



Cisco JTAPI Installation and Configuration

Cisco Java Telephony API (JTAPI) implementation comprises Java classes that reside on all client machines running Cisco JTAPI applications. Installation of the Cisco JTAPI implementation must take place before Cisco JTAPI applications will function correctly. Make sure the Cisco JTAPI classes are installed wherever JTAPI applications will run, whether on Cisco CallManager, on a separate machine, or on both. Installation requires 5 MB of local disk space.



Note

If you have upgraded to Cisco CallManager 3.1, you must upgrade the JTAPI client software on any application server or client workstation on which JTAPI applications are installed. If you do not upgrade the JTAPI client, your application will fail to initialize. If you need to upgrade, download the appropriate client from the Cisco CallManager Administration as described in the [“Installing the Cisco JTAPI Software”](#) section on page 48-2.

The upgraded JTAPI client software does not work with older releases of Cisco CallManager.

This section discusses the software installation and configuration required to ready the Cisco JTAPI environment for programming and running applications.

- [Installing the Cisco JTAPI Software, page 48-2](#)
- [Verifying the Installation, page 48-4](#)

- [Configuring Cisco JTAPI Tracing, page 48-4](#)
- [Administering User Information for JTAPI and TAPI Applications, page 48-15](#)

Installing the Cisco JTAPI Software

The Cisco JTAPI installation utility installs the following items on the local disk drive:

- JTAPI java classes in %SystemRoot%\java\lib
- JTAPI Preferences (jtprefs.exe) in Program Files\JTAPITools
- JTAPI sample applications (makecall, jtrace) in Program Files\JTAPITools
- JTAPI documentation in Program Files\JTAPITools\doc



Note

To run JTAPI applications, you need a Java 1.1-compatible environment such as the Sun JDK 1.1.x, JDK 1.2, or Microsoft Virtual Machine (bundled with Internet Explorer 4.0 and higher). Cisco JTAPI will also run on Sun JDK 1.2.

To use JTPREFS with Cisco JTAPI, however, requires Microsoft Java Virtual Machine 5.00.3190 or later. For information on JTPrefs, see the “[Configuring Cisco JTAPI Tracing](#)” section on page 48-4.

If you are installing Cisco JTAPI on a Windows 2000 workstation or server, you already have a compatible version of the Microsoft Virtual Machine. On all other Microsoft platforms such as Windows 95, Windows 98, and Windows NT, verify the current version of the Microsoft Virtual Machine by running the command "jview /?" and noting the version printed at the top of the console. You can download the latest version of the Microsoft Java Virtual Machine from <http://www.microsoft.com/java>.

Perform the following steps to install the Cisco JTAPI software.

Procedure

- Step 1** Log in to the computer where you want to install the Cisco JTAPI client software.
- Step 2** Close all Windows programs.
- Step 3** Open a web browser.
- Step 4** Go to the Cisco CallManager administration windows:
`http://Name/CCMAdmin/main.asp`
where:
- Name* specifies the name or IP address of the Cisco CallManager
- Step 5** Choose **Application > Install Plugins**.
- Step 6** Choose the **Cisco JTAPI** link.
- Step 7** Save the file on your desktop.
- Step 8** Follow the instructions in the popup windows.



Note Install Cisco JTAPI software on the default drive as directed by the installation software. When Windows NT is installed in C:\WINNT, the default directory, for example, is C:\WINNT\Java\lib.

Verifying the Installation

To verify the JTAPI installation, you can use the `makecall` application that allows you to place a call via JTAPI. Perform the following procedure to use the `makecall` application.

Procedure

Step 1 From the Windows NT command line, navigate to the directory where you installed Cisco JTAPI Tools. By default, this directory is `C:\Program Files\JTAPITools`.

Step 2 Execute the following command:

Jview makecall *<server name>* *<login>* *<password>* **1000** *<phone1>* *<phone2>*

where:

server name specifies the hostname or IP address of the Cisco CallManager (for example, CTISERVER).

phone1 and *phone2* designate directory numbers of IP phones or virtual phones that the user controls according to the user configuration. See the [“Adding a User” section on page 44-2](#) for details.

For *login* and *password*, use the user ID and password that you configured in the Cisco CallManager User Configuration window.

Configuring Cisco JTAPI Tracing

Use the Cisco JTAPI tracing preferences application (`JTPREFS.EXE`) to configure trace levels and trace destinations. Installation of the Cisco JTAPI Preferences into the `Program Files\JTAPITools` directory utility takes place by default. To open the Cisco JTAPI Preferences utility, choose **Start > Programs > Cisco JTAPI > JTAPI Preferences**.

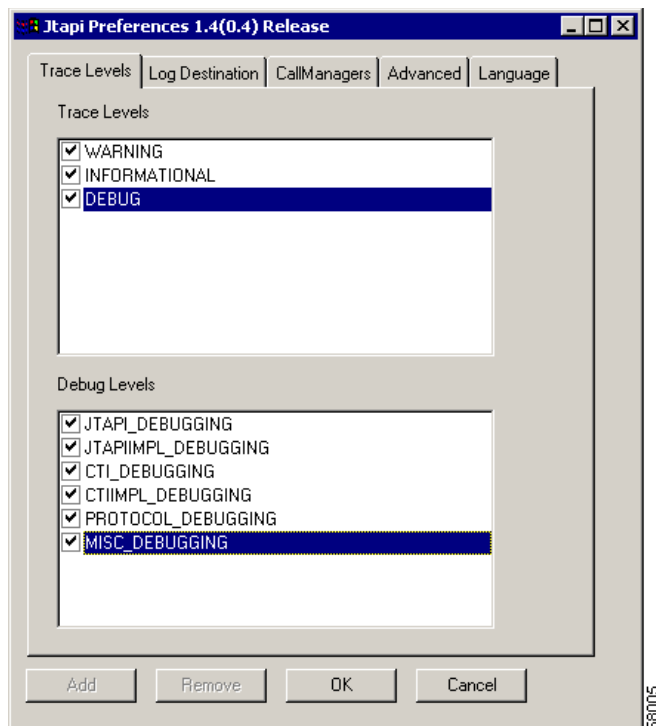
This section, which describes how to use the Cisco JTAPI preferences application, discusses the following topics:

- [Trace Levels, page 48-5](#)
- [Log Destination, page 48-7](#)
- [Cisco CallManager, page 48-10](#)
- [Advanced, page 48-11](#)

Trace Levels

Figure 48-1 illustrates the Trace Levels tab of the Cisco JTAPI preferences application. The window title shows the JTAPI version number.

Figure 48-1 Trace Levels Tab



The Trace Level tab allows you to enable or disable the following JTAPI trace levels:

- WARNING—Low-level warning events
- INFORMATIONAL—Status events
- DEBUG—Highest level debugging events

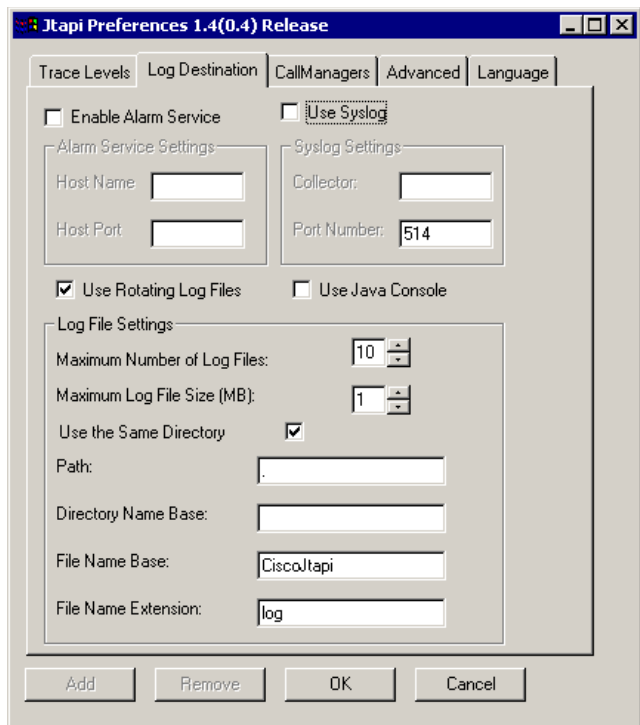
You may enable or disable additional debugging levels in the Debug Levels window, as described in the following list:

- JTAPI_DEBUGGING—JTAPI methods and events trace
- JTAPI_IMPLDEBUGGING—Internal JTAPI implementation trace
- CTI_DEBUGGING—Trace Cisco CallManager events that are sent to the JTAPI implementation
- CTIIMPL_DEBUGGING—Internal CTICLIENT implementation trace
- PROTOCOL_DEBUGGING—Full CTI protocol decoding
- MISC_DEBUGGING—Miscellaneous low-level debug trace

Log Destination

Figure 48-2 illustrates the Log Destination tab of the Cisco JTAPI preferences application.

Figure 48-2 Log Destination Tab



The Log Destination tab allows you to configure how JTAPI creates traces and how they are stored. [Table 48-1](#) contains descriptions of the log destination fields:

Table 48-1 Log Destination Configuration Fields

Field	Description
Enable Alarm Service	When this option is enabled, JTAPI alarms go to an alarm service running on the specified machine. You must specify the host name and port number when enabling this option.
Use Java Console	When this option is enabled, tracing goes to the standard output or console (command) window.
Use Syslog	When this option is enabled, traces go to a UDP port as specified in the Syslog Collector and Port Number fields. Syslog collector service collects traces and directs them to the CiscoWorks2000 server.
Use Rotating Log Files	This option allows you to direct the traces to a specific path and folder in the system. No fewer than two log files and no more than 99 files can exist. Cisco JTAPI rotates through the log files in numerical order, returning to the first log file after filling the last. Log files increase in size in 1-megabyte increments.
Use the Same Directory	<p>This option allows you to specify whether the same folder name should be used for each instance of an application.</p> <p>When the option is enabled, JTAPI traces the log files to the same directory. In this case, successive instances of a JTAPI application will restart the log files starting at index 01.</p> <p>When the option is disabled, each application instance, whether successive or simultaneous, will cause the trace files to be placed in a new folder sequential to the last folder written. Cisco JTAPI detects the last folder present in the trace path and automatically increments the numeric index.</p>

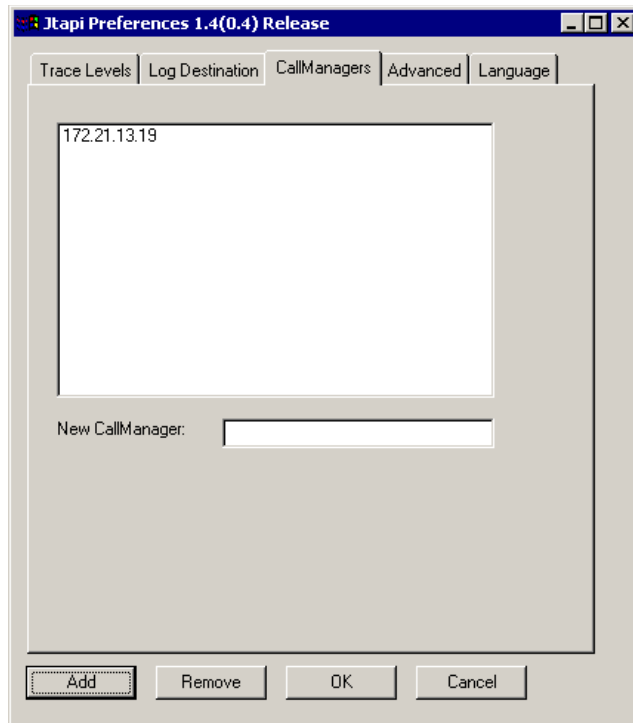
Table 48-1 Log Destination Configuration Fields (continued)

Field	Description
Path	This field allows you to specify the path name to which the trace files are written. When the path is not specified, JTAPI makes the default the application path.
Directory Name Base	This field allows you to specify a folder name where the trace files will be contained.
File Name Base and File Name Extension	<p>Use these values to create the trace file names with a numerical index appended to the file base name to indicate the order in which the files are created.</p> <p>For example, if you enter jtapiTrace in the File Name Base field and log in the File Name Extension field, the trace files would rotate between jtapiTrace01.log, jtapiTrace02.log and jtapiTrace10.log. If the File Name Base and File Name Extension fields are left blank, Cisco JTAPI picks the trace files names as CiscoJtapi01.log, CiscoJtapi02.log, and so on.</p>

Cisco CallManager

Figure 48-3 illustrates the Cisco CallManager tab of the Cisco JTAPI preferences application.

Figure 48-3 CallManager Tab

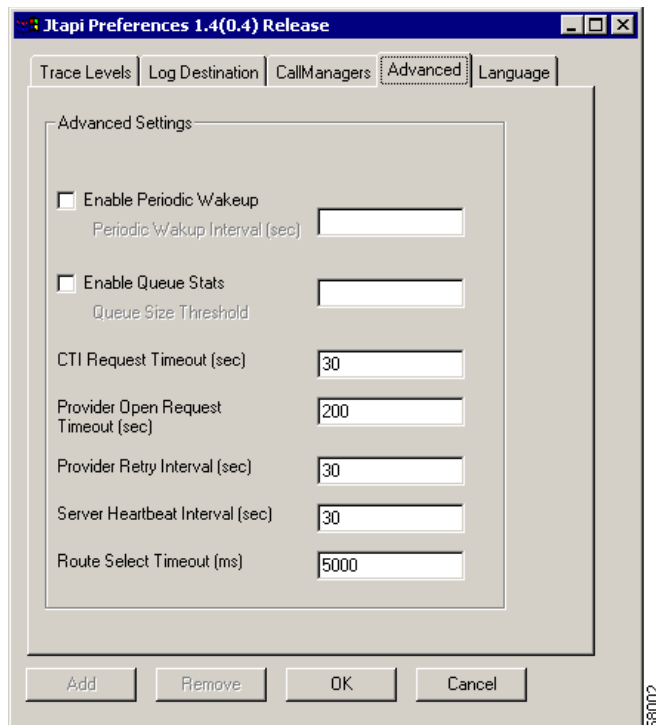


This tab allows you to define a list of Cisco CallManagers that a JTAPI application can present to the user for optional Cisco CallManager connectivity.

Advanced

Figure 48-4 illustrates the Advanced tab of the Cisco JTAPI preferences application.

Figure 48-4 Advanced Tab



You can configure the parameters in [Table 48-2](#) through the Advanced tab in the JTPrefs application. You may need these low-level parameters for troubleshooting and debugging purposes only.

**Note**

Cisco recommends that you *not* modify the parameters in [Table 48-2](#) unless instructed to do so by Cisco Technical Assistance Center (TAC).

Table 48-2 Advanced Configuration Fields

Field	Description
Enable CTI Port Auto Recovery	This setting specifies recovery treatment for CTI ports under the control of the JTAPI application. When this setting is enabled, JTAPI will reregister the CTI port in case of a system failure. When this setting is disabled, the application must reregister CTI ports after a system failure. The default setting is enabled.
Enable Periodic Wakeup	This setting enables a heartbeat in the internal message queue that JTAPI uses. It causes the thread to wake up if it has not received a message in the time defined in the PeriodicWakeupInterval and creates log an event. The default setting is disabled.
Periodic Wakeup Interval (sec)	This setting allows you to define a time of inactivity in the JTAPI internal message thread. If JTAPI has not received a message during this time, the thread wakes up and logs an event. The default is 50 seconds.
Enable Queue Stats	This setting causes JTAPI to log the max queue depth over the specified number of messages queued to JTAPI main event thread. In other words, for every x messages processed, JTAPI logs a DEBUGGING level trace reporting the maximum queue depth over that interval, where x is the number of messages specified in Queue Size Threshold. The default setting is disabled.
Queue Size Threshold	This setting allows you to specify the number of messages that define the time over which JTAPI will report the maximum queue depth. The default is 25 messages.

Table 48-2 Advanced Configuration Fields (continued)

Field	Description
CTI Request Timeout (sec)	This setting specifies the time in seconds that JTAPI will wait for a response from a CTI request. The default is 15 seconds.
Provider Open Request Timeout (sec)	This setting specifies the time in seconds that JTAPI will wait for a response for the Provider Open Request. The default is 30 seconds.
Provider Retry Interval (sec)	This setting specifies the time in seconds that JTAPI will retry opening a connection to the CallManager cluster in case of system failure. The default is 30 seconds.
Server heartbeat Interval (sec)	This setting specifies how often in seconds that the connection between JTAPI and the Cisco CallManager cluster will be verified to be alive. If JTAPI fails to receive heartbeats, it will establish a connection via the second CTIManager specified in the provider open request.
Route Select Timeout (ms)	This setting specifies the time in milliseconds that JTAPI will wait for the application to respond to the Route event. If the application does not respond within this time, JTAPI will end the route and send the corresponding RouteEnd event.

JTAPI Preferences on non-Microsoft environments

For non-Microsoft environments, you must create a jtapi.ini file manually and place it in the CLASSPATH. The following list provides the parameter names with sample values:

```

PROTOCOL_DEBUGGING=0
UseSameDirectory=1
JTAPIIMPL_DEBUGGING=0
UseSystemDotOut=0
QueueStatsEnabled=0
PeriodicWakeupInterval=50
RouteSelectTimeout=5000
UseTraceFile=0
ProviderOpenRequestTimeout=30
Directory=
DEBUG=0
DesiredServerHeartbeatInterval=30
AlarmServicePort=1444
CTI_DEBUGGING=0
SyslogCollector=
JTAPI_DEBUGGING=0
PeriodicWakeupEnabled=0
NumTraceFiles=10
AlarmServiceHostname=
MISC_DEBUGGING=0
TracePath=.
UseAlarmService=0
CTIIMPL_DEBUGGING=0
WARNING=0
Traces=WARNING; INFORMATIONAL; DEBUG
INFORMATIONAL=0
UseSyslog=0
CtiPortAutoRecovery=1
FileNameBase=CiscoJtapi
CtiRequestTimeout=15
TraceFileSize=1048576
Debugging=JTAPI_DEBUGGING; JTAPIIMPL_DEBUGGING; CTI_DEBUGGING;
    CTIIMPL_DEBUGGING; PROTOCOL_DEBUGGING; MISC_DEBUGGING
FileNameExtension=log
QueueSizeThreshold=25
ProviderRetryInterval=30
CallManagers=cml
SyslogCollectorUDPPort=514

```

Administering User Information for JTAPI and TAPI Applications

JTAPI and TAPI applications require that users be administered in the directory and be given privilege to control one or more devices. Follow the procedures for adding a user and assigning devices to a user in the “Adding a New User” section before using a JTAPI or TAPI application. The list of devices assigned to the user represents the phones that the user needs to control from the application (e.g., make calls, answer calls).

