, CAUSE_NUMBERCHANGED, CAUSE_ONLYRDIVEARERCAPAVAIL, CAUSE_OUTBOUNDCONFERENCE, CAUSE_OUTBOUNDTRANSFER, CAUSE_OUTOFBANDWIDTH, CAUSE_PROTOCOLERRORUNSPECIFIED, CAUSE_QSIG_PR, CAUSE_QUALOFSERVNAVAIL, CAUSE_QUIET_CLEAR, CAUSE_RECOVERYONTIMEREXPIRY, CAUSE_REDIRECTED, CAUSE_REQCALLIDHASBEENCLEARED, CAUSE_REQCIRCNAVIL, CAUSE_REQFACILITYNIMPL, CAUSE_REQFACILITYNOTSUBSCRIBED, CAUSE_RESOURCESNAVAIL, CAUSE_RESPONSETOSTATUSENQUIRY, CAUSE_SERVNOTAVAILUNSPECIFIED, CAUSE_SERVOPERATIONVIOLATED, CAUSE_SERVOROPTNAVAILORIMPL, CAUSE_SUBSCRIBERABSENT, CAUSE_SUSPCALLBUTNOTTHISONE, CAUSE_SWITCHINGEQUIPMENTCONGESTION, CAUSE_TEMPORARYFAILURE, CAUSE_UNALLOCATEDNUMBER,CAUSE_USERBUSY

インターフェイス javax.telephony.events.Ev から

CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

インターフェイス javax.telephony.events.Ev から

CAUSE_CALL_CANCELLED, CAUSE_DEST_NOT_OBTAINABLE, CAUSE_INCOMPATIBLE_DESTINATION, CAUSE_LOCKOUT, CAUSE_NETWORK_CONGESTION, CAUSE_NETWORK_NOT_OBTAINABLE, CAUSE_NEW_CALL, CAUSE_NORMAL, CAUSE_RESOURCES_NOT_AVAILABLE, CAUSE_SNAPSHOT, CAUSE_UNKNOWN, META_CALL_ADDITIONAL_PARTY, META_CALL_ENDING, META_CALL_MERGING, META_CALL_PROGRESS, META_CALL_REMOVING_PARTY, META_CALL_STARTING, META_CALL_TRANSFERRING, META_SNAPSHOT, META_UNKNOWN

メソッド

表 6-203　CiscoTransferStartEv のメソッド

インターフェイス	メソッド	説明
javax.telephony.Call	getTransferredCall()	転送の予定されているコールを返します。
javax.telephony.Call	getFinalCall()	転送の終了後にアクティブになっているコールを返します。
javax.telephony.TerminalConnection	getTransferController()	現在、最後のコールの転送コントローラとして機能している ACTIVE TerminalConnection を返します。transferController が SharedLine の場合、複数の TerminalConnection オブジェクトがあります。このメソッドは ACTIVE TerminalConnection を返します。ただし、アプリケーションが ACTIVE TerminalConnection を監視していない場合、このメソッドは PASSIVE TerminalConnection オブジェクトのうち 1 つを返します。
javax.telephony.TerminalConnection[]	getTransferControllers()	現在、最後のコールの転送コントローラとして機能している TerminalConnection オブジェクトのリストを返します。transferController が SharedLine ではないときは、TerminalConnection だけがリストに含まれます。転送コントローラにオブザーバがない場合、このメソッドは null を返します。
javax.telephony.Address	getTransferControllerAddress()	現在、最後のコールの転送コントローラとして機能しているアドレスを返します。
String	getControllerTerminalName()	転送が行われるコントローラの端末名を返します。

継承したメソッド

インターフェイス com.cisco.jtapi.extensions.CiscoCallEv から
getCiscoCause, getCiscoFeatureReason

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

インターフェイス javax.telephony.events.CallEv から
getCall

インターフェイス javax.telephony.events.Ev から
getCause, getID, getMetaCode, getObserved, isNewMetaEvent

関連資料

「定数フィールド値」（P.F-1）および getTransferControllers() を参照してください。

CiscoUrlInfo

CiscoUrlInfo オブジェクトは、SIP エンドポイントに関連付けられた Uniform Resources Locator（URL）のプロパティを指定します。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(1 および 2)	変更を記録するために履歴表が作成されました。

宣言

public interface CiscoUrlInfo

フィールド

表 6-204　CiscoUrlInfo のフィールド

インターフェイス	フィールド	説明
static final int	TRANSPORT_TYPE_UDP	エンドポイントは UDP を使用しています。
static final int	TRANSPORT_TYPE_TCP	エンドポイントは TCP を使用しています。
static final int	URL_TYPE_UNKNOWN	不明なタイプの URL です。
static final int	URL_TYPE_TEL	テレフォニー タイプの URL です。
static final int	URL_TYPE_SIP	SIP タイプの URL です。

メソッド

表 6-205　CiscoUrlInfro のメソッド

インターフェイス	メソッド	説明
java.lang.String	getUser()	SIP エンドポイントに関連付けられたユーザ名を文字列で返します。
java.lang.String	getHost()	SIP エンドポイントに関連付けられたホスト名を返します。
int	getPort()	SIP エンドポイントに関連付けられたポートを返します。

表 6-205　CiscoUrlInfro のメソッド

インターフェイス	メソッド	説明
int	getTransportType()	SIP エンドポイントで使用されているトランスポート レイヤ プロトコルのタイプを返します。タイプは CiscoUrlInfo.TRANSPORT_TYPE_UDP または CiscoUrlInfo.TRANSPORT_TYPE_TCP のいずれかです。
int	getUrlType()	このメソッドは、エンドポイント URL タイプを返します。CiscoUrlInfo.URL_TYPE_UNKNOWN, CiscoUrlInfo.URL_TYPE_TEL、および CiscoUrlInfo.URL_TYPE_SIP が可能性のある戻り値です。

関連資料

「定数フィールド値」（P.F-1）を参照してください。

ComponentUpdater

指定されたディレクトリにアップデータのログを作成する、オーバーロードされたメソッドが導入されました。

インターフェイス履歴

Cisco Unified Communications Manager リリース番号	説明
7.1(2)	7.1（2）で追加されました。

宣言

public interface ComponentUpdater

メソッド

指定されたディレクトリにアップデータのログを作成する、オーバーロードされたメソッドが導入されました。

表 6-206　ComponentUpdater のメソッド

インターフェイス	メソッド	説明
ComponentUpdater	String tracePath	コンサルト オペレーションがキャンセルされるコンサルト コールを返します。コンサルト コールが存在しない場合、NULL を返します。

関連資料

「定数フィールド値」（P.F-1）を参照してください。