



TAPI Troubleshooting

This chapter contains the following topics:

- [Overview, page 9](#)
- [Architecture, page 10](#)
- [Postinstallation Checklist, page 12](#)
- [Troubleshooting Tools, page 13](#)
- [Error Reporting, page 17](#)
- [Error and Status Codes, page 18](#)

Overview

TAPI

Telephony Applications Programming Interface (TAPI) is part of the Microsoft Windows Open System Architecture and provides developers with the ability to create telephony applications.

TAPI is:

- An open industry standard
- Defined with considerable and ongoing input from the worldwide telephony and computing community
- Switch-fabric independent

TAPI-compatible applications can run on a wide variety of PC and telephony hardware and can support a variety of network services.

TAPI supports many telephony features, allowing developers to telephone-enable virtually any general-purpose application. TAPI also supports Unicode, making it easier to effect global applications. With Active Controls provided by a variety of vendors, corporate developers can put together telephony applications using tools they are already familiar with.

For more information about TAPI refer to

<http://www.microsoft.com/ntserver/techresources/commnet/tele/tapi21.asp>.

TSP

Telephony Service Provider (TSP) is a dynamic-link library (DLL) that provides device-specific controls for communications processing. TSP provides an abstraction layer between TAPI applications and the underlying hardware and transport protocols.

Architecture

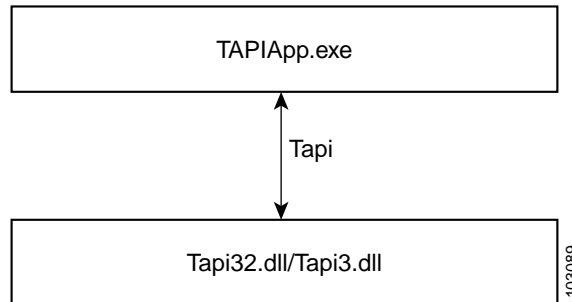
TAPI applications reside in their own process space and communicate with Tapi32.dll or Tapi3.dll using TAPI. See [Figure 1](#).



Note

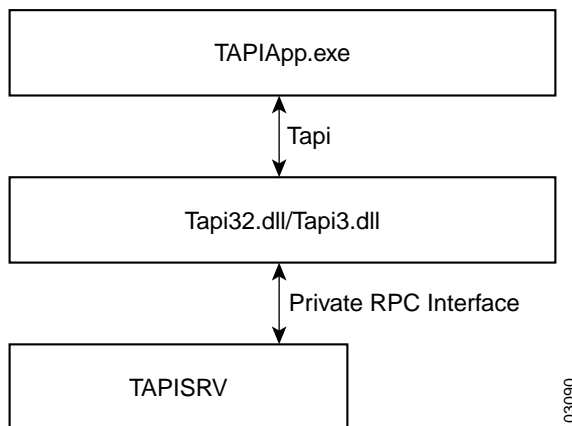
Tapi32.dll and Tapi3.dll are DLLs provided by Microsoft.

Figure 1 Telephony Application PC



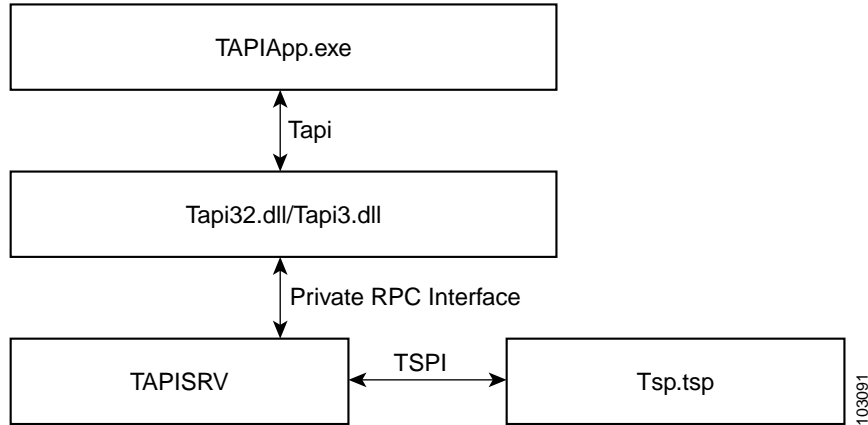
Tapi32.dll and Tapi3.dll communicate with the TAPISRV process using a private RPC interface. TAPISRV is a Microsoft process. In Windows 95, 98, and NT, this process is named TAPISRV.EXE. In Windows 2000, this process runs under the SVCHOST.EXE process. See [Figure 2](#).

Figure 2 Telephony Application PC



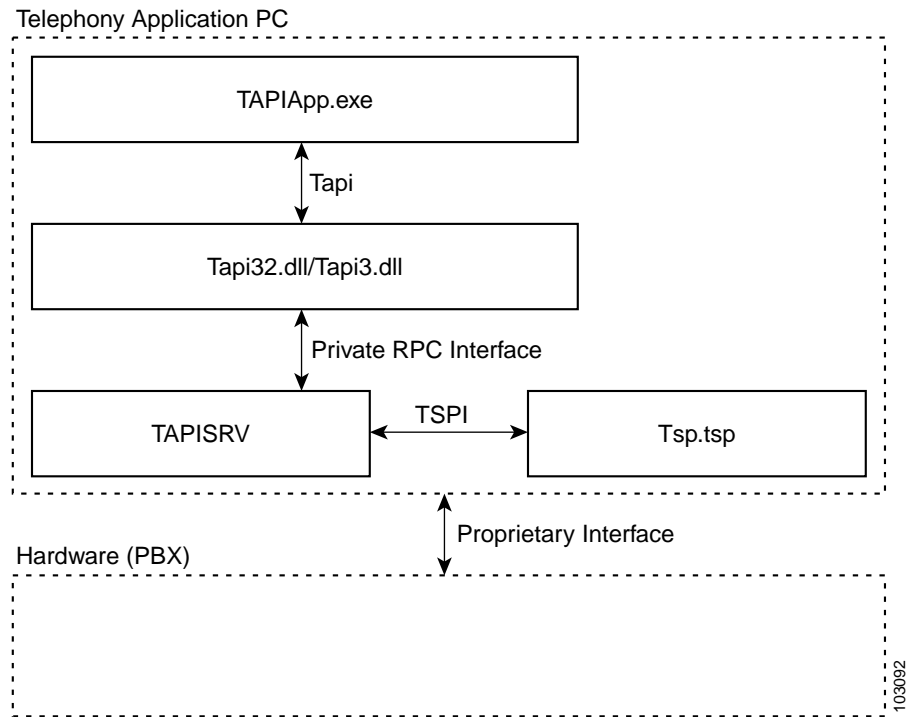
TAPISRV communicates with the TSP using a Microsoft-defined interface known as TSPI. The TSP is a DLL that is dynamically linked to the TAPISRV process. Each PBX vendor provides its own TSP. See [Figure 3](#).

Figure 3 *Telephony Application PC*



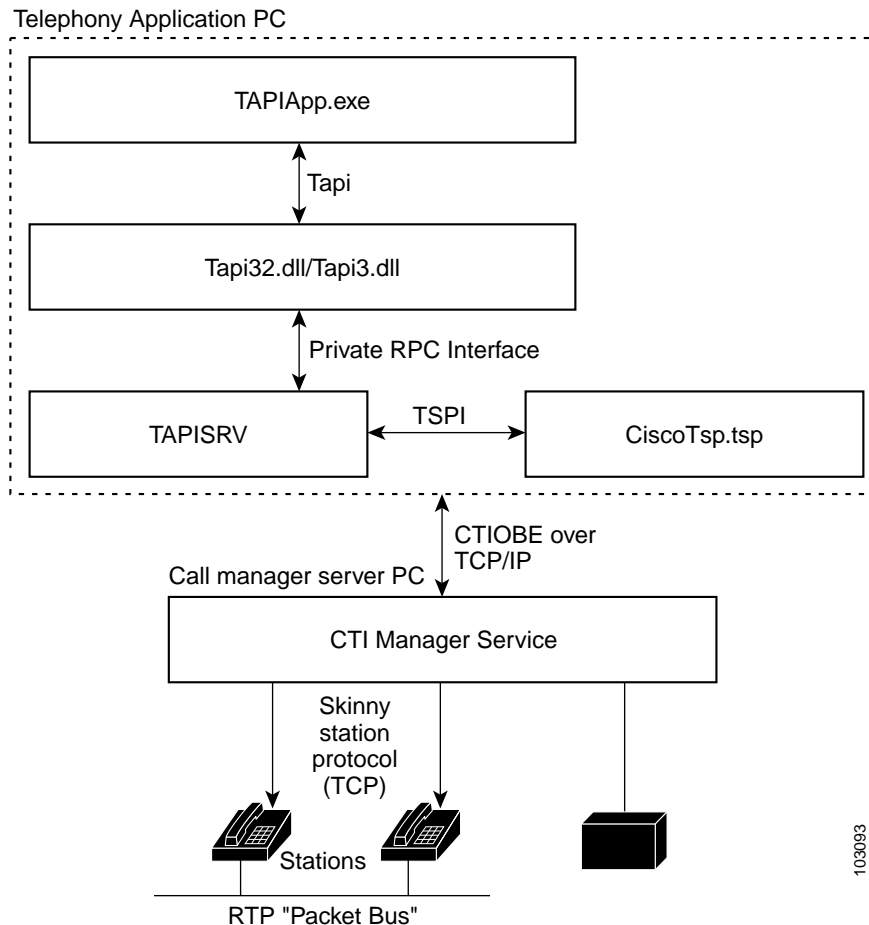
The TSP communicates with the underlying telephony hardware, such as a PBX, using a proprietary interface. The telephony hardware may be on the same PC or on a different PC. See [Figure 4](#).

Figure 4 *Telephony Application PC*



Cisco provides a TSP called CiscoTSP. CiscoTSP communicates with the Cisco CallManager using a proprietary interface known as CTIOBE. CiscoTSP allows customers to create customized IP telephony applications for the Cisco CallManager. See [Figure 5](#).

Figure 5 *Telephony Application PC*



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 CTI Manager Location is configured properly.
- _____ Check to see if the CTI Manager is up and running.
- _____ Check to see that the network connection is up between the TSP and CTI Manager machines.
- _____ Check to see if the Cisco CallManager DC Directory Service is up and running.
- _____ Check that the Username and Password that is configured in the TSP matches the Username and Password in the Cisco CallManager Directory.
- _____ Make sure that the Enable CTI Application Use flag has been checked in the Directory Administration on the Cisco CallManager for the TSP user.

- _____ Check that devices have been properly associated with the user in the Cisco CallManager Directory.
- _____ Check to see if the CTI Connection Limit has been reached.
- _____ For non-SoftPhone machines, ensure the Cisco Wave Drivers have been installed.
- _____ Use the tools in “[Troubleshooting Tools](#)” on page 13 to verify the TSP is working properly.
- _____ Ensure the CiscoTSP version is compatible with the version of Cisco CallManager. This can be verified by checking the Cisco CallManager Compatibility Matrix. Refer to the *Default 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

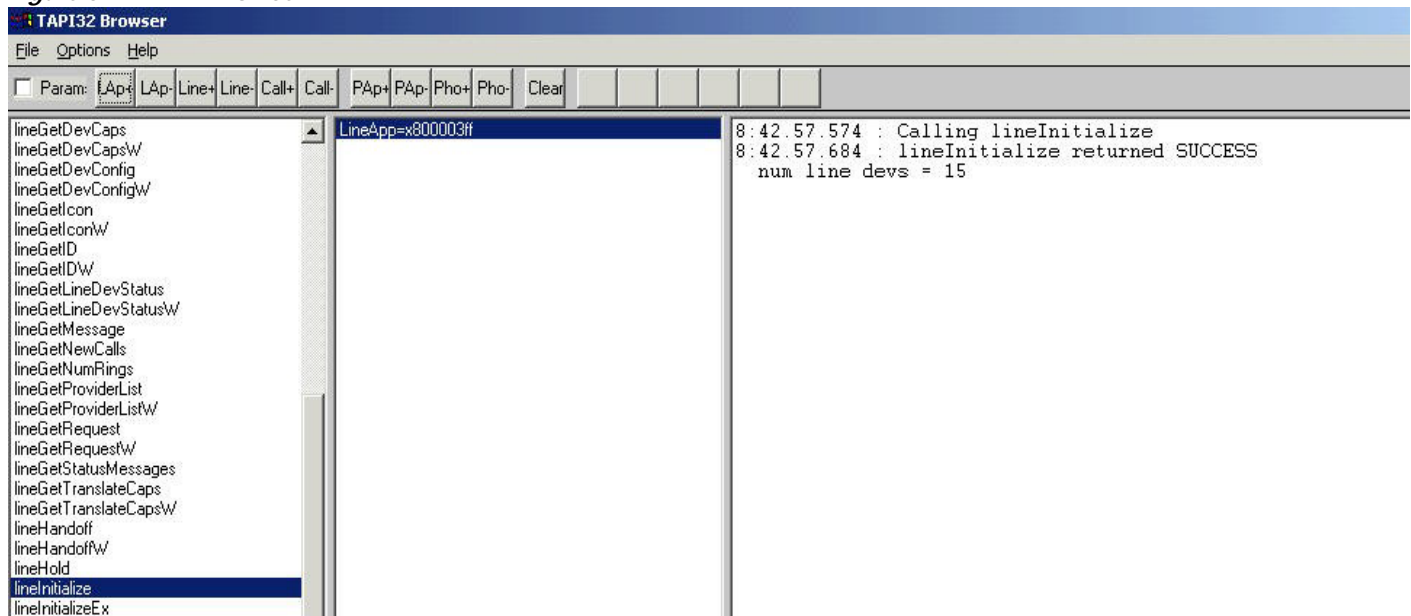
This section describes available troubleshooting tools.

TAPI Browser

The TAPI Browser does not come standard with the Windows operating system. It can be obtained from the Microsoft MSDN website. The TAPI Browser can be used to initialize TAPI. Its main use is by TAPI developers to test their TAPI implementation and to verify that the TSP is operational. See [Figure 6](#).

To do this, bring up the TAPI Browser (TB20.exe), then double-click **lineInitializeEx** in the leftmost window. If *num line devs* returns the expected number of the line device, the TSP is operational. `lineInitializeEx` initializes all lines of every configured TSP, not just the Cisco TSP. So, the number of *num line devices* is equal to all lines of every configured TSP.

Figure 6 TAPI Browser



TAPI Phone Dialer

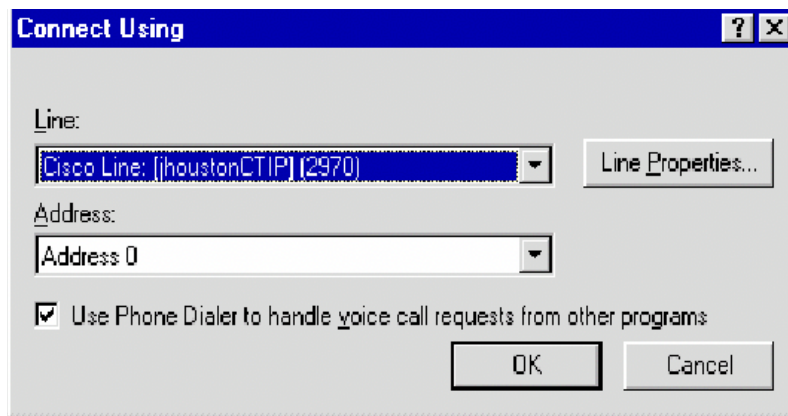
Another way to verify that the TSP is installed and configured properly is to use the TAPI Phone Dialer. The TAPI Phone Dialer is found on all of the Windows operating systems. You can do a Find to locate the file (dialer.exe).

The TAPI Phone Dialer is the same on Window 95/98/NT, but was changed in Windows 2000.

Windows 95/98/NT

For Windows 95/98/NT, start the **dialer.exe** program. If this is the first time Dialer has been used, the dialog box shown in [Figure 7](#) is displayed. If not, you need to click *Tools* and *Select Connect Using*.

Figure 7 Windows 95/98/NT TAPI Phone Dialer



In the *Connect Using* dialog box, select a *Cisco Line*. If there are no *Cisco Lines* in the line selection box, this indicates a connection problem between the Cisco TSP and the Cisco CallManager. Verify that all items in the “[Postinstallation Checklist](#)” on [page 12](#) are correct. For the *Address* selection box, select *Address 0*. Click **OK** to continue.

Now type a number to dial. See [Figure 8](#). If the phone call goes through, you know the Cisco TSP is operational.

Figure 8 Phone Dialer II



Windows 2000

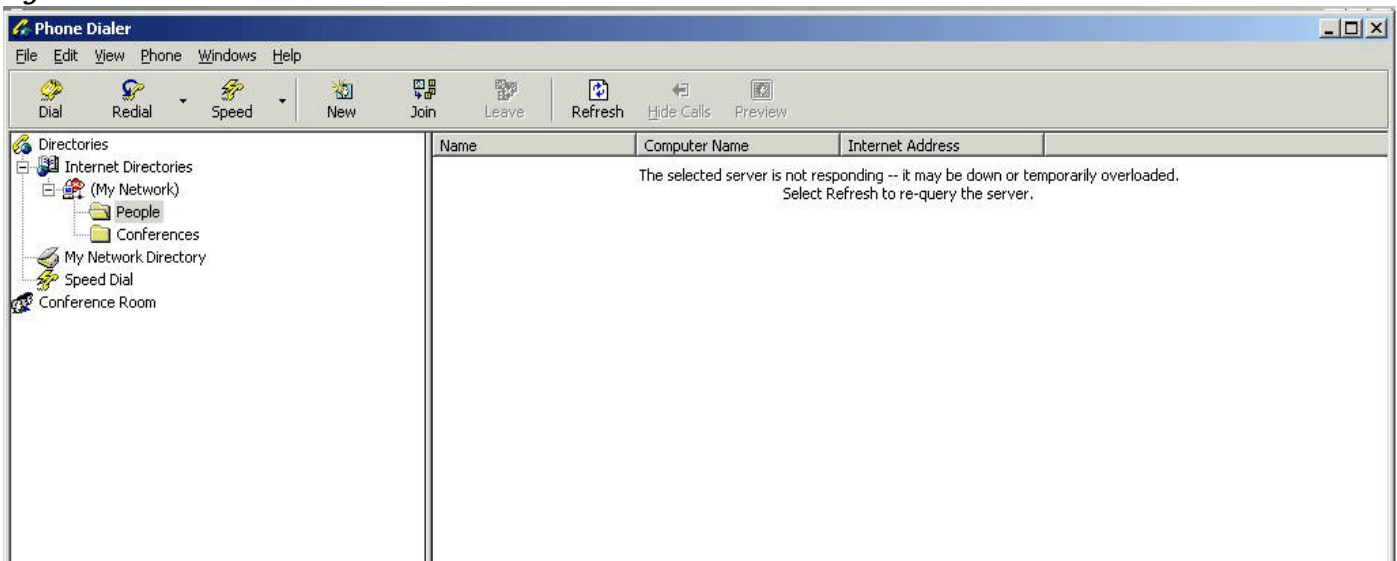
The Phone Dialer for Windows 2000 is shown in Figure 9. It works like the Phone Dialer in Windows 95/98/NT, but is a little fancier because it is a TAPI 3.0 application. From the *Edit* menu, select *Options* to select the line you want to control.



Note

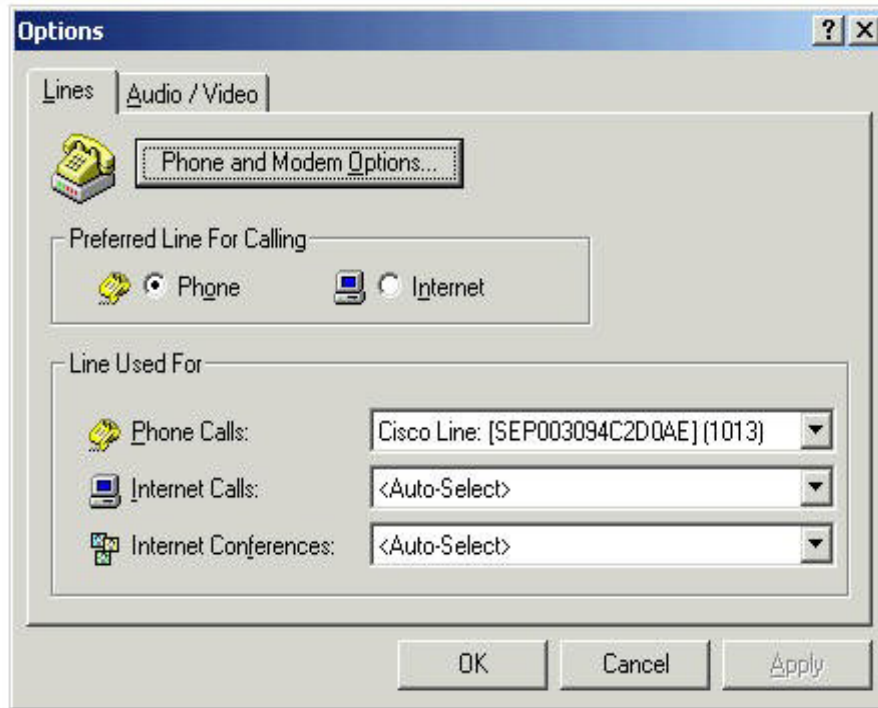
The Cisco TSP is written as a TAPI 2.1 provider; however, in Windows 2000 and higher, Phone Dialer uses the TAPI 3.0 API layer. While TAPI 3.0 applications are not officially supported by the Cisco TSP, TAPI 3.0 versions of Phone Dialer should still work as described below for troubleshooting purposes.

Figure 9 Windows 2000 TAPI Phone Dialer



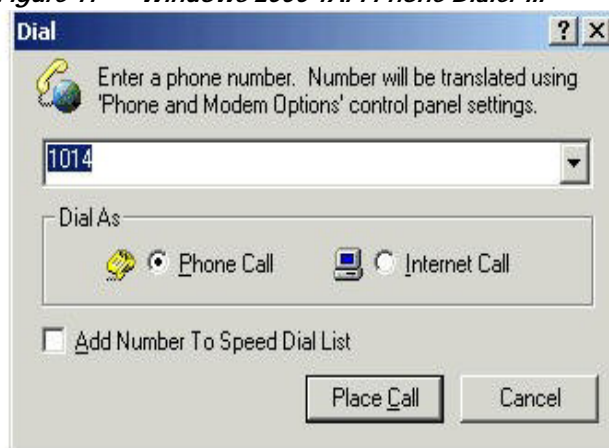
In the Lines tab, select *Phone* as the *Preferred Line For Calling*. See [Figure 10](#). In the *Phone Calls* selection box, select *Cisco Line*. If there are no Cisco Lines in the line selection box, this indicates a connection problem between the Cisco TSP and the Cisco CallManager. Verify that all items in the “[Postinstallation Checklist](#)” on [page 12](#) are correct. Click **OK** for the settings to take effect.

Figure 10 Windows 2000 TAPI Phone Dialer II



Next, click *Dial* and the dialog box in [Figure 11](#) appears. Type the phone number you want to call and click *Place Call*. If the call goes through, the Cisco TSP is working properly.

Figure 11 Windows 2000 TAPI Phone Dialer III



Error Reporting

Tracing for the TAPI Service Provider

Perform the following steps to initiate tracing for the TAPI Service Provider:

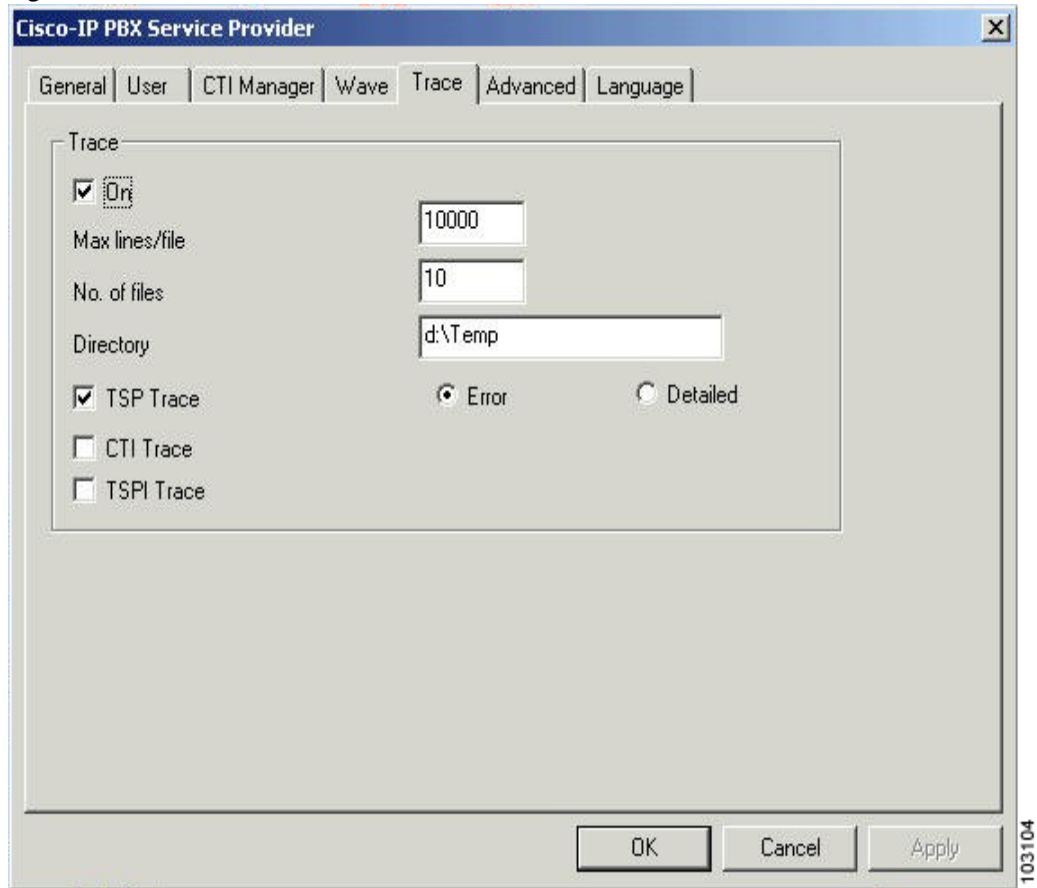
1. Go to *Start->Settings->Control Panel* and select **Phone and Modem Options**.
2. Go to *Advanced* tab. Select **CiscoTSP0XX** and click **Configure**.
3. Go to *Trace* tab. Select **Trace On** check box (see [Figure 12](#)) and select the following:



Note It is recommended that all trace types be enabled when troubleshooting.

- a. **TSP Trace** to trace the TSP internal messages. Select **Error** to just log errors in the TSP. Select **Detailed** to log internal messages for debugging purposes.
- b. **CTI Trace** to trace the messages sent between CTI and TSP
- c. **TSPI Trace** to trace the requests and events sent between TSP and TAPI

Figure 12 TAPI Service Provider Trace



The following definitions apply:

Term	Definition
Directory	The path for the trace log. For example, c:\Temp.
No. of Files	Set this to a value greater than or equal to 1 enables rolling log files. For example, a value of 10 will cause up to 10 log files to be used in a cyclic fashion.
Max lines/files	Specifies the maximum number of trace statements that will be written to each log file. For example, a value of 1000 will cause up to 1000 trace statements to be written to each log file.

Error and Status Codes

TAPI error codes are reported within the TSP traces, as configured in [“Tracing for the TAPI Service Provider” on page 17](#). Analysis of TSP trace logs is complex and error codes should be referred to the application developer support staff.

The following errors are normal operational warnings and can be safely ignored:

- CiscoTSP001.tsp| CSelsiusTSPLine::CallAsyncResponse() asyncResponseID=0x0000DCE1
ERROR Did not find call id 0x02044E99

This message is a normal message that results from an optimization in the TSP, whereby TSP execution is not blocked while waiting for a call tear-down confirmation from the CTI (this always succeeds). When the CTI does later return a message indicating successful tear-down of a closed call, the internal TSP call object may have already been destroyed, resulting in a “not found” call id.

- CiscoTSP001.tsp| CctiInterface::ProcessResponse() *ERROR* Response does not map to an existing request

Another similar optimization as above. When an application closes a call, the TSP immediately generates failure replies for any outstanding asynchronous requests on the call. When the actual CTI responses for these pending requests arrive at the TSP later, their corresponding request objects may have already been destroyed and this line gets logged to the trace file.

- CiscoTSP001.tsp| CctiInterface::RemoveFromSequenceNumberMap() *ERROR* Map entry does not exist for sequenceNumber(0x0000DD33)

Certain clean-up sequences in the TSP can result in attempting to remove a sequence map number from the internal request table more than once. This is a normal occurrence resulting from multi-threading issues and does not indicate a problem.

