



Preface

This chapter introduces Cisco JTAPI implementation, describes the purpose of this document, and outlines the required software. The chapter includes the following topics:

- [Introduction](#)
- [Purpose](#)
- [Audience](#)
- [New and Changed Information](#)
- [Organization](#)
- [Related Documentation](#)
- [Required Software](#)
- [Conventions](#)
- [Obtaining Documentation](#)
- [Documentation Feedback](#)
- [Obtaining Technical Assistance](#)
- [Obtaining Additional Publications and Information](#)

Introduction

Java Telephony Application Programming Interface (JTAPI) acts as a portable, object-oriented API for computer telephony integrated call control. The package of JTAPI interfaces, located in the `javax.telephony.*` hierarchy, defines a programming model by which Java applications interact with telephony resources such as PBXs and telephones. The Cisco JTAPI implementation supports Java application access to communication systems according to the JTAPI v 1.2 specification. Furthermore, Cisco JTAPI exposes Cisco-specific events and methods for certain telephony resources such as calls and connections.

Purpose

Providing an unchanging programming interface under which varied implementations may stand represents one of the primary goals of a standard Application Programming Interface (API) such as JTAPI. In implementing JTAPI for the Cisco CallManager platform, Cisco tries to conform as closely as possible to the JTAPI specification while providing extensions that enhance JTAPI and expose the advanced features of Cisco CallManager to applications.

As new versions of Cisco CallManager and the Cisco JTAPI implementation are released, variances in the API should be very minor and should tend in the direction of compliance. Cisco remains committed to maintaining its API extensions with the same stability and reliability, though additional extensions may be provided as new Cisco CallManager features become available.

This document outlines some basic JTAPI concepts including transfer and conference extensions. It also describes the support of extensions to the Sun JTAPI v 1.2 specification.

Audience

This document applies for telephony software developers who are developing Cisco IP Telephony applications that require JTAPI. This document assumes that the programmer is familiar with both the Java language and the Sun JTAPI v 1.2 specification.

New and Changed Information

This section describes new features and changes for Cisco JTAPI that are pertinent to the specified release of Cisco CallManager. This section also lists caveats that apply to a particular release, where applicable.

Cisco CallManager 4.2(1)

This section describes the changes to Cisco JTAPI for Cisco CallManager 4.2(1).

Cisco CallManager Version Compatibility

It is not necessary to upgrade Cisco JTAPI version 2.1(x) when upgrading to Cisco CallManager 4.2(1), unless you want to use features that are new in Cisco JTAPI 2.2. Previously, it was required to upgrade Cisco JTAPI when upgrading Cisco CallManager.

If you downgrade Cisco CallManager 4.2 to version 4.1, you must also downgrade Cisco JTAPI to from version 2.2 to version 2.1(x).

Forwarding on No Bandwidth & Unregistered DN

This feature enhances the forwarding logic to handle the No Bandwidth & Unregistered DN cases:

- **No Bandwidth:** When a call cannot be delivered to a remote destination due to no bandwidth, the call is rerouted to the AAR Destination Mask or Voice Mail. The user makes these configuration changes from the directory number page of the Cisco Call Manager GUI.
- **Unregistered DN:** When a call is placed to an unregistered DN, the call is delivered to a DN configured for Call Forward on No Answer (CFNA), as in previous releases.

Transfer Invoked Directed Call Park

This feature allows the user to park a call by transferring the call to a user-selected park code.

Examples

A calls B, B transfers the call to a dPark DN. On completion of transfer, the A to B call is parked at the specified dPark DN. A will hear MOH (if configured). When C un parks the call (by dialing the prefix code and park code), A and C are connected.

If A calls the dPark DN directly, A is connected to the dPark DN and this dPark DN will be marked as busy. A will be connected to this dPark DN until park reversion.

If C does not un park the call at the dPark DN, the call is park reversed back to the DN which parked the call (B) and A and B are connected again. B can again try to d-Park to another dPark DN. When park reversion occurs, Cisco JTAPI passes a new reason code to the application.

CTI sends the dPark number to Cisco JTAPI in the form “Park Number : (<Prefix Code>)<DPark DN>”. Cisco JTAPI parses this and exposes both “Prefix Code” and “DPark DN” to applications.

When a call is un parked, the parked party and un parking party both receive a CPIC event with the reason given by CTI, and the parked party is connected to the un parking party.

When party A calls a dPark DN and party B also calls the same dPark DN, either A or B can be connected to the dPark DN and the other party will be disconnected.

Cisco JTAPI Support

Cisco JTAPI supports this feature. When a call is transferred to a directed call park DN (d parked), the application sees a connection created for directed call park DN and the call control connection state is `CallControlConnection.QUEUED`. `CiscoTransferstart` and `end` events are delivered. An application can use the new interface on `CiscoConnection` to get the prefix code needed to un park the call.

Performance and Scalability

This feature has the same performance impact as the existing transfer feature.

VoiceMailBox Support

This feature exposes voicemailbox numbers, allowing Cisco JTAPI applications to forward calls from a directory number to the correct voice mailbox.

The Cisco CallManager Administrator can associate a voicemail profile for each directory number. When the voicemail option is enabled for any forward setting, and if the corresponding forward is enabled, the call rolls down to the voicemail pilot number associated with the voicemail profile.

The voicemail profile contains voicemail pilot number and voicemailboxmask fields. Voicemailboxmask specifies the mask used to format the voicemail box number for auto-registered phones. When forwarding a call to voice mail from a directory line on an auto-registered phone, Cisco CallManager applies this mask to the number that is configured in the Voice Mail Box field for that directory line.

For example, if you specify a mask of 972813XXXX, the voice-mail box number for directory number 7253 becomes 9728137253. If you do not enter a mask, the voice-mail box number is the same as the directory number (7253 in this example).

Cisco JTAPI Support

To support this feature, Cisco JTAPI exposes voicemail box numbers for called party, lastRedirected party and originalCalled party. These voicemailbox fields are exposed on CiscoPartyInfo, which is exposed on CiscoCall object. If voicemail is not configured for a party, then Cisco JTAPI will return empty Strings for voicemailbox fields.

In prior releases Cisco JTAPI did not expose voicemailbox fields to applications, so Cisco JTAPI voicemailbox applications could not determine whether a voicemailboxmask was configured for a voicemail profile, which could result in a voicemailbox number that is different from the directory number.

Performance and Scalability

This feature does not increase the traffic from the Cisco JTAPI layer to the application layer. However, there could be small performance impact because of additional fields passed over the network.

Privacy On Hold

This feature enhances the privacy of private held calls. When privacy is enabled, only the phone that placed a call on hold can retrieve that call, and the calling name and number are not displayed.

The feature provides a shared address the ability to determine whether other shared addresses may barge into a call. When privacy is enabled, other shared address cannot barge into the call. Privacy is a terminals property. On IP phones there is a Privacy feature button to enable and disable the privacy feature. Privacy can be dynamically enabled and disabled for the active calls on the terminal. When Privacy is on for a call, the TerminalConnection state available to other shared addresses is set to “In Use.” If Privacy status is changed during the CallProgress, CiscoTermConnPrivacyChangedEvent is delivered to the application.

In prior releases, if Privacy is enabled and the call is put on hold, then all TerminalConnections will be in TermConnHeld state and any other shared Address terminalConnection can unhold the call. In Cisco CallManager 4.2, if the “Enforce Privacy on Held Calls” service parameter is enabled, and if Privacy is enabled for a call, then putting the call on hold does not change the terminalConnections of other shared addresses and they remain in the “In Use” state.

Performance and Scalability

There is no performance impact with this feature because there is no additional traffic generated between Cisco JTAPI, applications, and Cisco CallManager.

Caveats

This section describes the caveats that apply to Cisco JTAPI 4.2.

VoiceMailBox in Conference Scenario

Applications should be aware that during a conference call, voicemailbox information might not be the same, but it should be consistent with the corresponding party.

For example if calledParty is B, then corresponding CalledPartyVoiceMailBox will be B's VoiceMailBox. If calledParty is A, then calledPartyVMBox will be A's VoiceMailBox

CiscoPartyInfo for gateway calls at the first address receiving the call may not be consistent with the address.

OriginalCalledPartyVoiceMailBox

When the reset originalCalled option is used in a redirect request, the voicemail box is not consistent with the party/address.

Example 1 OriginalCalledPartyVoiceMailBox

Scenario:

A calls B

B resets OriginalCalled and issues a redirect request to C

Case 1:

Application is monitoring A, B, and C

A calls B

JTAPI CallInfo before redirect request: calling =A, called=B, currentCalled=B, calledPartyVMBox = B, currentCalledPartyVMBox = B

B redirects to C with reset originalCalled

JTAPI CallInfo: calling =A, called=B, currentCalled=C, calledPartyVMBox = B, currentCalledPartyVMBox = C

Case 2:

For the same scenario, if application is monitoring only C

JTAPI CallInfo: calling =A, originalCalled =C, called = C, originalCalledPartyVMBox = C, currentCalledPartyVMBox = C

Cisco CallManager 4.1(3)

This section describes the changes to Cisco JTAPI for Cisco CallManager 4.1(3).

Windows 2003 Support

Cisco JTAPI provides support for Windows 2003, and the Install Shield for the Cisco_JTAPI plug-in also includes support for Windows 2003. Before Cisco_JTAPI 4.1(3) gets installed on a Windows 2003 client machine, you must first install Sun Java Virtual Machine (JVM).

Before Cisco JTAPI 4.0 gets installed on a Windows 2003 client machine, you must first install Microsoft JVM.

Cisco JTAPI contains no interface changes for Windows 2003 support.

Auto Call Pickup

For Release 4.1(3), Cisco CallManager provides support for the Auto Call Pickup feature. Cisco JTAPI does not support either the new Auto Call Pickup features or the existing Call Pickup features and contains no interface changes for Release 4.1(3). Cisco CallManager does send different call information for Auto Call Pickup feature than for Call Pickup, which causes different call information to be sent to the application.

The following scenarios illustrate the differences in call information between the existing Call Pickup feature and the new Auto Call Pickup feature. In each scenario, A calls B, and C picks up the call.

Call Information for Existing Call Pickup Feature

When C picks up the call, the application at A receives the following call information:

- `call.getCalledAddress()`=B
- `call.getCurrentCallingAddress`=A
- `call.getCurrentCalledAddress`=C
- `call.getLastRedirectedAddress()`=B

Call Information for New Auto Call Pickup Feature

When C picks up the call, the application at A receives the following call information:

- `call.getCalledAddress()`=C
- `call.getCurrentCallingAddress`=A
- `call.getCurrentCalledAddress`=C
- `call.getLastRedirectedAddress()`=B

Caveats

The following caveat applies to Cisco JTAPI 4.1.

CP Requires Previous Calls on the Device to be in the Connected Call State

Call processing requires previous calls on the device to be in connected call state before it answers another calls on the same device. If call processing answers calls without checking for the call state of previous calls on the same device, then CTI might return a successful answer response but the call will not go to connected state and needs to be answered again.

Cisco CallManager Release 4.1(2)

The following list provides the features or changes for Cisco JTAPI in Cisco CallManager release 4.1(2):

- **Device State Server**—Provides the cumulative state of all the addresses on a CTI-supported device. States include IDLE, ACTIVE, ALERTING, and HELD.
- **QSIG Path Replacement**—Provides new interfaces in JTAPI to support Q.Sigaling, which optimizes the real-time path (RTP) when calls are transferred or forwarded to other PBXs that are connected through QSIG trunks.
- **Forced Authorization Code (FAC)**—Forces the user to enter a valid authorization code prior to extending a call. JTAPI provides an event to indicate that an FAC is required.
- **Client Matter Code (CMC)**—Enables the user to enter a code that can used to enter “client matter” information, such as accounting or billing codes. JTAPI provides an event to indicate that a CMC is required.
- **Super Provider**—Enables applications to control and monitor any terminal in a Cisco CallManager cluster.

Caveats for Cisco CallManager Release 4.1(2)

The following paragraphs describe the caveats that apply to Cisco CallManager 4.1(2):

FAC-CMC

Forwarding should not be configured to a DN that requires an FAC or CMC code. Forwarding requests will be successful, but calls will not be forwarded to these DNs and will be rejected.

The application should always terminate the code with “#”; otherwise, the system waits for the T302 timer before extending the call. For these cases, the application could get the `postConditionTimeout` exception for `call.connect()` or `call.consult()`, but the call may actually be offered. If applications need to avoid this, either all the digits with a # terminated string are entered with post-condition timeout (which is by default equals 15 seconds in the JTAPI Prefs UI) in the `PlatformException` or increase the `postcondition` timeout.

When two identical `CiscoToneChangedEvents` are sent to applications, and the second one needs to be ignored if both the codes are entered with # separated upon receiving the first event.

setConferenceController

The party that starts a conference by adding a new party acts as the original conference controller. Only the original conference controller can add new parties into the conference. If the original conference controller drops out of the conference, no other party in that particular conference call can add a new party. Although the conference controller cannot be changed while a conference call is going on, applications can determine which `TerminalConnection` acts as the conference controller when initially setting up a conference call via the `CallControlCall.setConferenceController()` method. The `CallControlCall.getConferenceController()` method returns the current conference controller, or null if there is none. If no conference controller is set, the implementation chooses a suitable `TerminalConnection` when the conferencing feature is invoked.

Consider the following scenario as an example:

Scenario: A, B, C, and D belong to a conference call and all are in the TALKING state. A acts as the conference controller. A attempts to use the SetConferenceController API to change the conference controller to B, gets no error, and drops out of the conference. B then tries to add a new party, E, into the conference but cannot do so.

Interval During DTMF Digits

Change PlayDTMF to allow applications to specify the time delay; now, applications can configure the time delay during DTMF digits through Administration window Service parameter ('generate DTMF delay') for Cisco CallManager.

Shared Lines Support

Cisco JTAPI does not support configuration of same DN from different partitions on the same or any device, but does support configuration of different DNs from different partitions on the same device as well as different devices.

CP requires previous calls on the device to be in connected call state

CP requires previous calls on the device to be in connected call state before answering further calls on the same device. If calls are answered without checking for the call state of previous calls on the same device, then CTI might return a successful answer response, but the call will not go to connected state and needs to be answered again.

Cisco CallManager Release 4.0(1)

The following list provides the features or changes for Cisco JTAPI in Cisco CallManager release 4.0(1):

- Multiple Calls Per DN—Enables applications to have multiple calls on the same line with feature operations.
- Shared Line Support—Provides the following abilities to applications:
 - Control shared DN terminals
 - Hold a call on one Shared DN terminal and unhold the same call from another Shared DN terminal

- Make calls between two Shared lines
- Initiate a call from one Shared line terminal while another active call exists on another Shared line terminal with the same DN
- Transfer and DirectTransfer—Provides the following enhancements:
 - Application can transfer two held calls.
 - Application can have one held call and one connected call in any order.
 - Application can transfer any two calls that are present on the line.
- Conference and Join—Enhances ability to perform Arbitrary Conference of multiple calls
- Barge, CBarge, and Privacy Event Notification—For Barge and CBarge, supports manual feature activation on the application-controlled IP phones. The system does not support feature activation through the API. The Privacy feature provides a shared address with the ability to enable or disable other shared addresses to Barge into a call.
- CallSelect and UnSelect Even Notification—Provides events for applications when they monitor RemoteInUse terminals. Applications cannot invoke an API on the Passive or InUse TerminalConnection.
- Dynamic CTIPort Registration Per Call—Enables applications to provide an ipAddress and port number for each call or whenever media gets established.
- Media Termination at Route Point—Enables applications to terminate media for all active calls by specifying an ipAddress and port number for each call or whenever media gets established.
- Redirect Set Original Called ID—Provides applications with the ability to specify preferred original called party DN apart from the destination party information in the redirect request.
- Single Step Transfer—Provides applications with the following enhancements:
 - A new call does not get created
 - CiscoTransferStartEv and CiscoTransferEndEv do not get delivered to applications
 - The state of the original call gets retained if the transfer operation fails

- Auto Update of API—Provides a facility by which an application at startup can identify itself to a web server via an HTTP request and receives a response with the version of the required JTAPI API. The application compares the version that is available on the server to the local version in the application classpath and determines whether an upgrade is necessary.
- CiscoTerminal Filter and ButtonPressedEvents—Enables applications to receive the CiscoTermButtonPressedEv when a digit gets pressed on the phone.
- Modifying Calling Number—Enables applications to modify the calling party in the select route API from a route point.
- AutoAccept support for CTIPort and RoutePoint—Provides applications with the ability to enable or disable AutoAccept for the addresses on the CTIPort and RoutePoint. When changes occur to AutoAccept on the address, the application receives CiscoAddrAutoAcceptStatusChangedEv on AddressObservers.
- CiscoTermRegistrationFailed event—Provides the applications with an event when CiscoMediaTerminal or CiscoRouteTerminal registration asynchronously fails.
- SelectRoute Interface Enhancement—Enhances the SelectRoute interface to take the parameters “PreferredOriginalCalledNumber” and “PreferredOriginalCalledOption,” which enables applications to reset the OriginalCalled value to a specified “PreferredOriginalCalledNumber” when the call gets routed.
- Presentation Indicator (PI) for the Call—Provides applications with the ability to hide or reveal Calling/Called/CurrentCalling/CurrentCalled/LastRedirecting parties name and number to the end user.

Caveats

The following caveat applies to Cisco CallManager 4.0.

Transferring to a Conference

In Cisco JTAPI, whenever Transfer connects a normal call to a conference call, the GlobalCallId of the conference call always survives regardless of whether Call 1 or Call 2 performs the transfer.

When transferring to a conference, the system might deliver external connection creation events before `CiscoTransferStartedEv`, however, the system will still deliver Call Merge events within the `CiscoTransferStart` and `CiscoTransferEnd` boundary.

Cisco CallManager Release 3.3

The following list provides the features or changes for Cisco JTAPI in Cisco CallManager release 3.3:

- `CallParkRequest`, `CallParkResponse`, and `Parked DN Monitoring`—Defines new extensions to allow applications to park a call or unpark a call.
- `XSI Object Pass Through`—Allows application to pass an XML object through a Cisco JTAPI or CTI interface to an IP phone.
- `VG248 and ATA 186 Analog Phone Gateways`—Supports control of analog phones that are connected to these gateways.
- `Cisco JTAPI Installation Internationalization`—Supports multiple languages for the Cisco JTAPI installation and the user preference user interface. Refer to the *Cisco CallManager 3.3 JTAPI Installation Guide* for more information.
- `Enable or Disable Ringer`—Supports application control of ringer settings for each address on a device.
- `Clear Calls Interface`—Provides a `clearCallConnections` interface that allows applications to remove phantom calls without removing the call observer.
- `Display Name Interface`—Extends the `CiscoCall` interface to provide methods to get name displays of the calling party and the called party in a call. Applications can use `getCurrentCallingPartyDisplayName()` to get the display name of the calling party.
- `SetMessageWaiting Interface`—Provides a method for applications to set the message-waiting lamp or indicator for an address.
- `Quite Clear`—Provides `QuiteClear` at the other end when two parties are on a call and one address goes `OutOfService` because of a network outage, the Cisco CallManager goes down, application controlling `CTIPort` goes down, or `CTIManager` goes down.
- `GetCallInfo Interface`—Provides applications with the ability to query `CallInfo` on an address. A query returns the `CiscoAddressCallInfo` object, which contains information about the number of active or held calls,

maximum number of active or held calls, and the Call object for current calls on the address. This interface also provides information regarding what calls are at a specific address at a specific time.

- **DeleteCall Interface**—Provides applications with the ability to delete a call that was created by using the createCall interface. This method accepts a call and throws an InvalidStateException if a provider is not in service or if the call is not in the IDLE state. DeleteCall moves the call to the INVALID state.
- **GetGCID**—Provides an interface on the CiscoCallID to get the nodeID and the GCID of the call, which exposes the GCID information that is available in the internal call object.
- **GetCallID in RTP Events**—Provides an interface on RTP events to access any call information, so applications can link RTP events with the calls.

Cisco CallManager Release 3.2

The following list provides the features or changes for Cisco JTAPI in Cisco CallManager release 3.2:

- **Call Park**—Cisco JTAPI supports user interactions with Call Park and reports appropriate events to the applications.
- **Super Provider**—Supports static control of devices and the ability to query for devices.
- **Reconnect Logic**—Connects Cisco JTAPI applications to the secondary CTIManager after waiting for a random time, so all Cisco JTAPI applications do not connect to the secondary CTIManager at the same time.

Cisco CallManager Release 3.1

The following list provides the features or changes for Cisco JTAPI in Cisco CallManager release 3.1:

- **CTIManager Component**—Allows a JTAPI application to control devices on another Cisco CallManager in a cluster. It supports multiple Cisco CallManagers and CTI managers during failover and recovery and supports automatic device recovery during a failover.

- Directory Change Notification—Allows asynchronous directory change notification.
- Transfer and Conference Enhancement—Allows enhancements to Transferring & Conferencing.
- Call Forward Setting—Allows Cisco JTAPI implementation for supporting Call Forwarding.
- CiscoJtapiExceptions—Allows CiscoJtapiException handling modifications.
- Redirect—Allows a Cisco JTAPI Redirect request.
- Alarm Services—Allows Cisco JTAPI support for Alarm Services.
- Application Control of JTAPI Parameters—Allows control of the parameters within jtapi.ini.
- Dynamic Trace Enabling Using Jtprefs—Allows dynamic enabling of traces from the Jtprefs application.

Organization

The following table provides an outline of this document’s organization.

Chapter	Description
Chapter 1, “Overview”	This chapter introduces the major concepts with which you need to be familiar before creating JTAPI applications for Cisco IP Telephony Solutions.
Chapter 2, “Cisco JTAPI Implementation”	This chapter describes the interfaces and classes that are available.
Chapter 3, “JTAPI Examples”	This chapter provides the source code for makecall, which is the Cisco JTAPI program that is used to test the JTAPI installation.
Appendix A, “Message Sequence Charts”	This appendix contains message flow diagrams.

Chapter	Description
Appendix B, “Cisco JTAPI Classes and Interfaces”	This appendix contains a listing of all the classes and interfaces that are available in the Cisco JTAPI implementation for Cisco CallManager.
Appendix C, “Troubleshooting CiscoJTAPI”	This appendix contains CTI Error Codes, CiscoEvent IDs, and other information to assist with troubleshooting efforts.

Related Documentation

To obtain the latest version of the complete Sun Microsystems JTAPI specification files, go directly to the following web site:

- <http://java.sun.com/products/jtapi>

Required Software

The following table lists software requirements for the following applications: JTAPI applications, JTPREFS, and sample code.

Application	Required Software	Examples
JTAPI applications	Any JDK 1.1 compliant java environment	<ul style="list-style-type: none"> • Microsoft Internet Explorer 4.01 or later • Sun JDK 1.1, 1.2, or 1.3
JTPREFS	Any JDK 1.2 compliant environment.	
Sample code	Microsoft Internet Explorer 4.01 or later	

Conventions

This document uses the following conventions:

Convention	Description
boldface font	Commands and keywords are in boldface .
<i>italic font</i>	Arguments for which you supply values are in <i>italics</i> .
[]	Elements in square brackets are optional.
{ x y z }	Alternative keywords are grouped in braces and separated by vertical bars.
[x y z]	Optional alternative keywords are grouped in brackets and separated by vertical bars.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.
screen font	Terminal sessions and information the system displays are in screen font.
boldface screen font	Information you must enter is in boldface screen font .
<i>italic screen font</i>	Arguments for which you supply values are in <i>italic screen font</i> .
	This pointer highlights an important line of text in an example.
^	The symbol ^ represents the key labeled Control—for example, the key combination ^D in a screen display means hold down the Control key while you press the D key.
< >	Nonprinting characters, such as passwords are in angle brackets.

Notes use the following conventions:



Note

Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the publication.

Obtaining Documentation

Cisco documentation and additional literature are available on Cisco.com. Cisco also provides several ways to obtain technical assistance and other technical resources. These sections explain how to obtain technical information from Cisco Systems.

Cisco.com

You can access the most current Cisco documentation at this URL:

<http://www.cisco.com/univercd/home/home.htm>

You can access the Cisco website at this URL:

<http://www.cisco.com>

You can access international Cisco websites at this URL:

http://www.cisco.com/public/countries_languages.shtml

Documentation DVD

Cisco documentation and additional literature are available in a Documentation DVD package, which may have shipped with your product. The Documentation DVD is updated regularly and may be more current than printed documentation. The Documentation DVD package is available as a single unit.

Registered Cisco.com users (Cisco direct customers) can order a Cisco Documentation DVD (product number DOC-DOCDVD=) from the Ordering tool or Cisco Marketplace.

Cisco Ordering tool:

<http://www.cisco.com/en/US/partner/ordering/>

Cisco Marketplace:

<http://www.cisco.com/go/marketplace/>

Ordering Documentation

You can find instructions for ordering documentation at this URL:

http://www.cisco.com/univercd/cc/td/doc/es_inpkc/pdi.htm

You can order Cisco documentation in these ways:

- Registered Cisco.com users (Cisco direct customers) can order Cisco product documentation from the Ordering tool:

<http://www.cisco.com/en/US/partner/ordering/>

- Nonregistered Cisco.com users can order documentation through a local account representative by calling Cisco Systems Corporate Headquarters (California, USA) at 408 526-7208 or, elsewhere in North America, by calling 1 800 553-NETS (6387).

Documentation Feedback

You can send comments about technical documentation to bug-doc@cisco.com.

You can submit comments by using the response card (if present) behind the front cover of your document or by writing to the following address:

Cisco Systems
Attn: Customer Document Ordering
170 West Tasman Drive
San Jose, CA 95134-9883

We appreciate your comments.

Cisco Product Security Overview

Cisco provides a free online Security Vulnerability Policy portal at this URL:

http://www.cisco.com/en/US/products/products_security_vulnerability_policy.html

From this site, you can perform these tasks:

- Report security vulnerabilities in Cisco products.

- Obtain assistance with security incidents that involve Cisco products.
- Register to receive security information from Cisco.

A current list of security advisories and notices for Cisco products is available at this URL:

<http://www.cisco.com/go/psirt>

If you prefer to see advisories and notices as they are updated in real time, you can access a Product Security Incident Response Team Really Simple Syndication (PSIRT RSS) feed from this URL:

http://www.cisco.com/en/US/products/products_psirt_rss_feed.html

Reporting Security Problems in Cisco Products

Cisco is committed to delivering secure products. We test our products internally before we release them, and we strive to correct all vulnerabilities quickly. If you think that you might have identified a vulnerability in a Cisco product, contact PSIRT:

- Emergencies—security-alert@cisco.com
- Nonemergencies—psirt@cisco.com



Tip

We encourage you to use Pretty Good Privacy (PGP) or a compatible product to encrypt any sensitive information that you send to Cisco. PSIRT can work from encrypted information that is compatible with PGP versions 2.x through 8.x.

Never use a revoked or an expired encryption key. The correct public key to use in your correspondence with PSIRT is the one that has the most recent creation date in this public key server list:

<http://pgp.mit.edu:11371/pks/lookup?search=psirt%40cisco.com&op=index&exact=on>

In an emergency, you can also reach PSIRT by telephone:

- 1 877 228-7302
- 1 408 525-6532

Obtaining Technical Assistance

For all customers, partners, resellers, and distributors who hold valid Cisco service contracts, Cisco Technical Support provides 24-hour-a-day, award-winning technical assistance. The Cisco Technical Support Website on Cisco.com features extensive online support resources. In addition, Cisco Technical Assistance Center (TAC) engineers provide telephone support. If you do not hold a valid Cisco service contract, contact your reseller.

Cisco Technical Support Website

The Cisco Technical Support Website provides online documents and tools for troubleshooting and resolving technical issues with Cisco products and technologies. The website is available 24 hours a day, 365 days a year, at this URL:

<http://www.cisco.com/techsupport>

Access to all tools on the Cisco Technical Support Website requires a Cisco.com user ID and password. If you have a valid service contract but do not have a user ID or password, you can register at this URL:

<http://tools.cisco.com/RPF/register/register.do>



Note

Use the Cisco Product Identification (CPI) tool to locate your product serial number before submitting a web or phone request for service. You can access the CPI tool from the Cisco Technical Support Website by clicking the **Tools & Resources** link under Documentation & Tools. Choose **Cisco Product Identification Tool** from the Alphabetical Index drop-down list, or click the **Cisco Product Identification Tool** link under Alerts & RMAs. The CPI tool offers three search options: by product ID or model name; by tree view; or for certain products, by copying and pasting **show** command output. Search results show an illustration of your product with the serial number label location highlighted. Locate the serial number label on your product and record the information before placing a service call.

Submitting a Service Request

Using the online TAC Service Request Tool is the fastest way to open S3 and S4 service requests. (S3 and S4 service requests are those in which your network is minimally impaired or for which you require product information.) After you describe your situation, the TAC Service Request Tool provides recommended solutions. If your issue is not resolved using the recommended resources, your service request is assigned to a Cisco TAC engineer. The TAC Service Request Tool is located at this URL:

<http://www.cisco.com/techsupport/servicerequest>

For S1 or S2 service requests or if you do not have Internet access, contact the Cisco TAC by telephone. (S1 or S2 service requests are those in which your production network is down or severely degraded.) Cisco TAC engineers are assigned immediately to S1 and S2 service requests to help keep your business operations running smoothly.

To open a service request by telephone, use one of the following numbers:

Asia-Pacific: +61 2 8446 7411 (Australia: 1 800 805 227)

EMEA: +32 2 704 55 55

USA: 1 800 553-2447

For a complete list of Cisco TAC contacts, go to this URL:

<http://www.cisco.com/techsupport/contacts>

Definitions of Service Request Severity

To ensure that all service requests are reported in a standard format, Cisco has established severity definitions.

Severity 1 (S1)—Your network is “down,” or there is a critical impact to your business operations. You and Cisco will commit all necessary resources around the clock to resolve the situation.

Severity 2 (S2)—Operation of an existing network is severely degraded, or significant aspects of your business operation are negatively affected by inadequate performance of Cisco products. You and Cisco will commit full-time resources during normal business hours to resolve the situation.

Severity 3 (S3)—Operational performance of your network is impaired, but most business operations remain functional. You and Cisco will commit resources during normal business hours to restore service to satisfactory levels.

Severity 4 (S4)—You require information or assistance with Cisco product capabilities, installation, or configuration. There is little or no effect on your business operations.

Obtaining Additional Publications and Information

Information about Cisco products, technologies, and network solutions is available from various online and printed sources.

- Cisco Marketplace provides a variety of Cisco books, reference guides, and logo merchandise. Visit Cisco Marketplace, the company store, at this URL:
<http://www.cisco.com/go/marketplace/>
- *Cisco Press* publishes a wide range of general networking, training and certification titles. Both new and experienced users will benefit from these publications. For current Cisco Press titles and other information, go to Cisco Press at this URL:
<http://www.ciscopress.com>
- *Packet* magazine is the Cisco Systems technical user magazine for maximizing Internet and networking investments. Each quarter, *Packet* delivers coverage of the latest industry trends, technology breakthroughs, and Cisco products and solutions, as well as network deployment and troubleshooting tips, configuration examples, customer case studies, certification and training information, and links to scores of in-depth online resources. You can access *Packet* magazine at this URL:
<http://www.cisco.com/packet>
- *iQ Magazine* is the quarterly publication from Cisco Systems designed to help growing companies learn how they can use technology to increase revenue, streamline their business, and expand services. The publication identifies the challenges facing these companies and the technologies to help solve them, using real-world case studies and business strategies to help readers make sound technology investment decisions. You can access *iQ Magazine* at this URL:
<http://www.cisco.com/go/iqmagazine>

- *Internet Protocol Journal* is a quarterly journal published by Cisco Systems for engineering professionals involved in designing, developing, and operating public and private internets and intranets. You can access the Internet Protocol Journal at this URL:

<http://www.cisco.com/ipj>

- World-class networking training is available from Cisco. You can view current offerings at this URL:

<http://www.cisco.com/en/US/learning/index.html>

