JTAPI Troubleshooting

This chapter contains the following topics:

Overview

Cisco JTAPI is a Java-based telephony applications programming interface that serves as a basic call control API.

JTAPI:

- Has primitive media support.
- Is full call and object oriented
- Provides application portability, i.e., independence from any particular telephony gear or any particular Operating System.

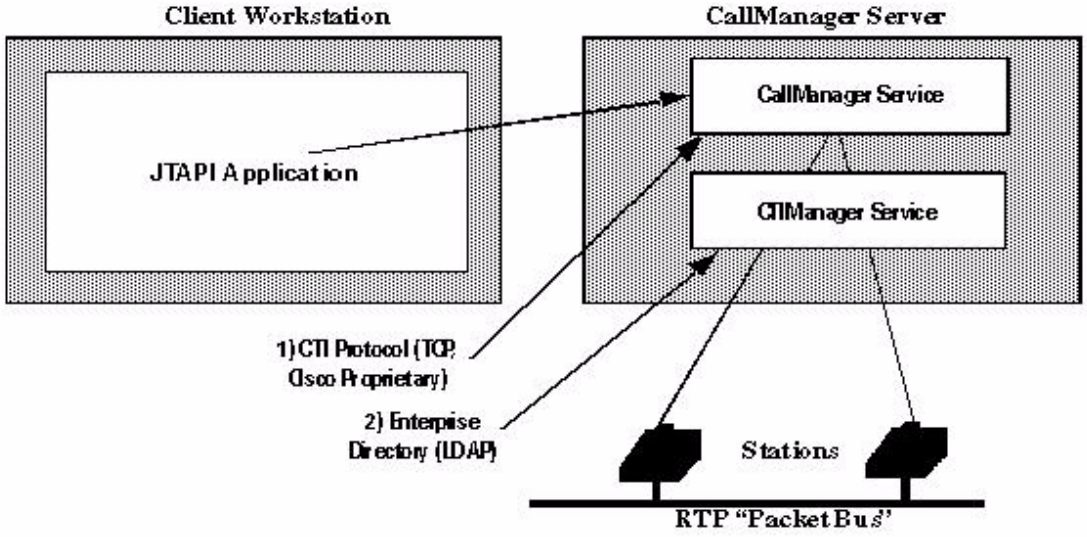
Note Cisco JTAPI is supported on Sun and Microsoft JVMs.

For more detailed information on JTAPI, refer to **http://java.sun.com/products/jtapi/**

Architecture

The Cisco JTAPI package abstracts the underlying Cisco CallManager architecture to applications. It communicates with the Cisco CallManager using a proprietary interface known as CTIQBE. See Figure 13.

Figure 13 JTAPI Architecture



Postinstallation Checklist

Some problems may be the result of improper installation. Use the following checklist to verify proper installation before proceeding with the troubleshooting process:

____ Check that the Cisco CTIManager location is configured properly in the application user interface.

____ Check to see if the Cisco CTIManager is up and running.

____ Check to see that the network connection is up between the JTAPI application server and Cisco CTIManager machines.

____ Check to see if the CallManager DC Directory Service is up and running.

____ Check that the Username and Password that is configured in the JTAPI application matches the Username and Password in the CallManager Directory.

____ Make sure that the Enable CTI Application Use flag has been checked in the Directory Administration on the CallManager for the TSP user.

____ Check that devices have been properly associated with the user in the CallManager Directory.

____ Check to see if the CTI Connection Limit has been reached.

____ Ensure the Cisco JTAPI version is compatible with the version of CallManager. The JTAPI version can be determined by running 'jview CiscoJtapiVersion' from the JTAPI server command line. This version can be verified by checking the CallManager Compatibility Matrix. Refer to the Defaul Loads section on the Cisco Connection Online website at http://www.cisco.com/univercd/cc/td/doc/product/voice/c_callmg/ccmcomp.htm#MinimumVersions

Troubleshooting Tools

Several sample applications are provided to verify the installation and to serve as sample code for developers. Sample applications can be found under the path *c:\Program Files\JTAPITools* and are named *makeCall* and *Jtrace*.

Note *Jtrace* and *makeCall* make use of the Microsoft Windows MFC and will not run under the Sun JDK.

From the command line, navigate to the makeCall directory and enter the following:

```
jview makecall <server name> <login> <password> <delay> <device1> <device2>
```

where:

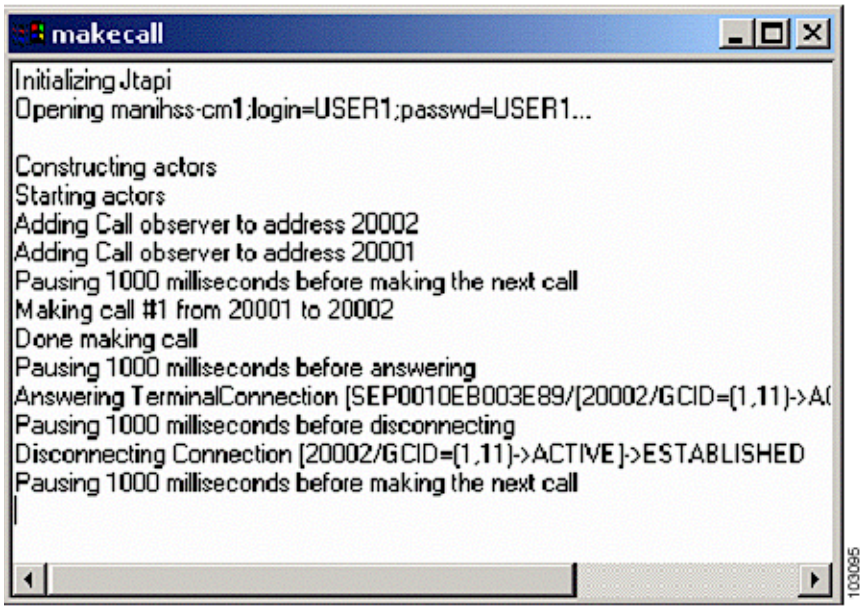
- *Server name* is the host name or IP address of the CTI Manager.
- *Login* and *password* are similar to those administered in the directory.
- Where *delay* is delay between two calls in milliseconds.
- *Device1* and *device2* are the directory numbers of the IP phones.

The application makes the call between 2 devices with an action delay of 1000ms until terminated. Figure 14 shows the result of a makeCall application that has been invoked with following parameters:

```
jview makecall manihss-cm1 USER1 USER1 1000 20001 20002
```

The application continues to make calls until the window is closed.

Figure 14 Running the makeCall application



From the command line, navigate to the jtrace directory and enter the following:

```
jview jtrace.Jtrace <server name> <login> <password> <device1> ... <deviceN>
```

where:

- Server name is the host name or IP address of the CTI Manager.

- Device1 and device2 are the directory numbers of the IP phones.
- Login and password are similar to those administered in the directory.
- Device1 ... deviceN are user controlled devices.

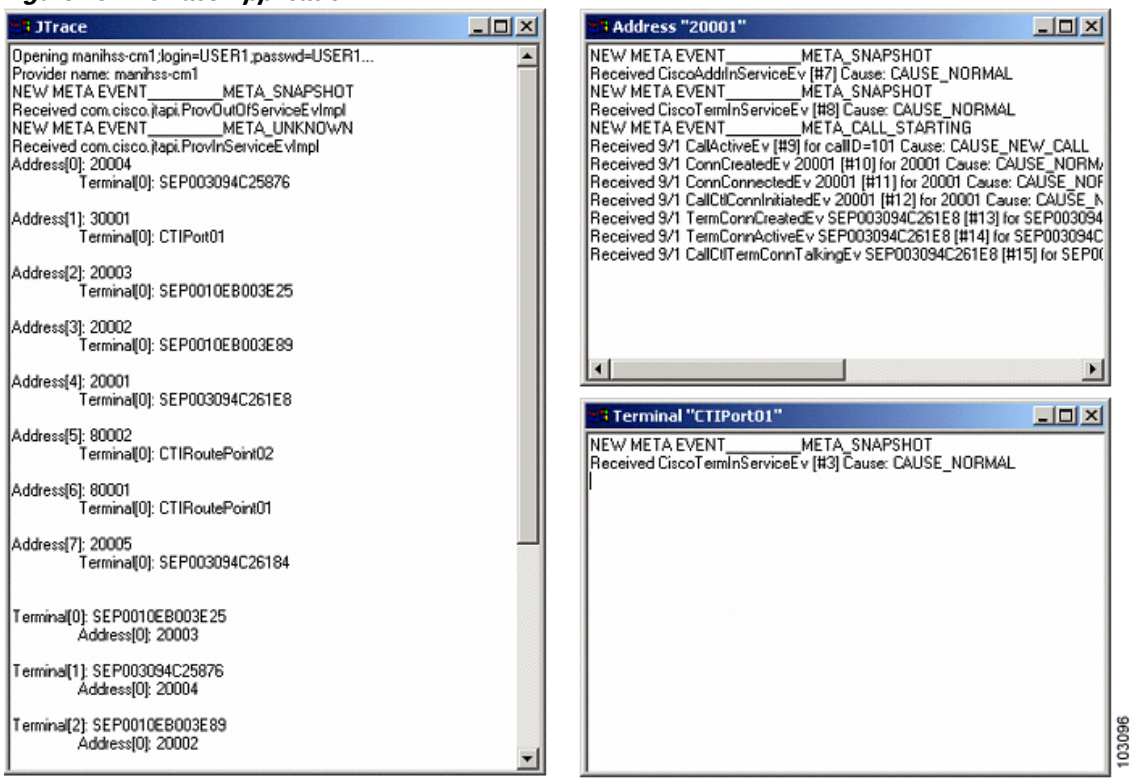
Figure 15 shows the results of a JTrace application that has been invoked with following parameters:

```
jview jtrace.Jtrace manihss-cm1 uesr1 user1 20001
```

The panel on the left shows the JTrace application window. The call manager, login, password, and a list of devices under this user control are listed. *Address* in JTAPI corresponds to a line or DN. *Terminal* corresponds to the device name.

The panel on the right top shows the JTAPI events when the device *20001* goes off hook. JTrace also opens a window for any CTI port in the application userid control list even if not explicitly specified.

Figure 15 JTrace Application



Error Reporting

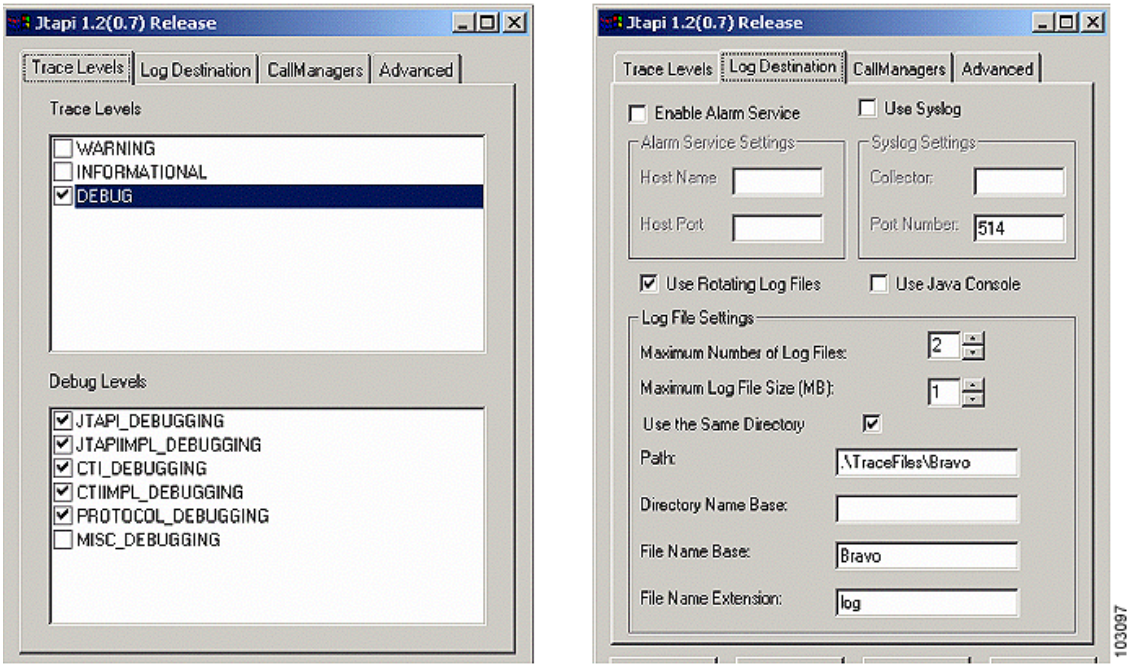
JTAPI Preferences Tool

The JTAPI Preferences Tool (JTPREFS) is a Windows-only tool that can be used to set tracing and error-logging levels and destinations, and to edit the default server names. See Figure 16.

Tracing and logging information can be sent to either rotating log files or to the system console. All settings are stored in the *JTAPI.INI* file.

Note Although the Cisco JTAPI implementation has been fully tested on Windows 2000 and Redhat Linux 7.2 platforms only, it is expected to function on other platforms, such as UNIX, if run under the supported Sun Microsystems JVM.

Figure 16 JTPREFS Tool



Error and Status Codes

For information on applicable error and status codes, refer to the JTAPI Developer Guide at http://www.cisco.com/univercd/cc/td/doc/product/voice/vpdd/cdd/3_3/3_3_3/jtapi/index.htm.