OfficeServ Open TSP

User Manual

Version 1.0

March. 2004.

SAMSUNG ELECTRONICS

COPYRIGHT

TRADEMARKS

Product names mentioned in this document may be trademarks and/or registered trademarks of their respective companies.

This manual should be read before the installation and operation, and the operator should correctly install and operate the product by using this manual.

This manual may be changed for the system improvement, standardization and other technical reasons without prior notice.

For further information on the updated manual or have a question the content of manual, contact **Document Center** at the address below .

Address : Document Center 2nd Floor IT Center. Dong-Suwon P.O. Box 105, 416, Metan-3dong Paldal-gu, Suwon-si, Gyeonggi-do, Korea 442-600

e-mail : manual@samsung.com

Or contact **Call Center** at the telephone below if you have any questions or concerns regarding the operation of your system.

Phone : 81-1588-4141

http://www.samsungnetwork.com

INTRODUCTION

Purpose

The OpenTSP User Manual provides a brief description of the Samsung OfficeServ OpenTSP, the installation procedure, and the procedure on using the functions of the OpenTSP, for engineers who develop the TAPI service and telephony application programs.

Document Content and Organization

This manual includes six chapters and the ‘Acronyms’. The chapters are summarized as follows:

CHAPTER 1. Introduction

This chapter provides an overview description of the Microsoft TAPI system used on the OfficeServ telephone systems, a list of standard supported functions, and a list of Samsung Specific functions used only on the Samsung TSP driver.

CHAPTER 2. OpenTSP Driver Installation

This chapter describes items that must be checked before installing the OpenTSP driver and the procedure for installing the OpenTSP driver.

CHAPTER 3. OpenTSP Window Description

This chapter provides descriptions of the screens, toolbars, and buttons of the various tools created during the installing the OpenTSP driver.

CHAPTER 4. OpenTSP Driver Guide

This chapter provides the procedure on dialing, receiving, and disconnecting calls through the OpenTSP driver.

CHAPTER 5. TAPI Functions

This chapter describes the TAPI functions and expansion functions supported by the OpenTSP driver.

CHAPTER 6. Call Processing Flow

This chapter describes the life cycle of the TAPI, various call processing events of the OpenTSP driver, and the call processing procedure.

ABBREVIATION

The frequently used abbreviations and acronyms and their meanings in this guide are all collected and explained.

Conventions

The following special paragraphs are used in this document to point out information that must be read. This information may be set-off from the surrounding text, but is always preceded by a bold title in capital letters.

WARNING

Indicates a potentially hazardous situation which if not avoided, could result in death or serious injury.

CAUTION

Indicates a potentially hazardous situation which if not avoided, may result in minor or moderate injury. It may also be used to alert against unsafe practices.

CHECKPOINT

Provides the operator with checkpoints for stable system operation.

NOTE

Indicates additional information as a reference.

OPERATION PROCEDURES

Indicates the operation procedures that should be executed in order.

Console Screen Output

- The lined box with '`Courier New`' font will be used to distinguish between the main content and console output screen text.
- '**`Bold Courier New`**' font will indicate the value entered by the operator on the console screen

References

OfficeServ Operator Manual

The OfficeServ Operator Manual describes the main features, installation procedure, service settings, and user guide of the OfficeServ Operator, an application program for telephony communication.

OfficeServ Call Manual

The OfficeServ Call Manual describes the main features, installation procedure, service settings, and user guide of the OfficeServ Call, an application program for telephony communication.

Revision History

Edition No.	Date of Issue	Remark
1.0	3 2004.	1St Release

This page is intentionally left blank.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

1 Introduction to OpenTSP

The OpenTSP Telephony Service Provider Driver 3.x (referred to as 'OpenTSP' hereinafter) interfaces with the Samsung OfficeServ telephone system through the TCP/IP system, based on the Microsoft TAPI 2.x/TAPI 3.x, specifications and enables call control and call processing of the Telephony Application Programming Interface(TAPI) service through the TSPI.
The OpenTSP driver is installed on a PC using the Windows O/S.

The Microsoft TAPI consists of the three modules shown below:

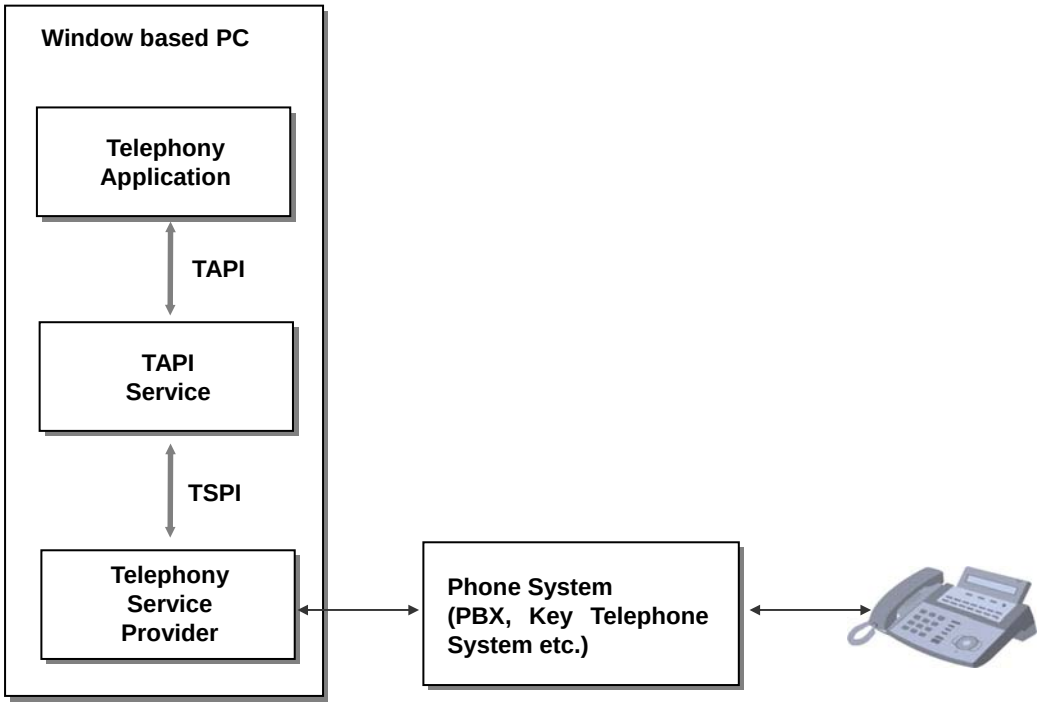


Figure 1.1 TAPI Configuration Diagram

Modules in Figure 1.1 are described below:

Telephony Application

Supplied by the application vendor, the Telephony Application provides features such as call processing to the users through TAPI, an API provided by the Microsoft TAPI Service.
Telephony applications include the ‘dialer’ program, embedded in the Windows OS, and Outlook, the Contact Manager program of Microsoft.

TAPI Service

As a basic module of the Microsoft Windows OS, the TAPI Service uses the Telephony Service Provider installed on a PC upon the request of the application program.

Telephony Service Provider (TSP)

Provided by the switch vendor, the TSP is a service provider that communicates with the Microsoft TAPI. The TSP is executed when the application program requests the TAPI feature.

2 OpenTSP Features

The Samsung OpenTSP driver is a Microsoft Windows compatible TSP, the Telephony Application program connects to the TAPI Service through a Computer telephony interface to use the features of the Samsung key telephone system.
The OpenTSP referrers to the appropriate Microsoft TAPI specifications, and supports the features introduced in Section‘5.2. (TAPI Functions) of this manual.’
The following system specific features that are supplied only by the OpenTSP driver are described in detail in ‘Section 5.3. OpenTSP Driver Extended Function Feature List’.

- Station Lock
- Vacant Station Message
- Follow me
- Make new trunk call
- Page
- System hold retrieval
- Clear Message Waiting
- Clear Call back
- OHVA
- Silent Monitoring
- Mute on/off
- Line Reset Function

CHAPTER 2

OpenTSP Driver Installation

This chapter describes the environment and procedure required for the installation of the OpenTSP driver. For proper installation and operation of the OpenTSP driver, the installation environment and conditions should be checked before installation.
Refer to the table below, in which the installation procedure is summarized, when installing the OpenTSP driver.

Table 2.1 OpenTSP Driver Installation Procedure

Step	Procedure	Description
1	Installation Environment and Conditions	Check the following environment and condition before installing the OpenTSP driver. - Check the H/W and S/W environments. - Check for the OpenTSP driver licence key. - Check if the OfficeServ Link program has been installed. - Previous versions of Samsung TAPI drivers should not be on the system.
2	OpenTSP Driver Installation Procedure	Install the OpenTSP driver according to the installation procedure. Read the Cautions and Notes carefully to prevent error during installation.
3	Installation Data Verification	After installing the OpenTSP driver, check if the installation is successful by verifying the driver file and the registration status.

1 Installation Environment and Conditions

The OpenTSP driver may be installed and executed for call processing on various versions of Microsoft Windows. This section describes the environment and conditions that are required for proper installation of the OpenTSP driver.

1.1 Installation Environment

Check the installation environment below before installing the OpenTSP driver.

Table 2.2 OpenTSP Driver Installation Environment

Type	Category	Requirement
Hardware	Compatible Switch	The OpenTSP may only be used in the Samsung key telephone system, which supports the TAPI 2.x interface.
	Switch Interface	The OpenTSP may only use the switch service through a separate S/W called OfficeServ Link.
	Network Interface	A network card supporting TCP/IP protocol should be installed on the PC.
Software	TAPI version	TAPI 2.x or higher
	Operating system	- Windows 98: Since the TAPI 2.x is included in the Second Edition of Windows 98, only the Windows 98SE version can be used. - Windows ME: Check the TAPI version and upgrade through the Service Pack if the version is lower than 2.x. - Windows 2000, and Windows XP: includes TAPI 3.x. - Windows NT: The TAPI 2.x is included in 4.0 or higher versions of the Service Pack. Check the service pack version and upgrade the service pack version to 4.0 or higher. - Windows CE: not supported -‘Unix and Apple systems, or systems using Terminal Services (including Citrix) are not supported

1.2 Installation Conditions

Check the items below before installing the OpenTSP driver on the system.

Valid Licence

The licence key is required for installing and using the OpenTSP driver.
Refer to your Samsung Dealer for details of how to obtain a licence.

OfficeServ Link Program

The OfficeServ Link program must be installed to use the CTI features of the Samsung telephone system. All CTI application programs are connected to the switch through the OfficeServ Link program.

OfficeServ Link Program

The OfficeServ Link program is software that enables multiple CTI application programs to connect to the switch, and controls the message flow between the application programs and the switch. Refer to the OfficeServ Link Manual for details such as installation and operation of the OfficeServ Link program.

Samsung TSP Driver

Delete previous versions of the Samsung TSP driver, if any, before installing the new driver.

Previous versions of the Samsung TSP driver may be installed in the following cases

- If the Samsung TAPI 2.x was installed on a PC using the Windows NT or 2000 Server OS
- If the computer where the Samsung TAPI 2.x driver was installed is set as the Telephony Client
- If iDCS Call Version 5.1 or lower is installed

The procedure for checking the information on the TSP driver installed on the computer is as follows:

1) Select 'Start→Settings→Control Panel' on the computer.

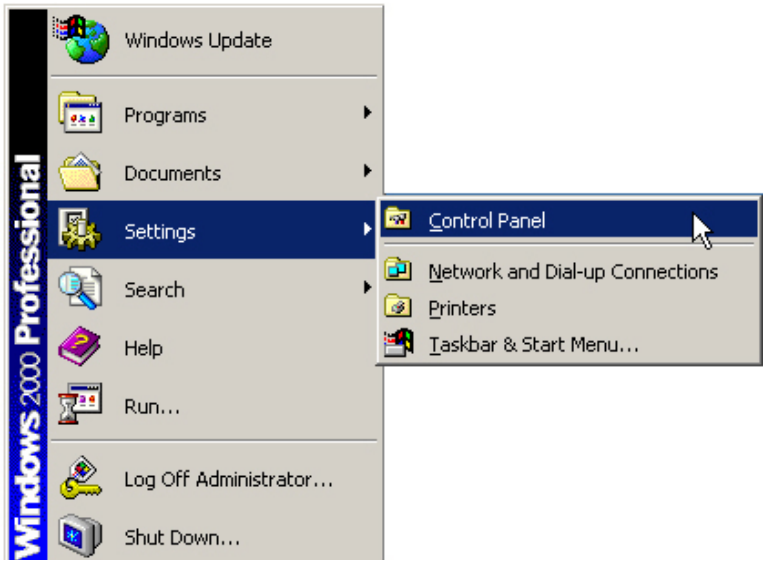


Figure 2.1 Starting the Control Panel

2) Double click 'Phone and Modem Options' from the 'Control Panel' shown below:

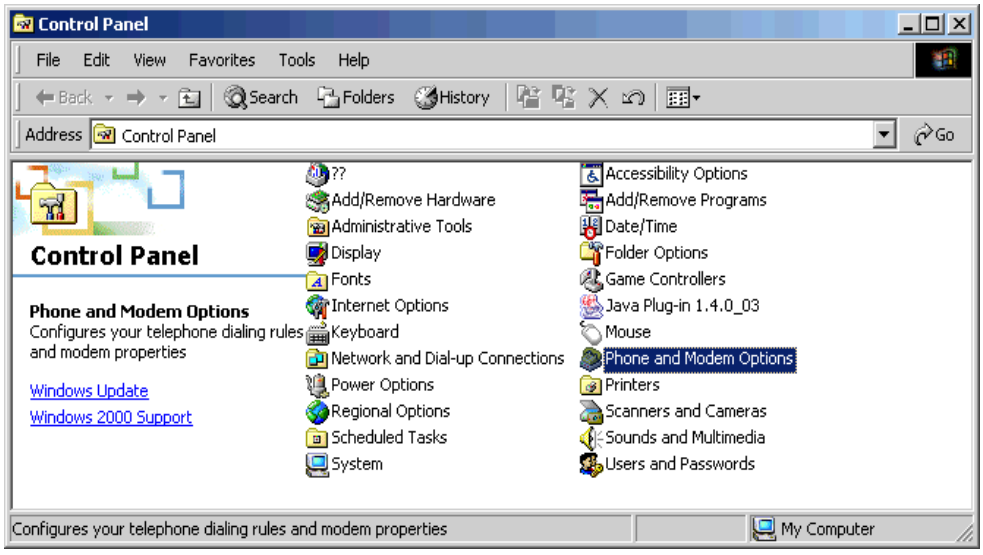


Figure 2.2 Selecting Phone and Modem Options

3) Select [Edit (E)] from the ‘Phone and Modem Options’ window.

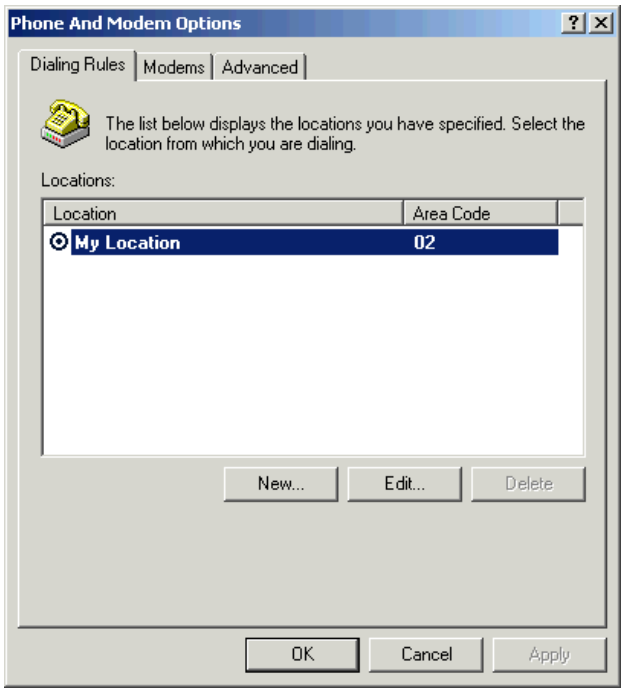


Figure 2.3 Phone and Modem Options Window

4) Enter the fields of the ‘Edit Location’ Window by referring to the figure below and click the [OK] button.

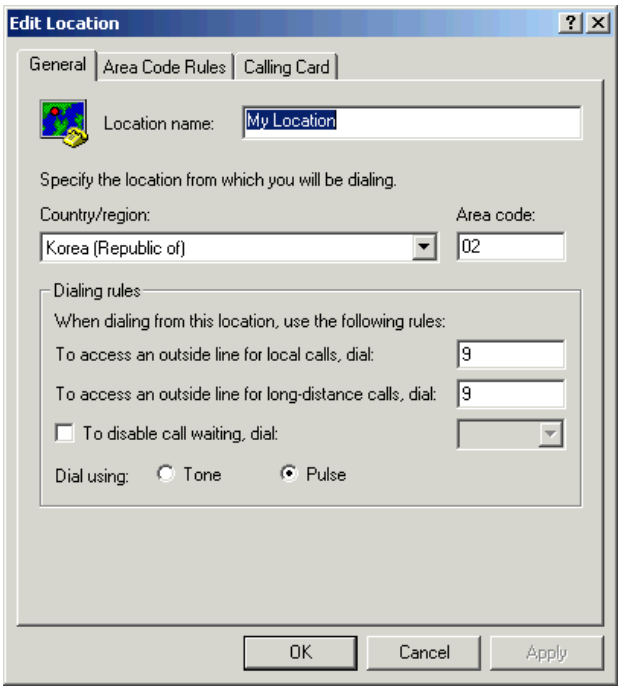


Figure 2.4 Edit Location Window

Select ‘Country/Region’ and enter your area code. Do not enter ‘0’ of the area code. For example, if the area code is 031, enter ‘31’.
The ‘Dialing Rules’ option is used for making external calls through TAPI.. Enter the number to be used for making external calls. In Australia the number assigned for outside calls is usually ‘0’.

NOTE

Dialing Rules

Since phones in offices usually connect to the trunk line through a private switch, consult the telephony manager of your company for information on the number assigned for outside calls.

5) Select the ‘Advanced’ tab from the ‘Phone and Modem Options’ screen to display the list of drivers (telephony service providers) installed on the system.

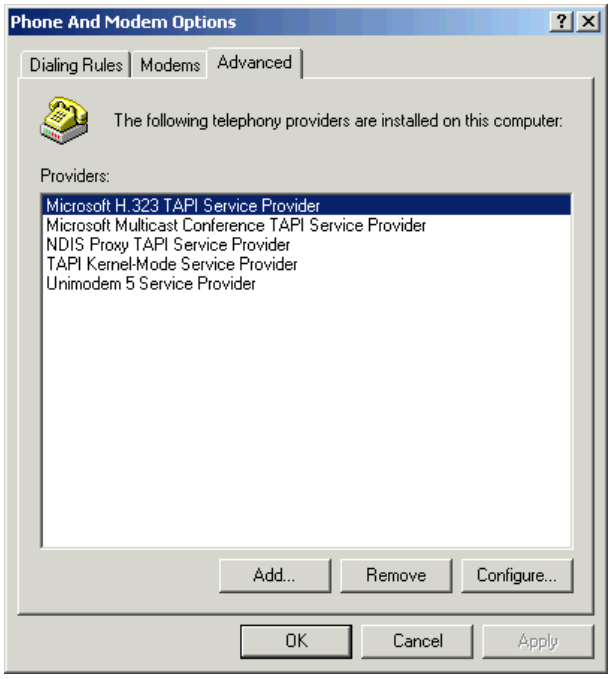


Figure 2.5 Advanced Tab of Phone and Modem Options

The TAPI compatible driver can be installed separately on each computer, and a newly added TAPI driver is displayed on the ‘Advanced’ tab of the ‘Phone and Modem Options’.
The TAPI driver is registered as ‘Samsung SCTSP32 TAPI2.x Compatible Telephony Service Provider’ or ‘Samsung DCSTSP Telephony Service Provider’.

1.3 Checking the Telephony Service

The procedure for checking if the Telephony service is normally operating is as follows:

1) Select ‘Start→Settings→Control Panel’ from the computer.

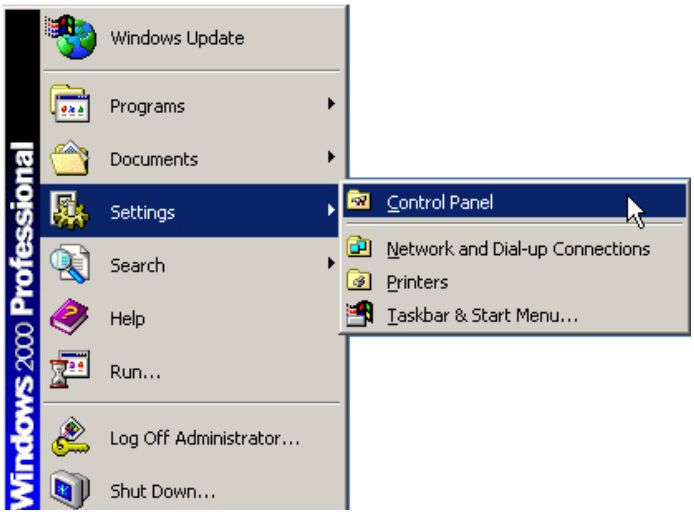


Figure 2.6 Executing the Control Panel

2) Double click the ‘Administrative Tools’ from the control panel below.

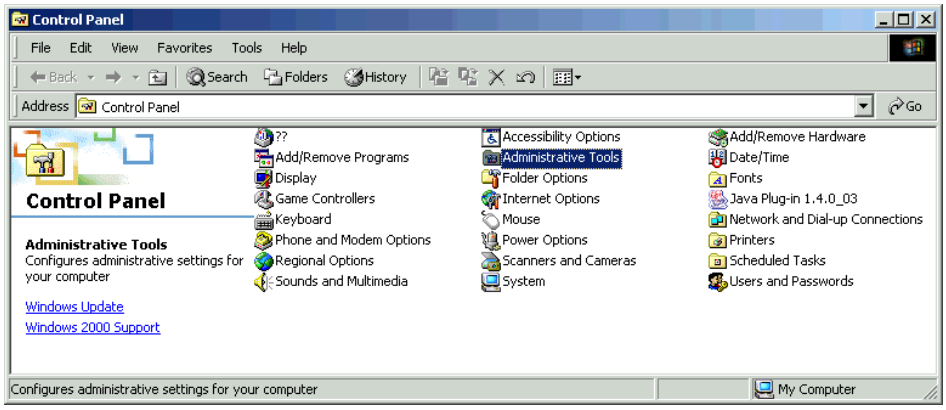


Figure 2.7 Selecting Administrative Tools

3) Select ‘Service’ from the ‘Administrative Tools’ window.

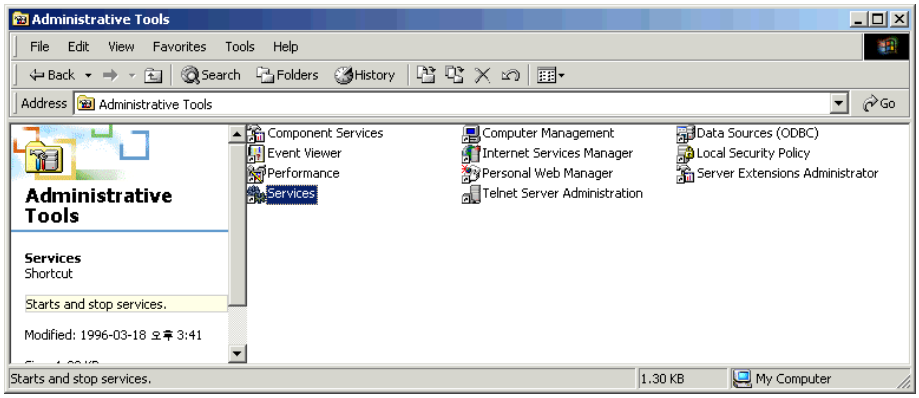


Figure 2.8 Selecting Service

4) Check the ‘Telephony’ service (TAPI service) status from the ‘Service’ window below. If the ‘Telephony’ service is displayed as ‘started’, as shown below, the TAPI service is operating.

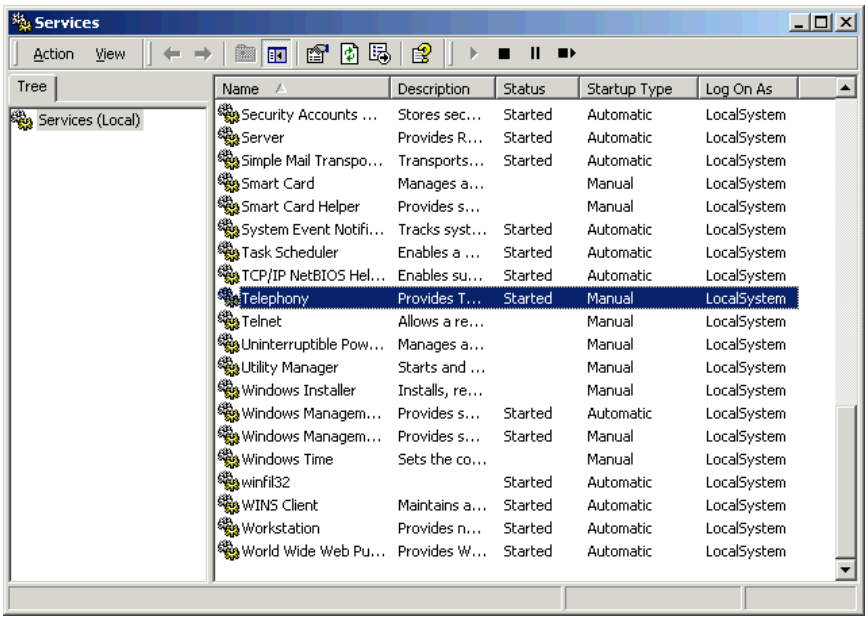


Figure 2.9 Service Window

NOTE

Terminating/Restarting the Telephony Service

Users can terminate or restart the telephony service, if necessary, through the above window.

5) The telephony service of the Windows OS is related to the following services. Thus, the services below should be checked for normal operation.

- Remote Access Auto Connection Manager
- Remote Access Connection Manager

These two Services should be set to Disable or Manual (and stopped) for normal operation.

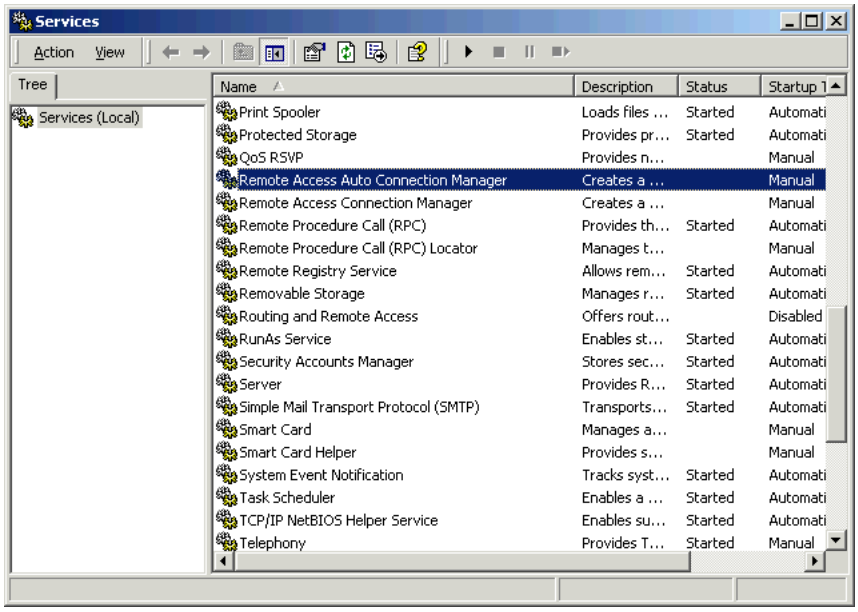


Figure 2.10 Telephony Service

CAUTION

Cases where the OpenTSP Driver is not Properly Loaded or Unloaded

If the two services above are operated 'Manually' the two services will act as a single module of the telephony service. If the two services were abnormally set during the system setup procedure, the two services and the telephony service may not operate normally and the OpenTSP driver may not be loaded or unloaded properly, disabling the use of related application programs. To avoid such incidents, it is recommended to set the start type as 'Disabled' to disable unnecessary services. If the two services are marked as 'Started' and the OpenTSP does not operate properly, change the 'start type' to 'Disabled' and reboot the system.

2 OpenTSP Driver Installation Procedure

The procedure for installing the OpenTSP driver is as follows:

1) Double click the OpenTSP installation file (Setup.exe) on the CD-ROM.

2) Click the [Next>] button on the screen below.

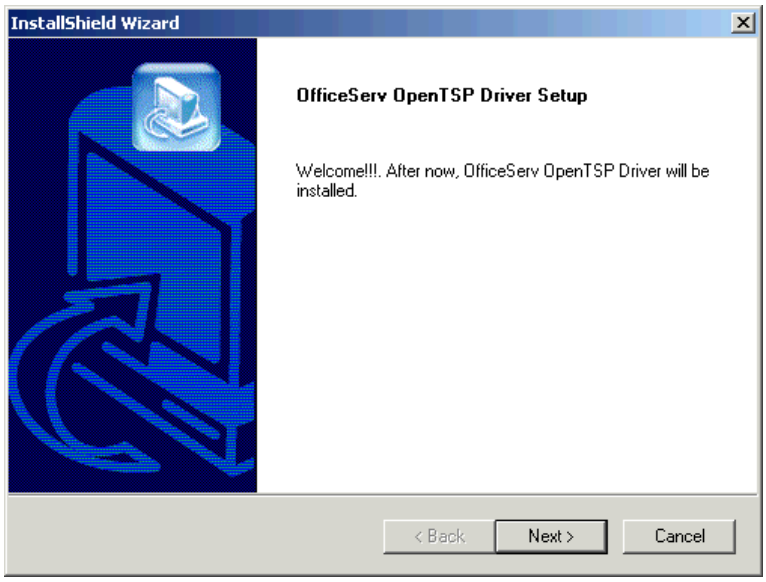


Figure 2.11 Installation Window

3) Read the Licence Agreement below and select [Yes] to approve. Select [No] to abort the installation program.

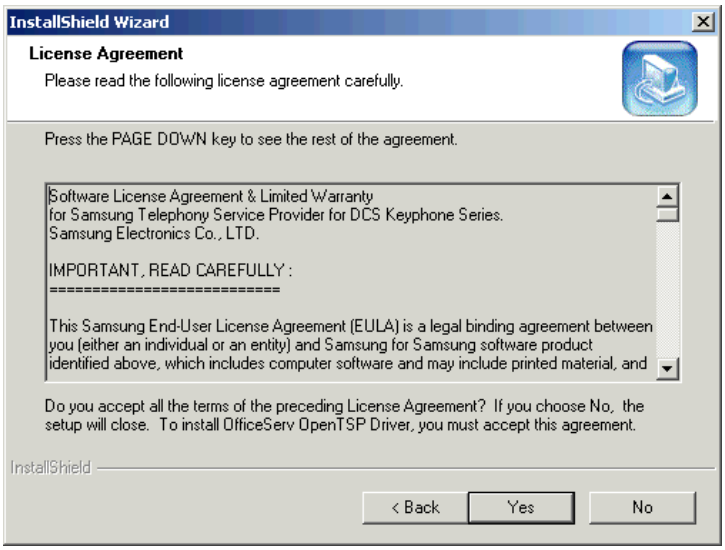


Figure 2.12 Licence Agreement

4) The ‘Choose Destination Location’ window appears as shown below. Click the [Next>] button to use the default path (C:\Program Files\Samsung Telephony Service Provider) or click the [Browse] button to change the installation folder.

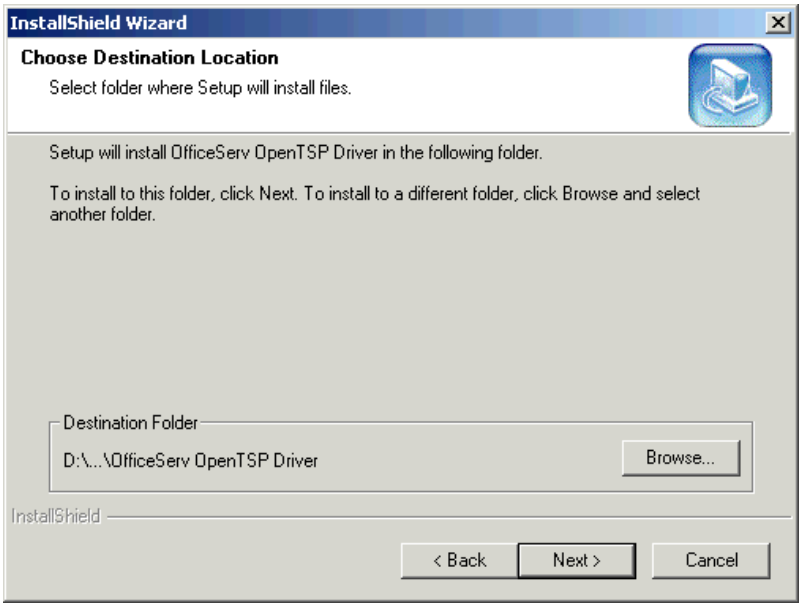


Figure 2.13 Selecting Installation Folder

NOTE

OpenTSP Driver Installation Folder

The OpenTSP installation program installs two types of programs on the user’s computer.
The SCTSP32.TSP file (basic Telephony Service Provider file) is copied to the C:\WINNT\system32 folder and is registered to TAPI.
Utility programs required for installing and operating the OpenTSP driver are copied to the C:\Program Files\Samsung Telephony Service Provider folder.
Thus, the folder selected during the installation procedure above (C:\Program Files\Samsung Telephony Service Provider) is the location to where the utility programs are copied.

5) Select ‘Typical’ from the ‘Setup Type’ window below and click the [Next>] button.

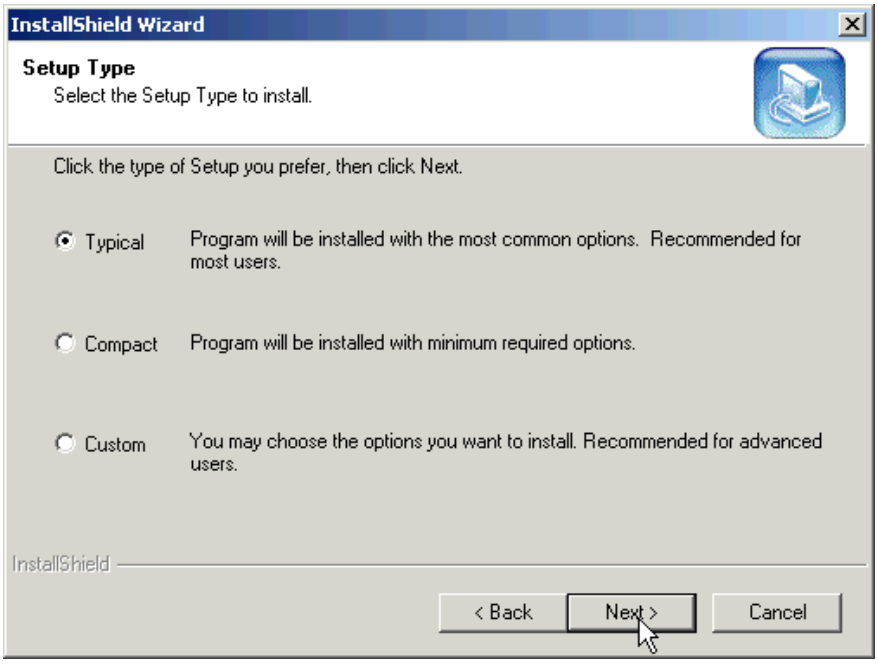


Figure 2.14 Selecting Installation Type

6) The ‘Select Program Folder’ window below appears. Click the [Next>] button to use the default name (Samsung Telephony Service Provider). Enter a new name into the field to change the folder name.

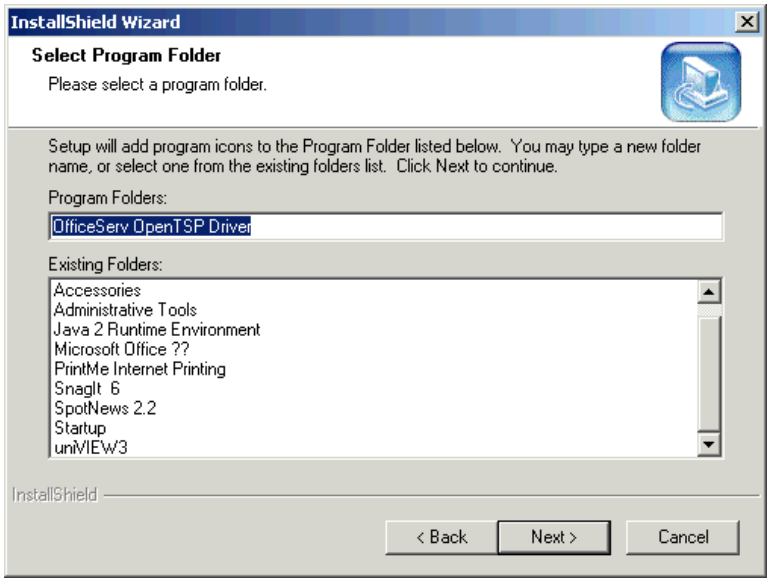


Figure 2.15 Selecting Installation Folder

7) Enter the entry items of the ‘Phone and Modem Options’ window below.

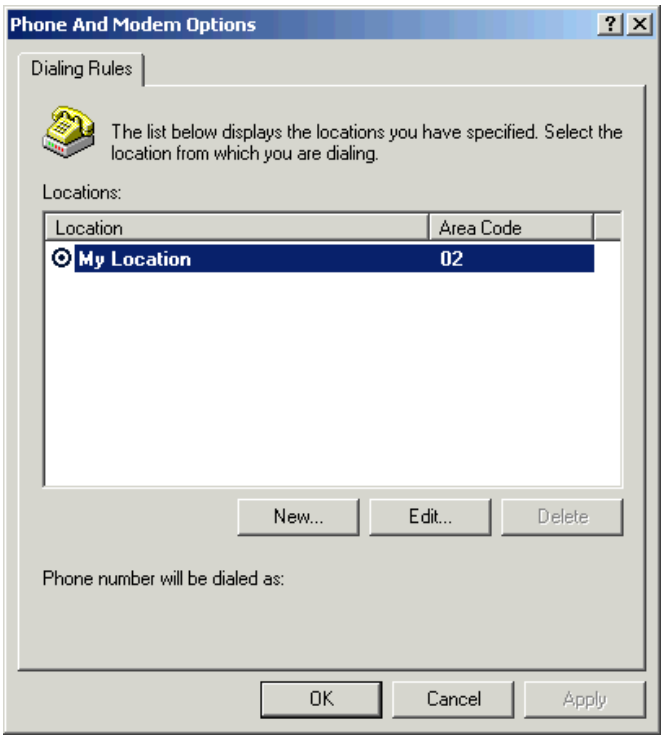


Figure 2.16 Phone and Modem Options

NOTE

‘Dialing Rules’ Setup
Refer to the steps from 1) to 4) of the ‘Samsung TAPI Driver‘‘ in ‘1.2 Installation Conditions’ for setting the ‘Dialing Rules’.

8) Enter the items of the ‘Communication Parameters’ window below and click the [OK] button.

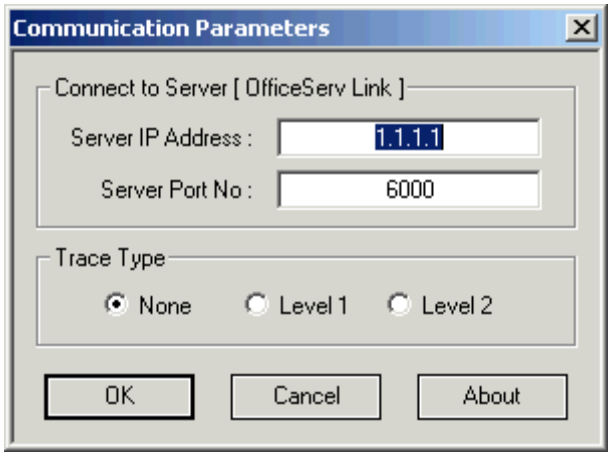


Figure 2.17 Communication Parameters

NOTE

Trace Type

The Trace Type setup affects the performance of the OpenTSP driver and should be set to ‘None’ under normal circumstances. Change the setting to ‘Level 1’ or ‘Level 2’ only when instructed by a trained support engineer.

NOTE

If the OpenTSP is installed when OfficeServ Link is not operating

Though the OpenTSP driver can be installed while the OfficeServ Link is not operating, the CTI application program automatically attempts connection to the corresponding port after the installation. Thus, the ‘Communication Parameter’ should be set to enable connection to the OfficeServ link before starting the CTI application program.

9) Upon successful installation of the OpenTSP driver, the ‘OpenTSP Setup Complete’ window appears. Click the [Finish] button.

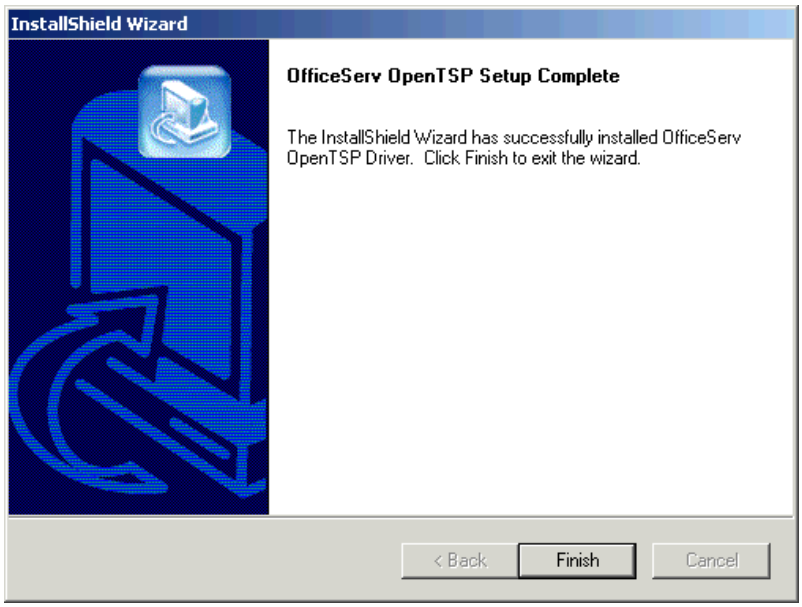


Figure 2.18 Installation Complete

3 Checking Installation Data

3.1 Checking the OpenTSP Driver Files

During the installation of the OpenTSP driver, the OpenTSP driver and utilities should have been copied to the folders below:

Location of the OpenTSP driver file

The OpenTSP driver is copied to different folders depending on the OS.

- Windows NT/Windows 2000 : Winnt\System32\sctsp32.tsp
- Windows XP : Windows\system32\sctsp32.tsp
- Windows 98/Me : windows\system\sctsp32.tsp

Location of the OpenTSP driver utility files

The OpenTSP utility files should have been copied to the folder below if the default location in ‘Figure 2.15 Selecting Installation Folder’ was not changed.

- Program Files\Samsung Electronics\Samsung Telephony Service Provider

Utilities for OpenTSP driver

The utilities required for installing/operating the OpenTSP driver are as follows:

- Scavenger.exe : Program for removing Mismatch Call Handle
- TAPISampler.exe : Test program for simple dialing/receiving/disconnecting calls and for tracking call status/data
- OpenTSP Config Tool : Communication environment setup program for OpenTSP driver

As shown below, the utility programs are located at 'Start→Programs→OfficeServ OpenTSP Driver' for convenient use.

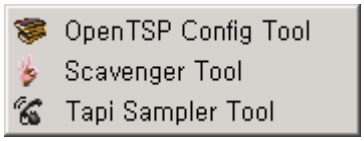


Figure 2.19 OpenTSP Utility Program

3.2 Checking the OpenTSP Driver Registration

The registration of the OpenTSP driver file, which is registered as the TAPI driver of the Microsoft Windows OS, can be verified as follows. :

Check for the 'Samsung SCTSP32 TAPI2.x Compatible Telephony Service Provider' on the 'Advanced' tab of the 'Start→Control Panel→Phone and Modem Options'.

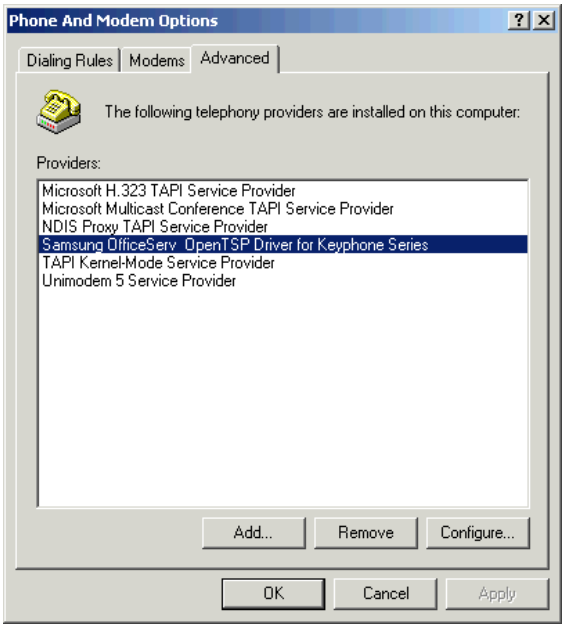


Figure 2.20 Advanced Tab of Phone and Modem Options

3.3 Changing the OpenTSP Driver Environment Settings

The user may change the environment settings of the OpenTSP driver. The driver environment can be changed either through the 'Phone and Modem Options' screen or through the OpenTSP Config Tool program.

Changing from the 'Phone and Modem Options' screen

1) Click the [Configure] button from the 'Start→Control Panel→Phone and Modem Options→Advanced' screen.

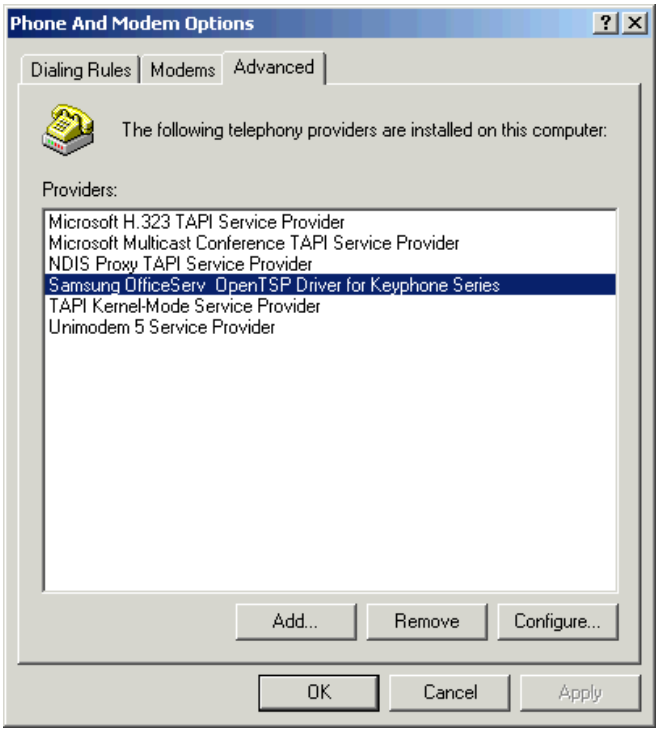


Figure 2.21 Selecting Configure

2) On the OpenTSP environment configuration screen below, change the settings and click the [OK] button.

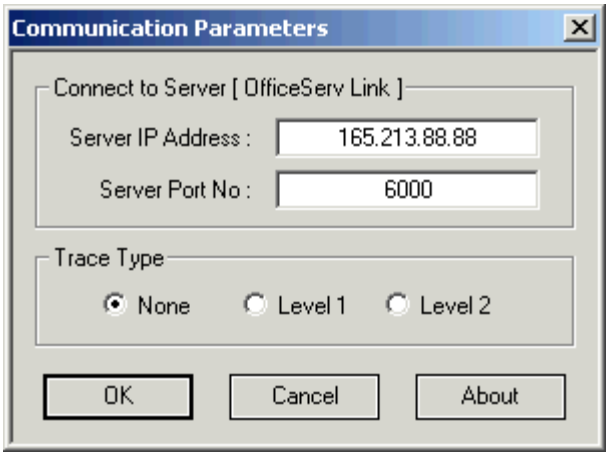


Figure 2.22 Communication Parameters

3) The message below appears to confirm the changes.

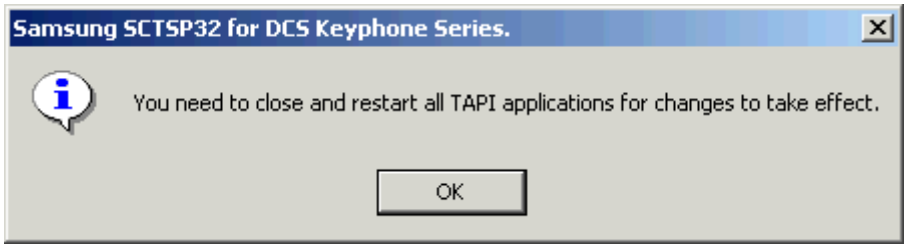


Figure 2.23 Confirming Changes in Environment Settings

4) The OpenTSP driver must be restarted to apply the changes. Thus, close all CTI application programs and restart the OpenTSP driver.

Using the OpenTSP Config Tool program

The OpenTSP Config Tool program, which allows the user to change the configuration of the OpenTSP driver, is installed in the OpenTSP driver installation folder. Through this program, users can easily check and change the settings.

The procedure for changing the OpenTSP driver environment through the OpenTSP Config Tool program is as follows:

1) Execute the 'OpenTSP Config Tool' under the 'Start→Programs→OfficeServ OpenTSP Driver'. The screen below is displayed.

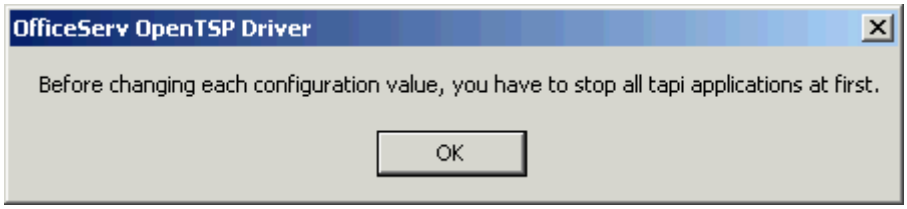


Figure 2.24 Closing TAPI Compatible Program

2) The message above informs the user that all TAPI compatible programs need to be closed before changing the environment settings. Close all TAPI compatible programs that are currently operating and click the [OK] button.

3) Enter the fields in the window shown below and click the [OK] button.

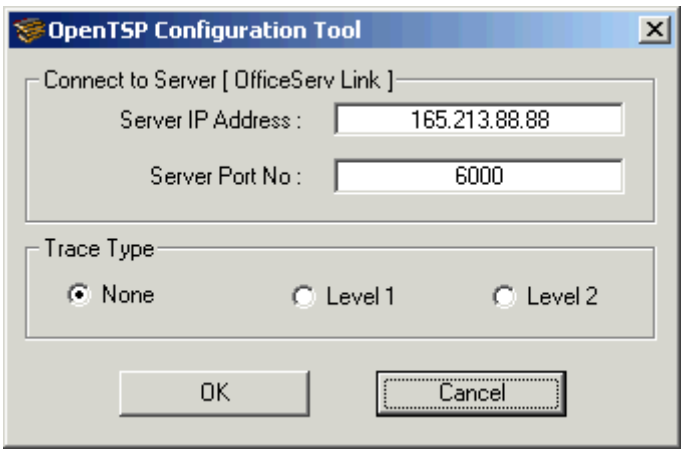


Figure 2.25 OpenTSP ConfigTool

NOTE

OpenTSP Configuration Tool

Refer to item 9) of the '2 OpenTSP Driver Installation Procedure' for detailed descriptions of each field.

4 Removing the OpenTSP Driver

Remove the OpenTSP driver installed on the system when the driver is no longer needed or when removing a previous version to install a new version of the driver.

> CAUTION
>
> **Closing all TAPI compatible programs**
>
> All TAPI compatible application programs that are currently running must be closed before removing the OpenTSP driver. If the OpenTSP driver is being operated by a TAPI compatible application program, error may occur during the uninstallation process.

The procedure for removing the OpenTSP driver is as follows:

1) Select 'Start→Settings→Control Panel→Add/Delete Program' to display the screen below. Then, select the 'Samsung Telephony Service Provider' and click the [Change/Remove] button.

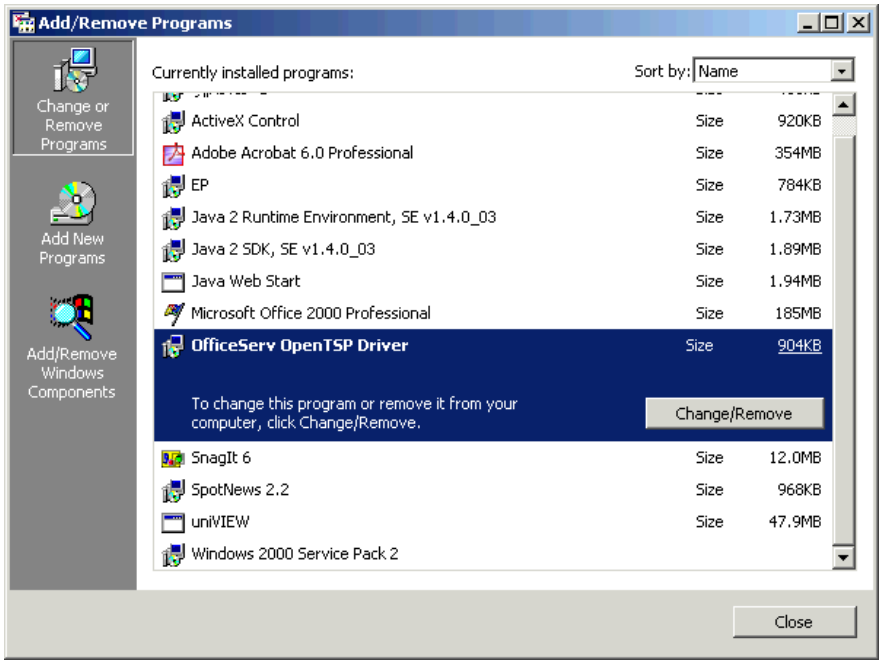


Figure 2.26 Add/Delete Program

2) Among the radio buttons, select the 'Remove' item and click the [Next>] button.

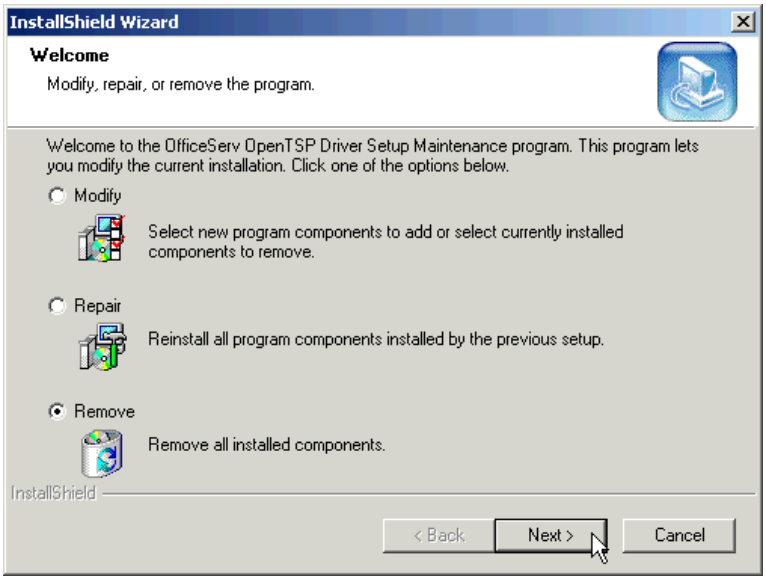


Figure 2.27 Delete Window

3) Click [OK] on the below message window confirming the removal of the OpenTSP driver files.

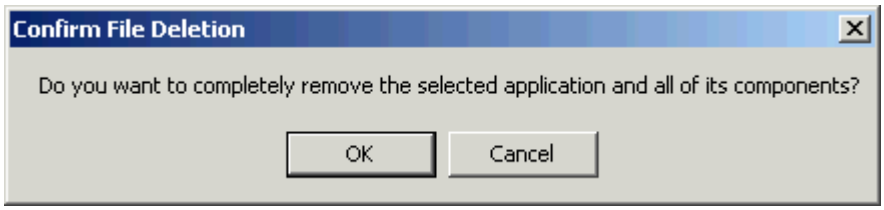


Figure 2.28 Confirm Deletion

4) The window below appears to confirm the deletion of the C:\WINNT\System32 \SCT32.TSP file. Click the [Retry] button to delete this file.

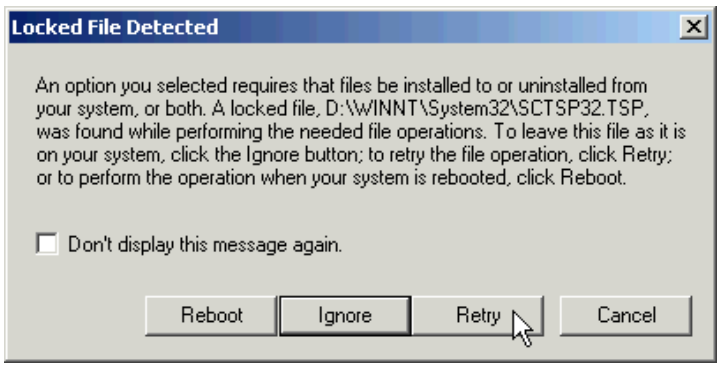


Figure 2.29 Confirm File Deletion

5) Files related to the OpenTSP driver are removed from the system. Click the [Finish] button on the screen below.

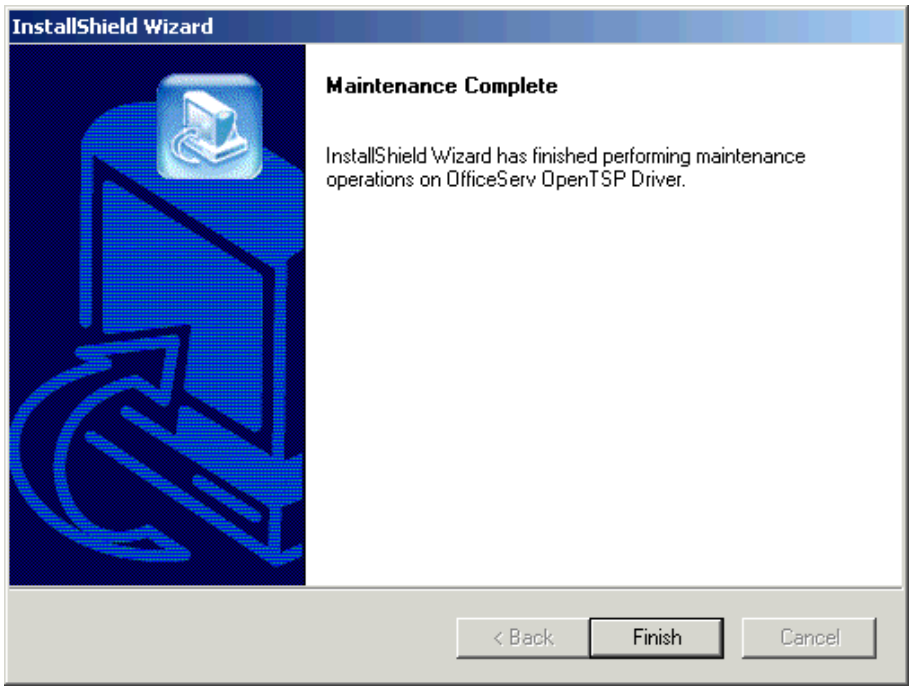


Figure 2.30 Deletion Complete

CHAPTER 3

OpenTSP Window Description

This chapter provides description on the screens, toolbars, and buttons of the various tools offered by the OpenTSP.
Tools provided by the OpenTSP are displayed as five submenus under the 'Programs →OfficeServ OpenTSP Driver'.

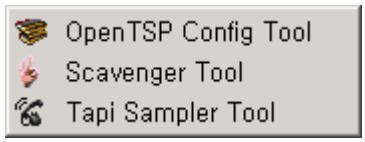


Figure 3.1 OpenTSP Submenu

Tools provided by the OpenTSP are as follows:

- OpenTSP Config Tool
- Scavenger Tool
- TAPI Sampler

1 OpenTSP Config Tool

The OpenTSP Config Tool allows the user to set the network information, Trace Type, and the Licence Key of the OfficeServ Link, to which the TSP connects.

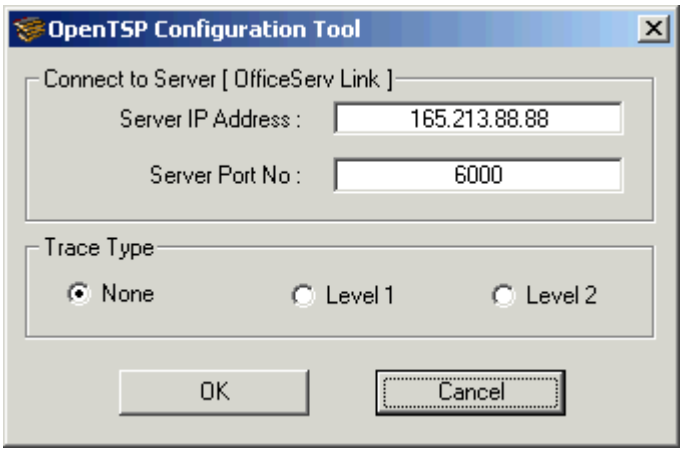


Figure 3.2 OpenTSP Config Tool Screen

- Server IP Address: Enter the IP address of the computer where the OfficeServ Link is installed. The OfficeServ Link program may or may not be installed and operated on the same computer where the OpenTSP driver is installed.
- Server Port No: This is the number of the port where the OfficeServ Link program is waiting for connection. The default number is 6000. This port number should be set as the same port number set at the OfficeServ Link program.
- Trace Type: The OpenTSP driver displays its operation data through the Tool (DBGView.exe). Set the details of the operation data to be displayed.
  - None : No display(default)
  - Level 1: Displays only basic information.
  - Level 2: Displays detail information.

2 Scavenger Tool

The Scavenger Tool allows the user to initialize the TSP driver without restarting the driver during operation.

This can be used to remove the call details from one or more devices if required. This tool should be used with care and only under instruction from a trained engineer.

Figure 3.10 Scavenger Tool

Item	Description
Setup	Displays the Scavenger option setup screen. On the option setup screen, select the phone line from which the Call details should be removed and specify the operation time.
Close	Sends the Scavenger Tool to the system tray.
Exit	Closes the Scavenger Tool.

3 TAPI Sampler

The TAPI Sampler is a utility program that dials, receives, or disconnects calls through the OpenTSP driver or displays various TAPI events received by the application programs through the TAPI service.

Start the TAPI Sampler while the OfficeServ Link program is normally operating to display the screen below.

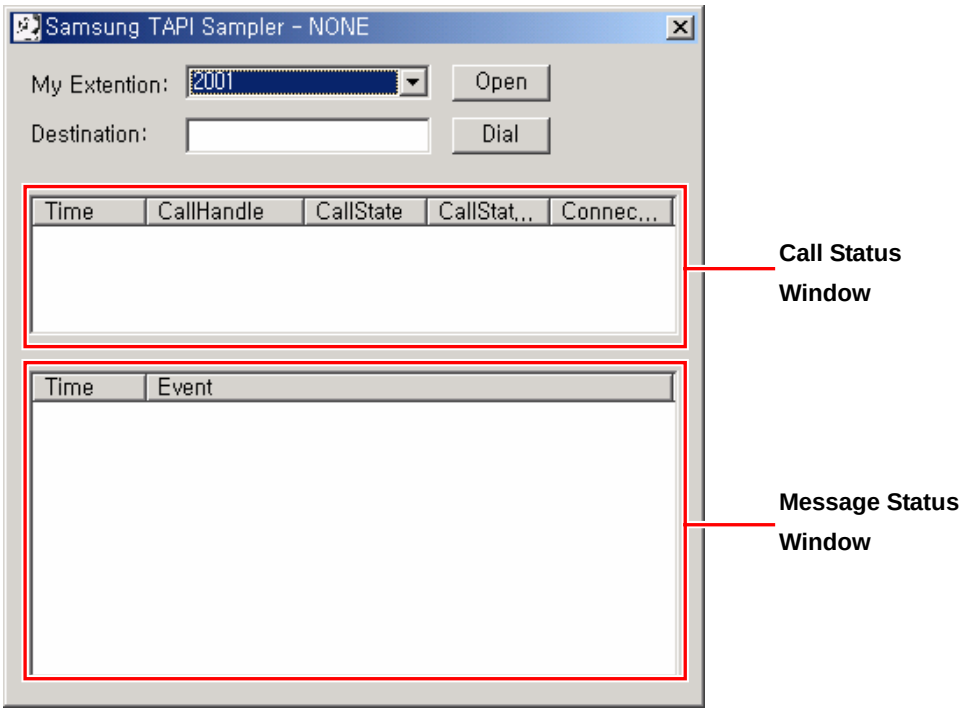


Figure 3.15 TAPI Sampler Screen

As shown above, the TAPI Sampler is simply configured.

- My Extension: Select the number of the device to be used by the TAPI Sampler.
- Destination: Enter the destination number for an intercom/external call.
- Call Status Window: Displays the progress of call origination/termination.
- Detail Message Information Window: Displays detail information on messages the TAPI Sampler received through the TAPI service.

Call Status Window

Parameters of the Call Status Window are as follows:

Parameter	Description
Time	Time when the event occurred.
CallHandle	Call Handle. Displayed in 4 byte hexadecimal.
CallState	Displays the call status as Idle, Connected, Busy, etc
CallStatDetail	Displays additional information, if any, on the call status.
ConnectedID	Displays the phone number of the other party.

Message Status Window

Parameters of the Message Status Window are as follows:

Parameter	Description
Time	Time when the event was received.
Event	Displays details on the received event.

This page is intentionally left blank.

CHAPTER 4

OpenTSP Driver Guide

This chapter describes the procedures for dialing, receiving, and disconnecting calls using the OpenTSP driver after successfully installing the driver on the PC.

The tools or programs used for procedures from setting the environment to processing calls are as follows.

Step	Item	Used Tool(Program)
1) Environment Setup	Set IP address and port	Phone and Modem Options (Advanced tab), OpenTSP Config Tool
	Set Trace type	Phone and Modem Options (Advanced tab), OpenTSP Config Tool
	Set Dialing Rules	Edit location of Phone and Modem Options (Refer to '1.3 Checking Telephony Service' in Chapter 2.)
2) Call Processing	Dialing, receiving, or disconnecting calls	Programs→Accessories→Communication →Dial, TAPI Sampler
3) Checking Call Processing Messages	OpenTSP driver operation status	Programs→Accessories→Communication →Dial, Message viewer, TAPI32 Browser
	Execute OpenTSP TAPI call function	TAPI32 Browser
	Receive call processing result log	Message viewer

TAPI Compatible Application Program

TAPI Compatible Application Programs can be used when the OfficeServ Link program is connected through the OpenTSP driver and is operating normally.

1 Environment Setup Procedure

There are two ways to set the environment for setting the IP address and port, for selecting the trace type, and for entering the licence key.

- Phone and Modem Options(Advanced tab)
- OpenTSP Config Tool

1.1 Setup through the Phone and Modem Options (Advanced tab)

Procedure for setting the environment through the Phone and Modem Options (Advanced tab) of the OS is as follows:

1) Select ‘Start→Settings→Control Panel’ on the computer.

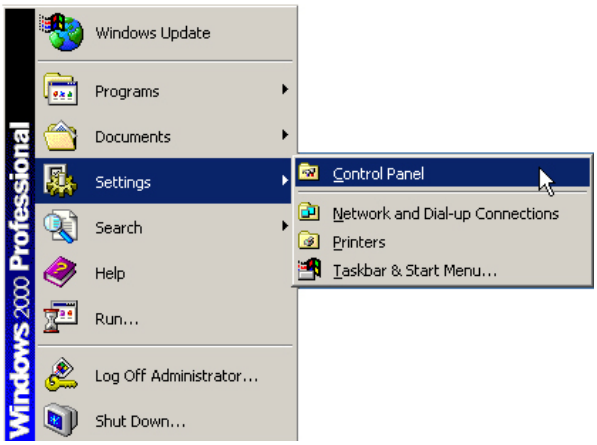


Figure 4.1 Starting Control Panel

2) Select the ‘Phone and Modem Options from the ‘Control Panel’ shown below.

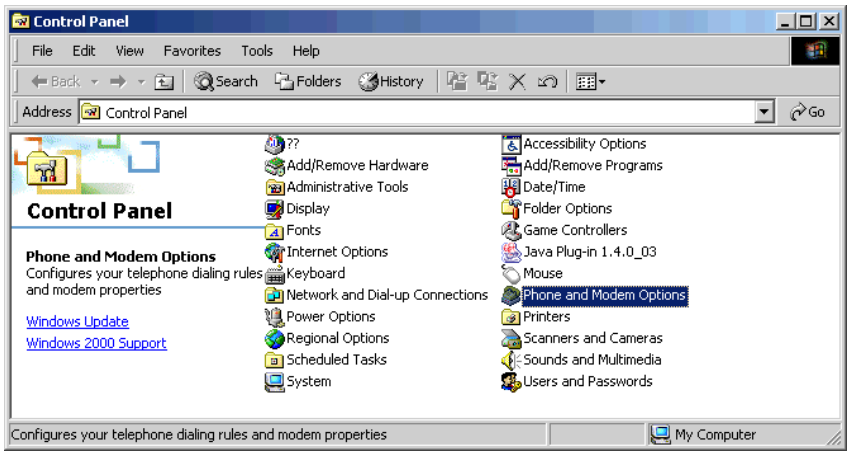


Figure 4.2 Selecting Phone and Modem Options

3) Click the [Edit (E)] button on the ‘Phone and Modem Options’ window.

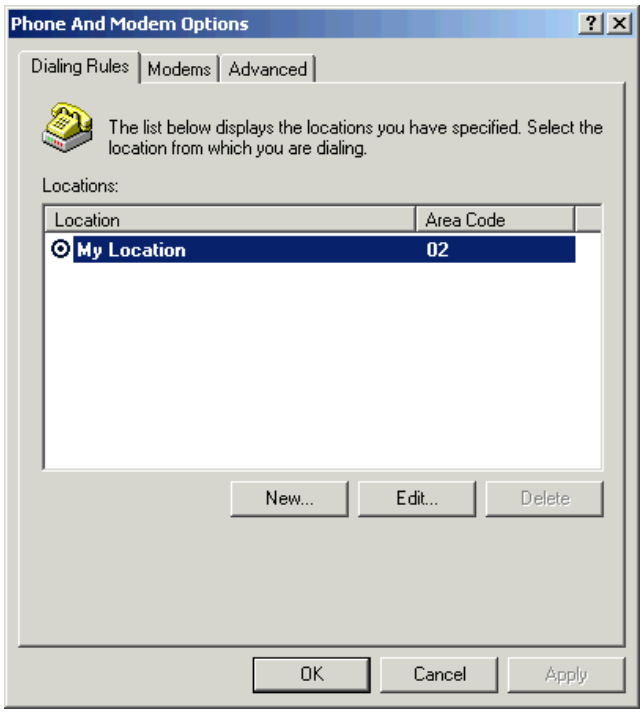


Figure 4.3 Phone and Modem Options Window

5) Select the ‘Advanced’ tab on the ‘Phone and Modem Options’ window.

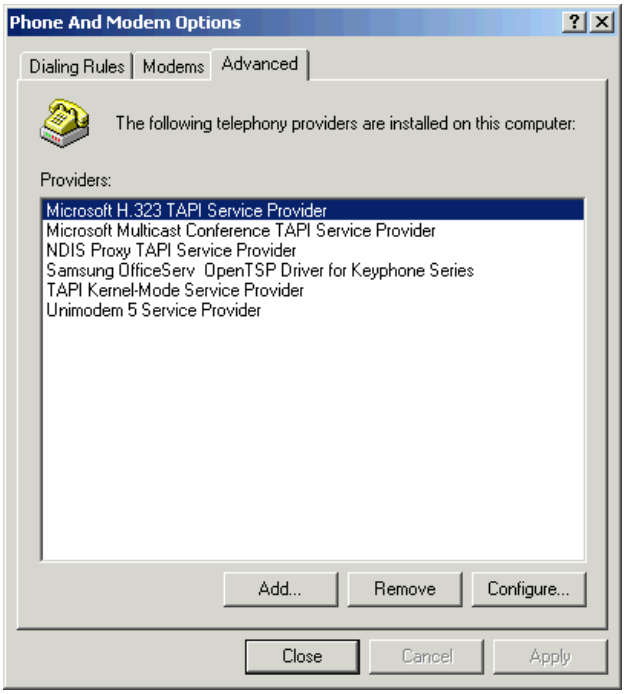


Figure 4.4 Advanced Tab of Phone and Modem Options Window

6) Select the ‘Samsung OfficeServ OpenTSP Driver for Keyphone Series’ and click the [Configure(C)] button.

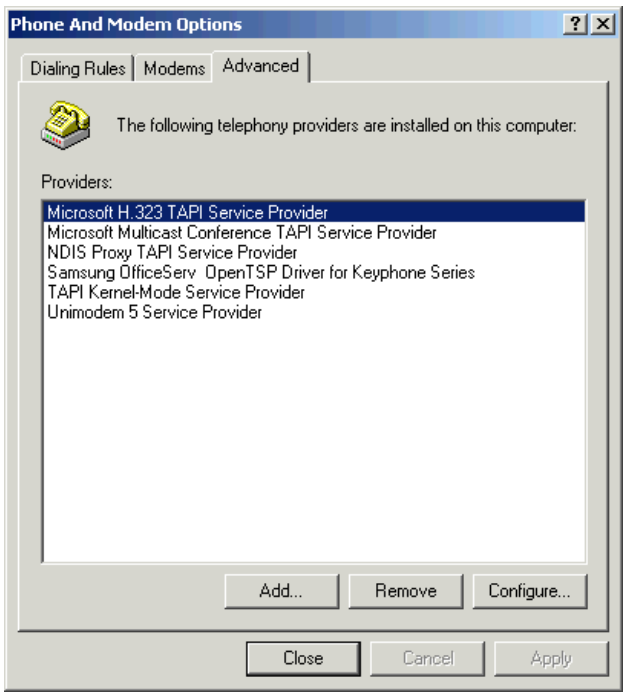


Figure 4.5 Selecting Configure Button of Advanced Tab

7) Enter the fields of the ‘Communication Parameters’ window and click the [OK] button.

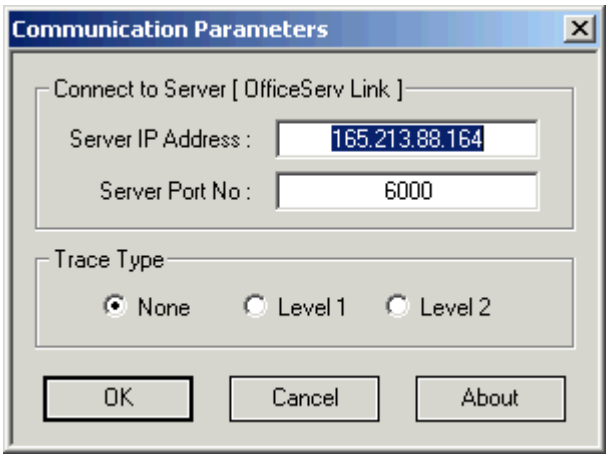


Figure 4.6 Communication Parameters Window

- Server IP Address: IP address of the PC where the OfficeServ Link is installed.
- Server Port No: Use the default number, 6000. (This port number should be set as the same port number set at the OfficeServ Link program.)
- Trace Type: Select ‘Level 1’ or ‘Level 2’ to display call processing messages or select ‘None’ not to display messages.

1.2 Setup through OpenTSP Config Tool

Procedure for setting the OpenTSP driver environment through the OpenTSP Config Tool is as follows:

1) Execute the 'OpenTSP Config Tool' under the 'Start→Programs→OfficeServ OpenTSP Driver'. The screen below is displayed.

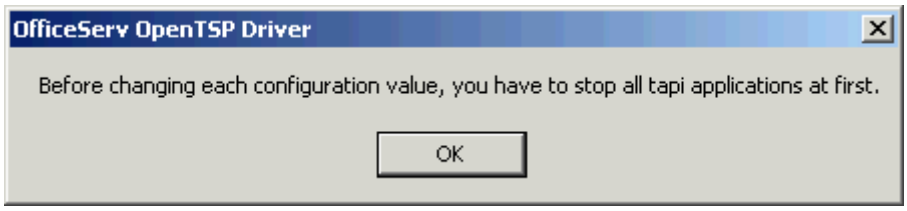


Figure 4.7 Closing TAPI Compatible Program

2) The message above informs the user that all TAPI compatible programs need to be closed before changing the environment settings. Close all TAPI compatible programs that are currently operating and click the [OK] button.

3) Enter the fields in the window shown below and click the [OK] button.

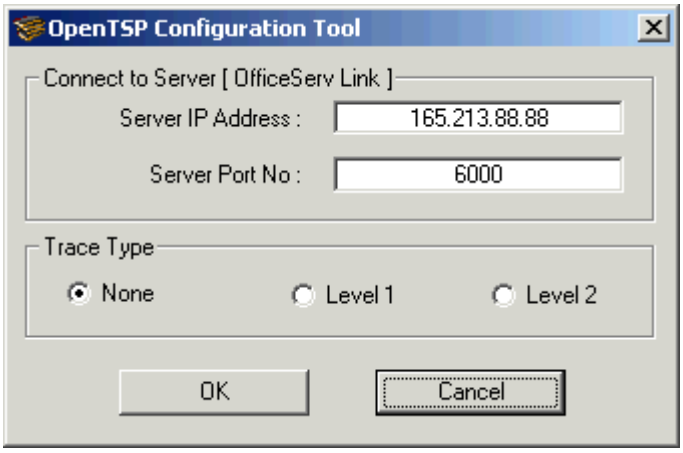


Figure 4.8 OpenTSP Config Tool

2 Call Processing

2.1 Call Processing of the Phone Dialer Program

A dialing program is installed as standard in all versions of Windows.

Dialing Procedure

Procedure for dialing through the 'Phone Dialer' is as follows:

1) Connect the Samsung Key telephone system and the OfficeServ Link program through the CTI link.

2) Execute the 'Phone Dialer' program of the PC by clicking [Start→Programs→ Accessories→Communication→Phone Dialer]. The 'Phone Dialer' screen shown below appears.

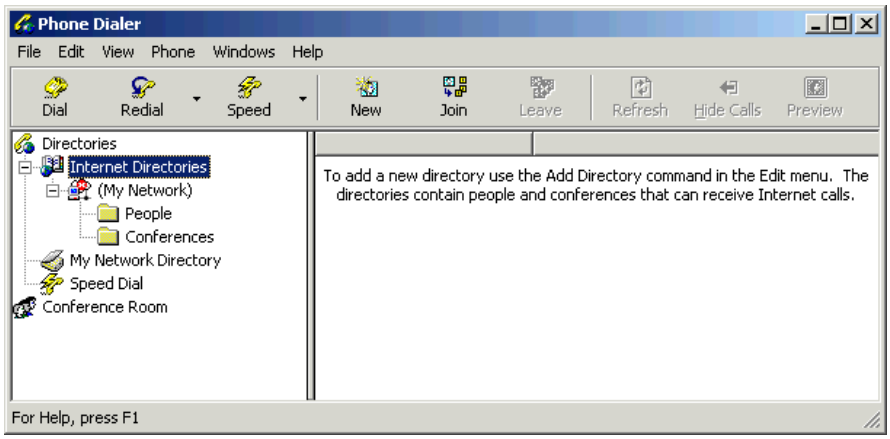


Figure 4.9 Phone Dialer Screen

3) Select [Edit→Option] and display the screen below.

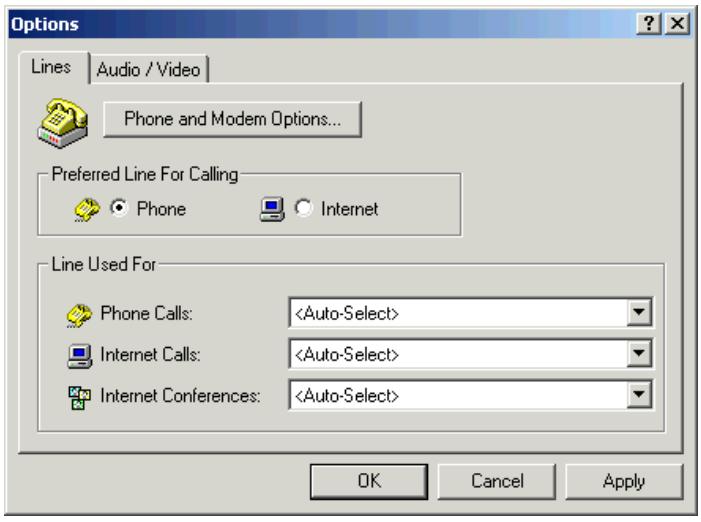


Figure 4.10 Option Screen

- Default line for dialing: Select ‘Telephone’ since the OpenTSP driver is used for line telephones.
  - Telephone (O): Selects the communication line for line telephones.
  - Internet (N): Selects the communication line for Internet lines.
- Used lines: Applies according to the default line used for dialing.
  - Phone (P): Sets telephone lines. This number should be identical to the actual extension number of the Samsung switch.
  - Internet Communication(I) :
  - Internet Conference(F) :

4) The settings of the ‘Line’ tab are displayed below. The ‘DCS Line 2001’ in the ‘Phone’ field represents that the extension number is 2001. The OpenTSP driver displays the device list of the Samsung key telephone system as ‘DCS Line XXXX (extension number)’.

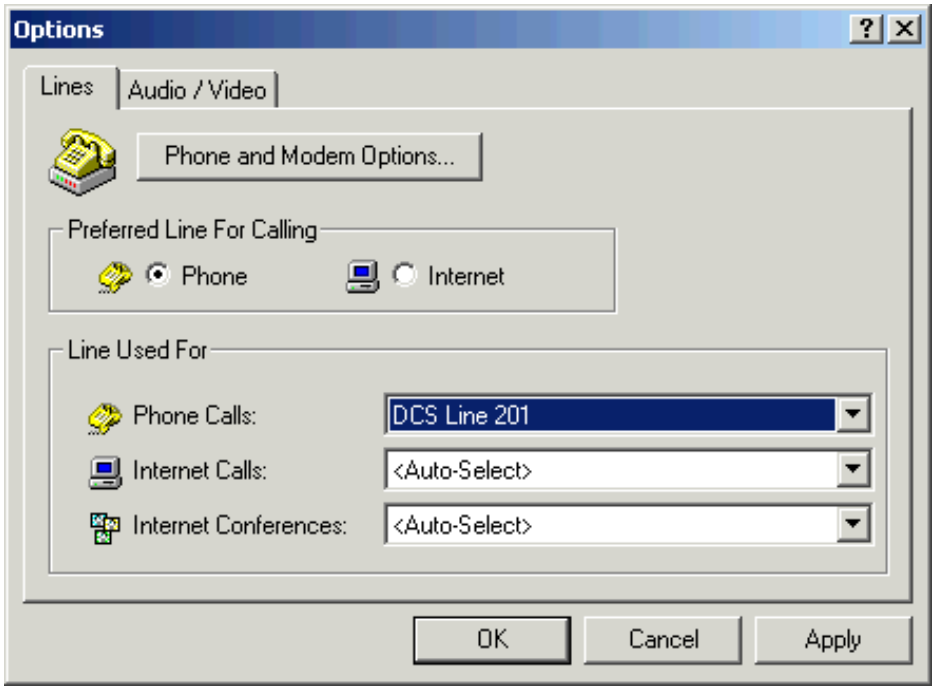


Figure 4.11 Line Tab of Option Screen

5) Select the 'Audio/Video (A)' tab and check if the **'Line'** item of the '**Dialing Device'** is set to **Telephone**, and click the [OK] button.

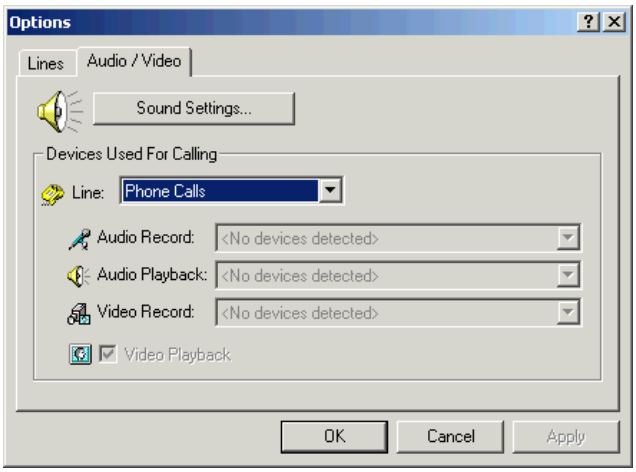


Figure 4.12 Audio/Video Tab of Option Screen

6) From the 'Phone Dialer' screen below, click 'Phone→Dial'.

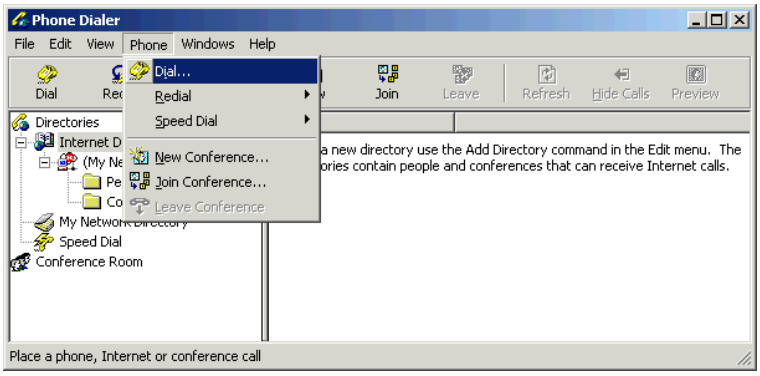


Figure 4.13 Selecting Dial from Phone Dialer Screen

7) From the 'Dial' screen below, check if the '**Dialing Pattern'** is set to Phone(P) and enter '2002' into the entry field. Then, Click the Connect(C) button.

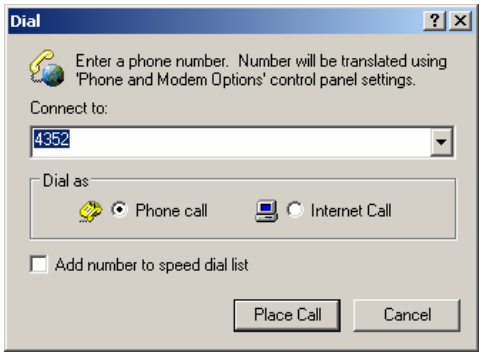


Figure 4.14 Dialing from the Dial Screen

8) The screen below appears and shows that extension 2001 is dialing extension 2002.

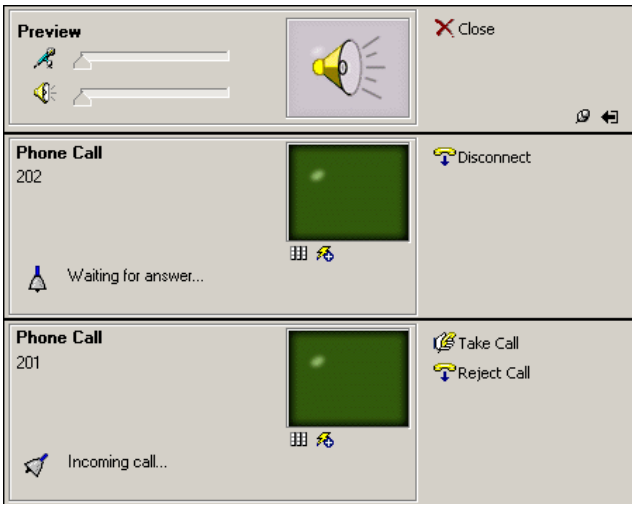


Figure 4.15 Dialing Display Screen

9) The screen below appears upon successful connection. Select Disconnect (D) to terminate the connection after completing the call.

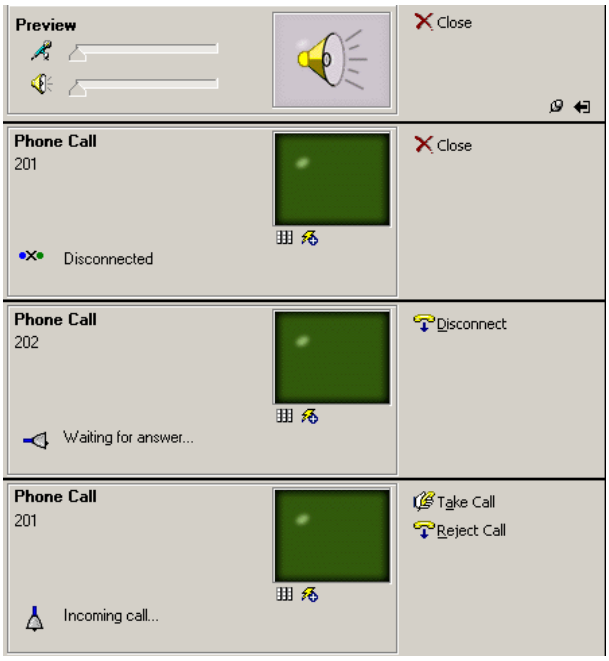


Figure 4.16 Disconnecting the Call

NOTE

Checking the OpenTSP driver operation

If the OpenTSP driver is successfully connected to the TAPI service through the dialing program, the messages exchanged can be viewed through the Message Viewer.

2.2 Call Processing of the TAPI Sampler

Users can dial, receive, or disconnect calls through the TAPI Sampler. The OfficeServ Link program should be normally running to 'Dial' using the TAPI program.

Dialing Calls

Procedure for dialing calls using the TAPI Sampler is as follows:

1) Click [Start→Programs→OfficeServ OpenTSP Driver→TAPI Sampler Tool] as shown below.

Figure 4.17 Executing the TAPI Sampler Tool

2) Select your extension number from the 'My Extension' field on the screen below and click the 'Open' button.

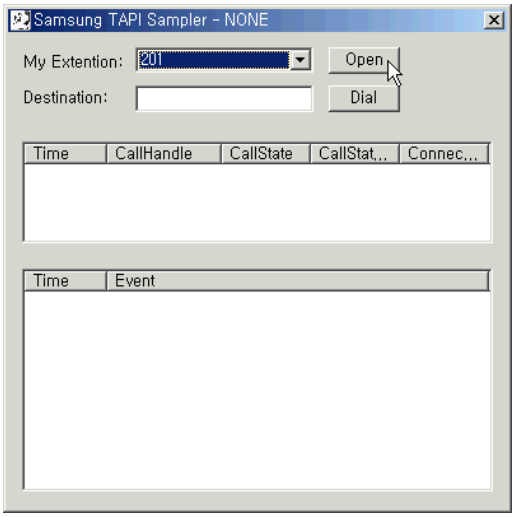


Figure 4.18 Selecting Extension Number

3) Enter the destination number in the ‘Destination’ field and click the ‘Dial’ button. Setting different numbers for the ‘My Extension’ and ‘Destination’ fields will enable receiving calls through TAPI Sampler.

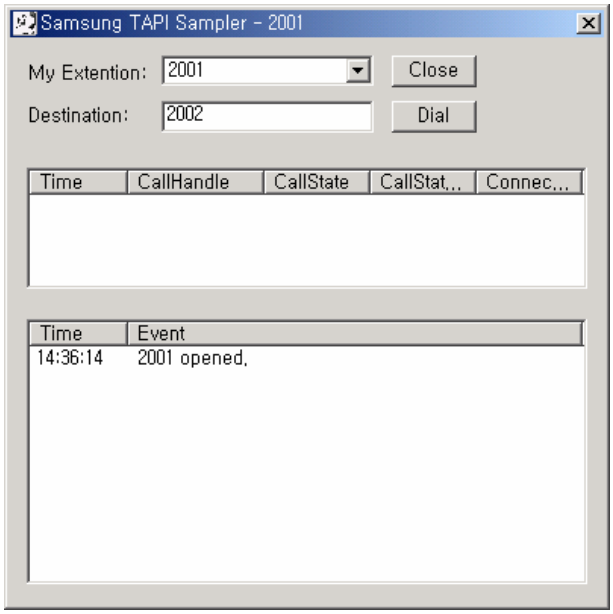


Figure 4.19 Selecting Destination Number

4) Information on the TAPI messages, sent to the TAPI Sampler by the TAPI service, are displayed ‘Message Status Window’.

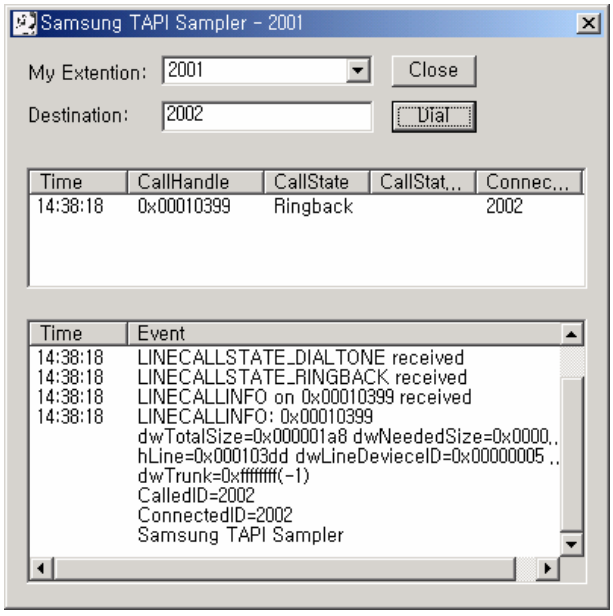


Figure 4.20 TAPI Message Display Screen

Receiving Calls

Calls may also be received through the TAPI Sampler program.

A call is being sent from extension 2002 to extension 2001 in the figure below. The Call Status Window displays information on the call, such as time of event, call status, and destination number, and the Message Status Window displays detail information on the TAPI messages sent to the TAPI Sampler by the TAPI service.

If a call arrives, right click the information of the call on the Call Status Window to display the Context Menu as shown below. Select [Answer] to answer the call.

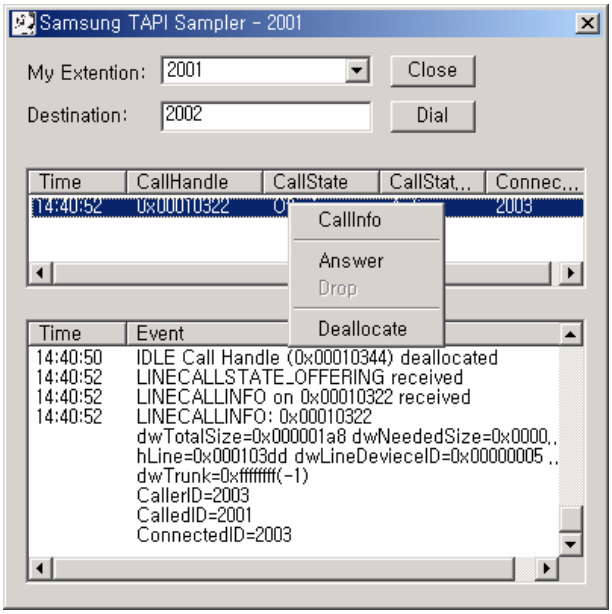


Figure 4.21 Context Menu Display Screen

The Context Menu is described as follows:

- CallInfo: Displays Detail information on the call on the Message Status Window. This feature displays the result of the TAPI function, lineGetCallInfo().
- Answer: Answers the call.
- Drop: Disconnects the connected call.
- Deallocate: Clears all displayed call information regardless of the phone status. This feature initializes the line device managed by the OpenTSP driver and deletes all call data on the corresponding line. (Irrespective to the actual status of the device) This feature can be used to manually remove any inconsistencies between the status of the actual phone and the reported status in TAPI..

3 Checking Call Processing Messages through the Message Viewer Tool

The Message Viewer Tool is used for verifying the messages processed during the operation of the OpenTSP driver.

> **Downloading the DBGView.exe file**
>
> As the execution file of the Message Viewer Tool, the DBGView.exe is used along with the OpenTSP driver. Made by Sysinternal, this program is used for logging debug messages within the Windows OS. Download and update your DBGView.exe file from http://www.sysintenal.com.

1) Click [Start→Programs→OfficeServ OpenTSP Driver→TAPI Sampler Tool] as shown below:

Figure 4.22 Executing the TAPI Sampler Tool

2) The Message Viewer Tool (= DBGView.exe) screen below appears and displays the messages exchanged during the operation of the OpenTSP driver.

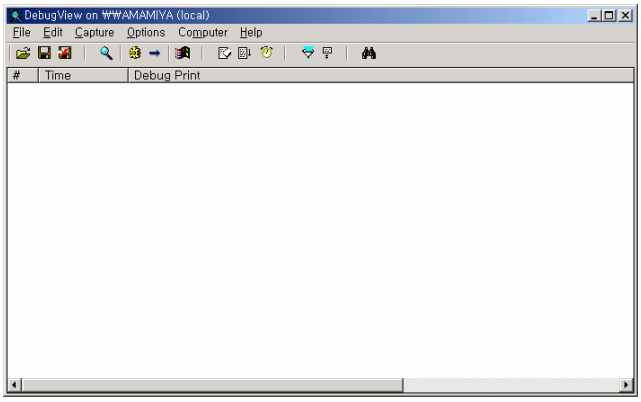


Figure 4.23 DBGView.exe Screen

Setting the message display level

Messages on the operation status of the OpenTSP driver are displayed on the Message viewer only when the Trace Type, the OpenTSP driver's environment setup item, is set to 'Level 1' or 'Level 2', and are not displayed when the Trace Type is set to 'None'.

Verifying the operation of the OpenTSP driver

Through the Message Viewer, users can view detail messages related to the operation of the OpenTSP driver and can also save the displayed messages as files if necessary.

CHAPTER 5

TAPI Functions

This chapter describes the features of the standard TAPI functions and Samsung Specific functions that the OpenTSP driver supports.

1 Relationship Between the TAPI and TSPI

When the TAPI-compatible application requests a TAPI function, the Telephony Service Provider provides the TSPI functions related to the TAPI function. That is, the TAPI-compatible application receives the TAPI service offered by the key telephone system through the Telephony Service Provider.

The procedure for exchanging messages between the TAPI and TSPI is shown in the figure below:

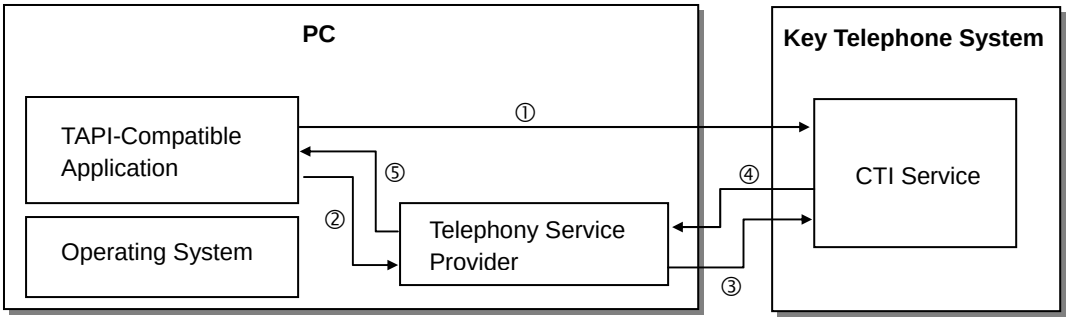


Figure 5.1 Flow of Messages Between the TAPI and TSPI

Each step shown in Figure 5.1 is described below:

① The TAPI-compatible application calls a TSPI function to the TAPI service of the key telephone system in order to process calls.

② The TAPI-compatible application calls a TSPI function to the Telephony Service Provider.

③ The Telephony Service Provider forwards the event requested by the TAPI-compatible application to the TAPI service of the key telephone system.

④ The TAPI service of the key telephone system processes the event and notifies the Telephony Service Provider of the results.

⑤ The Telephony Service Provider forwards the results received from the CTI service of the key telephone system to the TAPI-compatible application.

Example of Internal Calling Using the Phone Dialer

Figure 5.2 shows the procedures for using the phone dialer offered by the PC to make a call from Extension 2001 to Extension 2002:

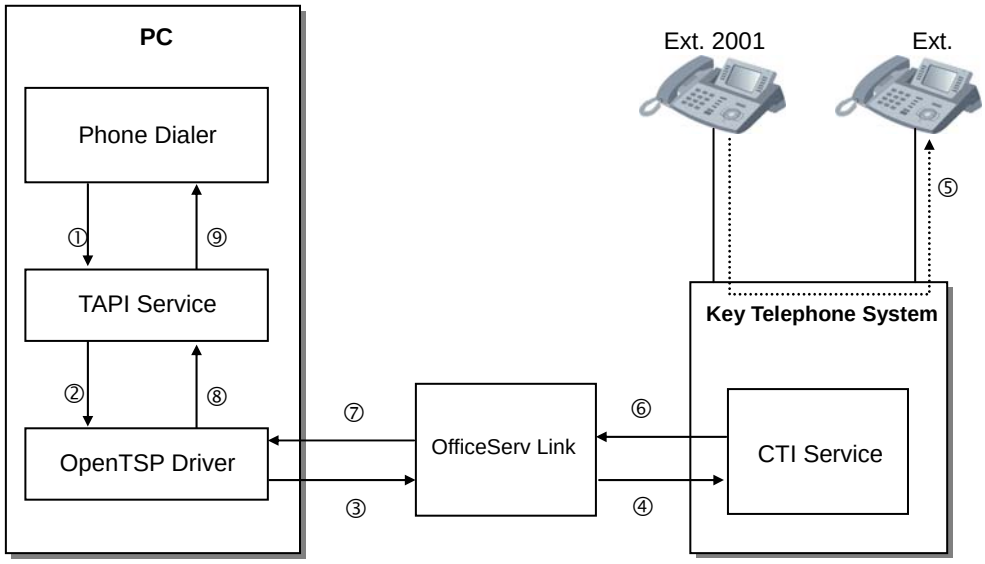


Figure 5.2 Example of Internal Calling in the Phone Dialer

The commands and events to be sent or received during internal calling are processed as described below:

① lineMakeCall : The Phone Dialer uses the lineMakeCall TAPI function to press Extension 2002 for making a call.

② TSPI_lineMakeCall : The TAPI service calls the TSPI_lineMakeCall function to the OpenTSP driver after being asked to process the TAPI function.

③ The OpenTSP driver creates the command that can be processed by the key telephone system and forwards it to the OfficeServ Link program in order to perform the functions requested by the TAPI service.

④ The OfficeServ Link forwards the command received from each OpenTSP driver to the key telephone system of Samsung.

⑤ The key telephone system interprets the forwarded command to make a call from Extension 2001 to Extension 2002.

⑥ The key telephone system forwards the extension processing results to the OfficeServ Link.

⑦ The OfficeServ Link forwards the extension processing results received from the key telephone system to the OpenTSP driver.

⑧ The OpenTSP driver converts the event for the processing results into the form that can be processed by the TAPI service and forwards the event to the TAPI service.

⑨ The TAPI service forwards the results for TAPI function processing to the Phone Dialer through the TAPI service.
Once the steps above are completed, the Phone Dialer offers the call processing results to users through an internal processing module.

2 List of the TAPI Functions

Restriction

The OpenTSP driver supports INTERACTIVEVOICE mode and only the Line Device function out of the list of the TAPI functions of Microsoft.

List of the TAPI Functions

The list of the TAPI functions that the OpenTSP driver enables is shown below:

TAPI Functions of Microsoft	Supported or Not	Remarks
LineAddToConference	O	Consultation Call
LineAnswer	O	Off-Hook
LineBlindTransfer	O	Consultation Call+Transfer
LineClose	O	
LineCompleteCall	O	Camp on+Msg Waiting+OHVA+Callback
LineCompleteTransfer	O	Transfer
LineDeallocateCall	O	Idle Call Remove
LineDevSpecific	O	Refer to 5.3 List of the OpenTSP Driver Expansion Functions.
LineDial	O	Make Call
LineDrop	O	On-Hook
LineForward	O	Set/Reset Forward/DND
LineGenerateDigits	O	Send DTMF Digits
LineGetAddressCaps	O	
LineGetAddressID	O	
LineGetAddressStatus	O	
LineGetCallInfo	O	
LineGetCallStatus	O	
LineGetDevCaps	O	
LineGetDevConfig	O	
LineGetID	O	
LineGetLineDevStatus	O	
LineHold	O	Hold
lineMakeCall	O	Make Call
lineNegotiateExtVersion	O	
lineOpen	O	
linePark	O	Direct Park : OK, UnDirect Park[=System Hold] : OK
linePickup	O	Direct Pickup+Group Pickup

TAPI Functions of Microsoft	Supported or Not	Remarks
linePrepareAddToConference	O	Consultation Call
lineRedirect	O	Redirect
linePark	O	Direct Park : OK, UnDirect Park[=System Hold] : OK
lineRemoveFromConference	O	Consultation Call
lineSetAppSpecific	O	
LineSetCallData	O	
lineSetCallParams	O	
lineSetMediaMode	O	
lineSetStatusMessages	O	
lineSetupConference	O	Consultation Call
lineSetupTransfer	O	Consultation Call
lineSwapHold	O	Consultation Call for T-Hold And Hold+Retrieve for S-Hold
lineUnhold	O	Consultation Call for T-Hold and Retrieve for S-Hold
lineUnpark	O	System Hold Retrieval

The list above shows only the TAPI functions supported by the OpenTSP driver: Some functions from the list of the TAPI functions might be processed by the TAPI service itself. Also, some functions, which are used to add the Telephony Service Provider to the system, are not included in the list. If the functions that are not supported by the OpenTSP driver are called, an error message defined in the TAPI will appear.
The user can find the entire list of the Microsoft TAPI functions from the Microsoft site (http://www.msdn.microsoft.com/library/default.asp). The user can check the format of each function, how to use the functions, or the status values returned from the list and refer to the list to develop an application.

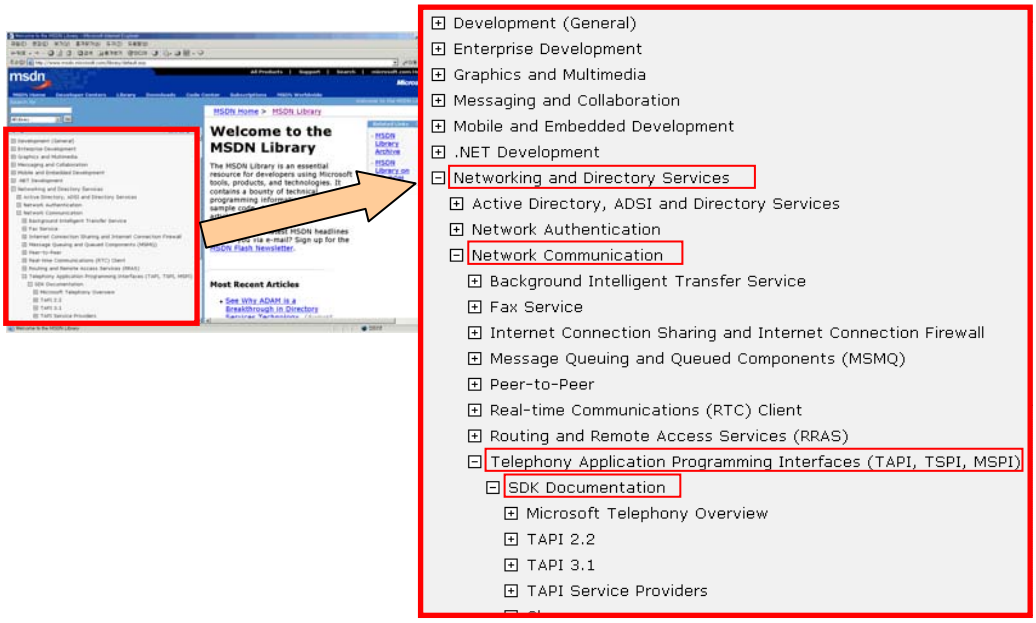


Figure 5.3 URL of the Entire List of Microsoft TAPI Functions

3 Feature List of the Expansion Functions in the OpenTSP Driver

Besides the functions defined by the TAPI, the OpenTSP driver provides a variety of expansion functions.
Call the lineDevSpecific function to use the expansion functions.

Enter the syntax below to call the lineDevSpecific function:

```
LONG lineDevSpecific (HLINE hLine, DWORD dwAddressID, HCALL hCall,
LPVOID lpParams, DWORD dwSize);
```

The features available by the lineDevSpecific function in the Samsung key telephone system are as follows:

- Station Lock
- Vacant Station Message
- Follow Me
- Make New Trunk Call
- Page
- System Hold Retrieval
- Clear Message Waiting
- Clear Call Back
- OHVA
- Silent Monitoring
- Mute On/Off
- Line Reset

3.1 Station Lock

The Station Lock disables other users from using their own phones to make or answer calls.

The available modes are as follows:

- Unlock: Release lock.
- Locked all: Lock call outgoing and incoming.

Enter the syntax below to call the lineDevSpecific function when the Station Lock is used in the TAPI application:

```
LONG lineDevSpecific (HLINE hLine, DWORD dwAddressID, HCALL hCall,
LPVOID lpParams, DWORD dwSize);
```

Input Parameter Values

- hLine : Processes the lines to be used.
- dwAddressID : 0
- hCall : Not used
- lpParams : Enters the command strings as shown below :

Value	Byte
'D' 'C' 'S'	3 Bytes
'L'	1 Byte
Option : 0-Unlock 2-Lock All	1 Byte
Phone Password	Up to 4 bytes

- DwSize : Buffer length(Null value included)

3.2 Vacant Station Message

The Vacant Station Message enables the phone to display a vacant message on the LED of the caller's extension phone when a user sets the 'vacant message' to the user's phone before he or she is away from the phone.

Number of Vacant Messages

The number of vacant messages that can be set to the system depends on the Samsung key telephone systems.

Enter the syntax below to call the lineDevSpecific function when the Vacant Station Message is used in the TAPI application:

```
LONG lineDevSpecific (HLINE hLine, DWORD dwAddressID, HCALL hCall,
LPVOID lpParams, DWORD dwSize);
```

Input Parameter Values

- hLine : Processes the lines to be used.
- dwAddressID : 0
- hCall : Not used
- lpParams : Enters the command strings as shown below :
- DwSize : Buffer length(Null value included)

Value	Byte
'D' 'C' 'S'	3 Bytes
'V'	1 Byte
Message number(Hexa value) : 0-Clears a message. 1~20-Number of the messages	1 Byte

3.3 Follow Me

The Follow Me function enables call forwarding so that the user can answer a call even if the user is away from the phone. This feature is the same as ‘call forwarding unconditional.’ However, the ‘call forwarding unconditional’ is set in the user’s phone while the Follow Me is set in another phone.

Enter the syntax below to call the lineDevSpecific function when the Follow Me is used in the TAPI application:

```
LONG lineDevSpecific (HLINE hLine, DWORD dwAddressID, HCALL hCall,
LPVOID lpParams, DWORD dwSize);
```

Input Parameter Values

- hLine : Processes the lines to be used.
- dwAddressID : 0
- hCall : Not used
- lpParams : Enters the command strings as shown below :
- DwSize : Buffer length(Null value included)

Value	Byte
‘D’ ‘C’ ‘S’	3 Bytes
‘F’	1 Byte
Phone number to be forwarded	Up to 4 Bytes

3.4 Make New Trunk Call

The Make New Trunk Call enables the user to make a trunk call continuously without making the call again even after the trunk call is completed.

Enter the syntax below to call the lineDevSpecific function when the Make New Trunk Call is used in the TAPI application:

```
LONG lineDevSpecific (HLINE hLine, DWORD dwAddressID, HCALL hCall,
LPVOID lpParams, DWORD dwSize);
```

Input Parameter Values

- HLine : Processes the lines to be used.
- dwAddressID : 0
- hCall : Not used
- lpParams : Enters the command strings as shown below :
- DwSize : Buffer length(Null value included)

Value	Byte
'D' 'C' 'S'	3 Bytes
'T'	1 Byte
Digit of the dialed phone number-n	1 Byte
Digit to be dialed	n Byte(s)

3.5 Page

The Page enables the user to give a notice to people simultaneously through the speaker installed on the key telephone (or the external speaker installed separately). The page is categorized into internal page and external page. The internal page is made to the key phones, which are set as the internal page zone of the current key telephone system. The external page is made through the speakers, which are set as the external page zone. When the external page is made, an external speaker should be set in the <System Programming> of the Samsung key telephone system.

Enter the syntax below to call the lineDevSpecific function when the Page is used in the TAPI application:

```
LONG lineDevSpecific (HLINE hLine, DWORD dwAddressID, HCALL hCall,
LPVOID lpParams, DWORD dwSize);
```

Input Parameter Values

- hLine: Processes the lines to be used.
- dwAddressID : 0
- hCall : Not used
- lpParams : Enters the command strings as shown below :
- DwSize : Buffer length(Null value included)

Value	Byte
‘D’ ‘C’ ‘S’	3 Bytes
‘P’	1 Byte
Number of page	1 Byte

The page zone numbers are described below:

- ‘1’~‘4’ : Internal page zone
- ‘5’~‘8’ : External page zone
- ‘0’ : Entire internal page
- ‘9’ : Entire external page
- ‘*’ : Entire internal/external page

3.6 System Hold Retrieval

The System Hold Retrieval enables the user to ‘hold’ an incoming call momentarily and answer the call from another extension.

Enter the syntax below to call the lineDevSpecific function when the System Hold Retrieval is used in the TAPI application:

```
LONG lineDevSpecific (HLINE hLine, DWORD dwAddressID, HCALL hCall,
LPVOID lpParams, DWORD dwSize);
```

Input Parameter Values

- hLine : Processes the lines to be used.
- dwAddressID : 0
- hCall : Not used
- lpParams : Enters the command strings as shown below :
- DwSize : Buffer length(Null value included)

Value	Byte
‘D’ ‘C’ ‘S’	3 Bytes
‘S’	1 Byte
Number of the calls on hold	Up to 4 Bytes

3.7 Clear Message Waiting

The Clear Message Waiting disables the message waiting LED to be displayed when a message is left in the user's phone.

Checking if a message has been left

In the Samsung key telephone system, the LED on the message button of the connected phone turns on to notify the other party that a message has been left when a caller leaves the message because the caller cannot speak to the other party.

Enter the syntax below to call the lineDevSpecific function when the Clear Message Waiting is used in the TAPI application:

```
LONG lineDevSpecific (HLINE hLine, DWORD dwAddressID, HCALL hCall,
LPVOID lpParams, DWORD dwSize);
```

Input Parameter Values

- hLine : Processes the lines to be used.
- dwAddressID : 0
- hCall : Not used
- lpParams : Enters the command strings as shown below :
- DwSize : Buffer length(Null value included)

Value	Byte
'D' 'C' 'S'	3 Bytes
'M'	1 Byte
Number of the devices where a message is left	Up to 4 Bytes

3.8 Clear Call Back

The Call Back enables the user to make a call reservation when the other party is on the phone or does not answer. Then, a caller's phone rings automatically when the other party's phone becomes available. The Clear Call Back disables the Call Back.

Enter the syntax below to call the lineDevSpecific function when the Call Back is used in the TAPI application:

```
LONG lineDevSpecific (HLINE hLine, DWORD dwAddressID, HCALL hCall,
LPVOID lpParams, DWORD dwSize);
```

Input Parameter Values

- hLine : Processes the lines to be used.
- dwAddressID : 0
- hCall : Not used
- lpParams : Enters the command strings as shown below :
- DwSize : Buffer length(Null value included)

Value	Byte
'D' 'C' 'S'	3 Bytes
'C'	1 Byte
Phone number to which a reservation has been made	Up to 4 Bytes

3.9 OHVA

The Off Hook Voice Announcement (OHVA) enables a caller to leave a message in the other party's phone when the other party is on the phone. This is useful when the caller needs to leave a message urgently.

Enter the syntax below to call the lineDevSpecific function when the OHVA is used in the TAPI application:

```
LONG lineDevSpecific(HLINE hLine, DWORD dwAddressID, HCALL hCall,
LPVOID lpParams, DWORD dwSize);
```

Input Parameter Values

- hLine : Processes the lines to be used.
- dwAddressID : 0
- hCall : Not used
- lpParams : Enters the command strings as shown below :
- DwSize : Buffer length(Null value included)

Value	Byte
'D' 'C' 'S'	3 Bytes
'I'	1 Byte
Phone number for the OHVA	Up to 4 Bytes

3.10 Silent Monitoring

The Silent Monitoring enables a caller to speak to the extension subscriber by interruption even while the subscriber is on the phone. In the Samsung key telephone system, the Silent Monitoring operates in 'Without Tone (the monitored subscriber does not the monitoring sound).'

Enter the syntax below to call the lineDevSpecific function when the Silent Monitoring is used in the TAPI application:

```
LONG lineDevSpecific(HLINE hLine, DWORD dwAddressID, HCALL hCall,
LPVOID lpParams, DWORD dwSize);
```

Input Parameter Values

- hLine : Processes the lines to be used.
- dwAddressID : 0
- hCall : Not used
- lpParams : Enters the command strings as shown below :
- DwSize : Buffer length(Null value included)

Value	Byte
'D' 'C' 'S'	3 Bytes
'B'	1 Byte
Phone number for silent monitoring	Up to 4 Bytes

3.11 Mute On/Off

The Mute On/Off enables (Mute Off) or disables (Mute On) the other party to listen to a caller's voice while the caller is speaking to the other party or is on the Intrude or Silent Monitoring. Although a caller sets the Mute On, the caller can listen to the other party's voice.

Enter the syntax below to call the lineDevSpecific function when the Mute On/Off is used in the TAPI application:

```
LONG lineDevSpecific (HLINE hLine, DWORD dwAddressID, HCALL hCall,
LPVOID lpParams, DWORD dwSize);
```

Input Parameter Values

- hLine : Processes the lines to be used.
- dwAddressID : 0
- hCall : Not used
- lpParams : Enters the command strings as shown below :
- DwSize : Buffer length(Null value included)

Value	Byte
'D' 'C' 'S'	3 Bytes
'm'	1 Byte
Phone number for Mute On/Off	Up to 4 Bytes

3.12 Line Reset

The Line Reset enables the user to initialize the device managed by the OpenTSP driver forcibly when the call status of the device is different from that of the Samsung key telephone system.

The status of calls in each device of the OpenTSP driver should be the same as that of calls in the Samsung telephone system. However, the device status of the PBX might be different from the call status of the device managed by the OpenTSP driver due to an error during the operation of the OpenTSP driver.
In this case, the request of initializing the device in the application can be made. The initialization can be made in the two ways described below: The first way is that only the device managed by the OpenTSP driver is initialized. That is, only the device of the OpenTSP driver is initialized irrespective of the device status of the PBX. The second way is that the device of the PBX is initialized along with the device of the OpenTSP driver. The two ways of initialization can be requested by the application at a right time if needed.

When Only the Device of the OpenTSP Driver is initialized

- hLine : Processes the lines to be used.
- dwAddressID : 0
- hCall : Not used
- lpParams : Enters the command strings as shown below :
- DwSize : Buffer length(Null value included)

Value	Byte
'D' 'C' 'S'	3 Bytes
'R'	1 Byte
Optional Value '0/1' '0' : Deletes all the calls located on the line. '1' : Deletes the disconnected calls	1 Byte

When the Device of the PBX is initialized as Well

- hLine : Processes the lines to be used.
- dwAddressID : 0
- hCall : Not used
- lpParams : Enters the command strings as shown below :
- DwSize : Buffer length(Null value included)

Value	Byte
'D' 'C' 'S'	3 Bytes
'x'	1 Byte
Optional Value : '0' : Reads the information on the current line status.(Not used) '1' : Initializes the current line status.	1 Byte

This page is intentionally left blank.

CHAPTER 6

Call Processing Flow

This chapter describes the life cycle of the TAPI, the call processing events of the OpenTSP driver, and call processing procedures.

1 Life Cycle of the TAPI

The knowledge of the TAPI life cycle shown below is needed to use the TAPI-based application to process calls:

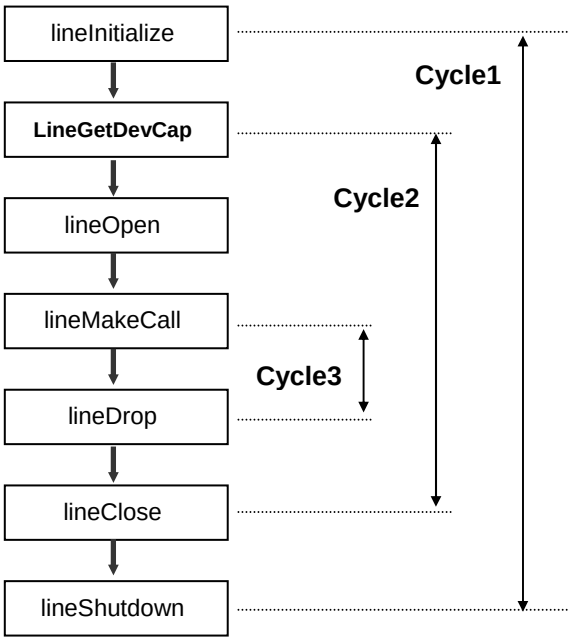


Figure 6.1 Life Cycle of the TAPI

Cycles 1 to 3 shown in Figure 6.1 are described below:

Cycle 1

The Phone Dialer and call center makes/connect/answers a call according to the steps of Cycle 1 as shown in Figure 6.1 :
Each application can use the lineInitialize() function and then other TAPI functions.

Also, the lineShutdown() function should be called to prevent the TAPI-compatible application from using the TAPI function.

Each TAPI-compatible application can call the lineInitialize() function to check the number of the devices available by the TAPI service and register the processing modules for call processing events generated from each device. Also, the TAPI service loads the unloaded Telephony Service Provider (TSP) on the TAPI service by executing the lineInitialize() function to change each TSP driver to an operating state. Different TAPI-compatible applications can simultaneously call the lineInitalize() function. The information registered during each calling is automatically classified and sorted by the TAPI service.

The lineShutdown() function is used when each TAPI-compatible application does not use the TAPI functions any more. If this function is called, the TAPI service will delete the information registered during the lineInitialize() process in order not to report the call status events generated from each device.
Also, if the lineShutdown() function is called when any application does not use the TAPI service, the TAPI service will upload all the loaded TSPs.

Cycle 2

Call the lineInitialize() function to find out the number of devices available in the TAPI service of the system. Then, the TAPI-compatible application calls the lineOpen() function to make necessary line devices available to each application. The application, which has got permissions for the line after executing the lineOpen() function, can receive information on call processing in each line device and use functions on calls.
The TAPI-compatible application calls the lineClose() function when the application disables the line devices. If the lineClose() function is called, the call processing events generated from the line devices will not be reported and the functions for call processing cannot be used for the line devices.

Cycle 3

The TAPI-compatible application that has permissions for each line device through the lineOpen() function can use call processing functions for the line. Also, since the TAPI-compatible application receives the call processing events for the status of all calls, it can be defined to perform necessary operations according to the processing rule of the application. The call processing functions are available only if call objects exist in the line device.

How to Use Functions

For information about how to use the functions, refer to the Microsoft web sites about the TAPI.

2 Call Processing Events for the OpenTSP Driver

The TAPI service of the system offers the call processing events generated from the line device to the application after calling the lineOpen() function so that the TAPI-compatible application can use a specific line device as shown in the TAPI Life Cycle 2 of Figure 6.1.
This section describes the type of the call processing events to be reported while, the call processing events are generated from the Samsung key telephone system and forwarded to the TAPI service through the OpenTSP driver as well as processing procedures.

2.1 Major Events

LINE_CALSTATE and LINE_CALLINFO are the events that all the TAPI-compatible applications should process by default. These events are reported when the status of a specific call and the details of each call are changed in each line device.

LINE_CALLSTATE

The LINE_CALSTATE event is reported when the status of calls is possibly changed in each line device. Examples of the call status include IDLE, RINGBACK, OFFERING, CONNECT, HOLD, and DISCONNECT. The call status is reported in event of status transition. When the LINE_CALLSTATE event is generated, the TAPI-compatible application calls the lineGetCallState() function to read the details of call status.

LINE_CALLINFO

The LINE_CALLINFO event is reported when information on calls in each line device is changed. Information on calls needed during call processing includes caller ID/name, called party ID/name, the phone number/name of the person to whom a call is forwarded, and call status. The information can be changed. When the information is changed, the LINE_CALLINFO event is reported. The TAPI-compatible application calls the lineGetCallInfo() function to read the details when the LINE_CALLINFO event is generated.

2.2 Flow Chart of Call Status

When call status is changed, the OpenTSP driver reports information on call status through the LINE_CALLSTATE and LINE_CALLINFO events. The LINE_CALLSTATE event reports the information on the status change of the generated calls and the LINE_CALLINFO event reports the information on the details of each call are changed.
The figure below briefly shows the change of call status from call generation to termination:

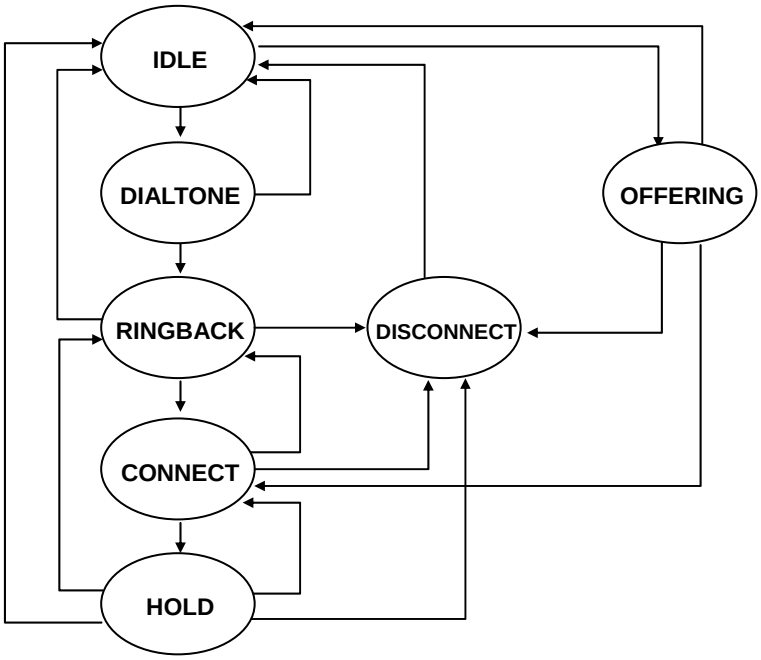


Figure 6.2 Example of Flow Chart of Call Status

Outgoing and incoming calls are exemplified below:

When Making a Call from Extension 2001 to Extension 2002

The procedure for making a call from Extension 2001 is as follows:

- IDLE→DIALTONE→RINGBACK→CONNECTED→DISCONNECTED→ IDLE

The procedure for connecting a call with Extension 2002 is as follows:

- IDLE→Offering→CONNECTED→DISCONNECT→IDLE

When call status is changed as described above, the OpenTSP driver forwards the LINE_CALLSTATE event to the TAPI-compatible application through the TAPI service. The TAPI-compatible application calls the lineGetCallInfo() function to obtain the details of call status.

2.3 Flow Chart of the Status of Calls in Progress

The events and messages generated while a call is being processed (i.e. from making a call to connecting a call) by the TAPI-compatible application are as follows:
The figure below shows the example of events on call status sent to the TAPI-compatible applications of both a caller and called party when a call is in progress:

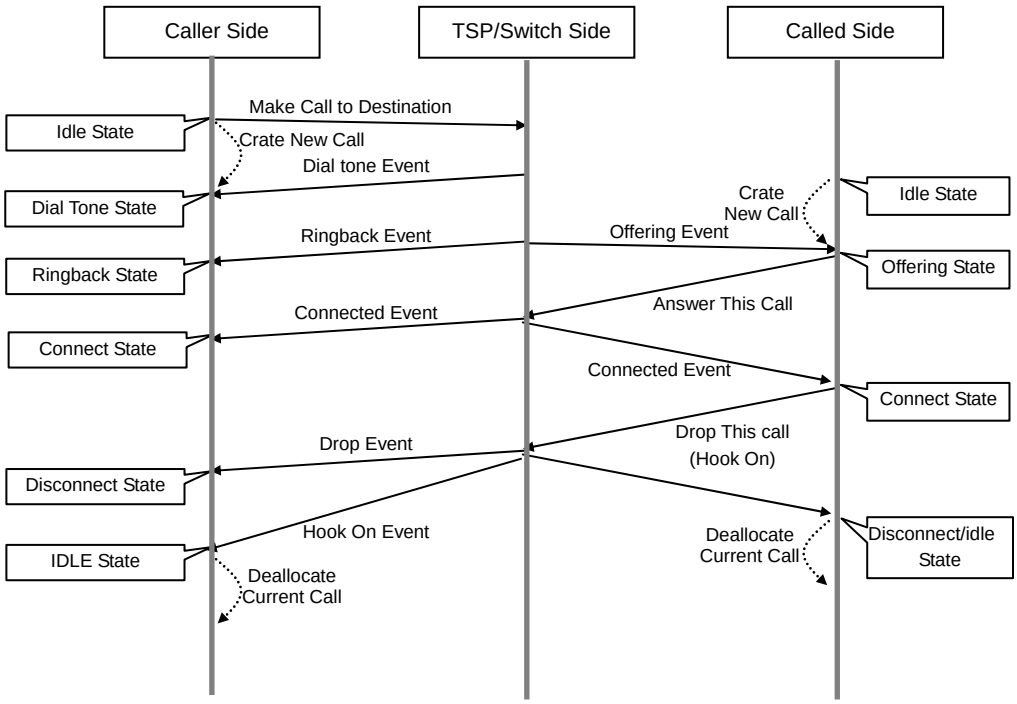
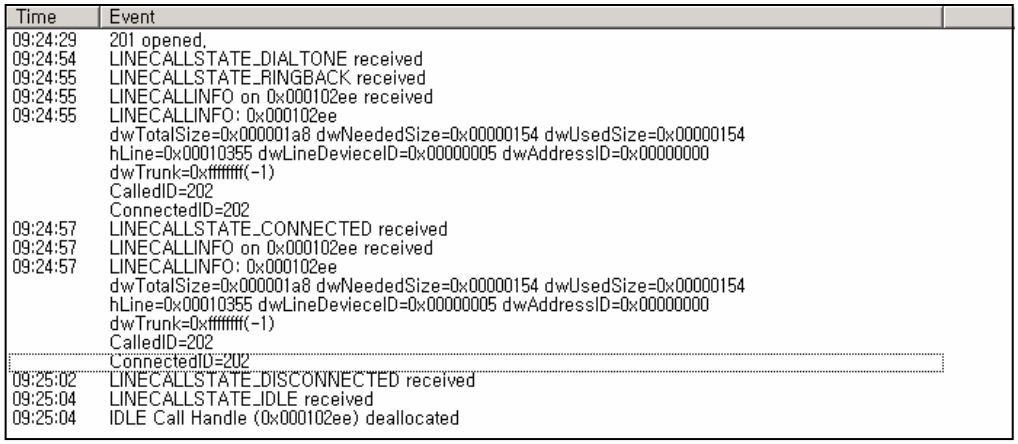


Figure 6.3 Flow Chart of the Status of Calls in Progress

2.4 Details of Calls

When call status is changed, for example when a call is made from Extension 201 to Extension 202 by using the TAPI Sampler program, the status message of the call is displayed in real time for each event. The call status change messages include caller IDs, called IDs, trunk numbers, DNIS information, call directions, and reasons for call generation.
As shown in the displayed screen below, the messages of call status show both the status change of the call (LINE_CALLSTATE) and the change of the details of each call (LINE_CALLINFO). Once the details of calls are changed, the details are forwarded from the OpenTSP to the TAPI service through the LINE_CALLINFO event. Also, the TAPI-compatible application calls the lineGetCallInfo() function to read the changed information or one to be checked.

Caller (Extension 201)



Time	Event
09:24:29	201 opened.
09:24:54	LINECALLSTATE_DIALTONE received
09:24:55	LINECALLSTATE_RINGBACK received
09:24:55	LINECALLINFO on 0x000102ee received
09:24:55	LINECALLINFO: 0x000102ee
	dwTotalSize=0x000001a8 dwNeededSize=0x00000154 dwUsedSize=0x00000154
	hLine=0x00010355 dwLineDevieceID=0x00000005 dwAddressID=0x00000000
	dwTrunk=0xffffffff(-1)
	CalledID=202
	ConnectedID=202
09:24:57	LINECALLSTATE_CONNECTED received
09:24:57	LINECALLINFO on 0x000102ee received
09:24:57	LINECALLINFO: 0x000102ee
	dwTotalSize=0x000001a8 dwNeededSize=0x00000154 dwUsedSize=0x00000154
	hLine=0x00010355 dwLineDevieceID=0x00000005 dwAddressID=0x00000000
	dwTrunk=0xffffffff(-1)
	CalledID=202
	ConnectedID=202
09:25:02	LINECALLSTATE_DISCONNECTED received
09:25:04	LINECALLSTATE_IDLE received
09:25:04	IDLE Call Handle (0x000102ee) deallocated

Figure 6.4 Messages of Call Status

Called Party (Extension 202)

Time	Event
09:24:36	202 opened.
09:24:55	LINECALLSTATE_OFFERING received
09:24:55	LINECALLINFO on 0x000102cc received
09:24:55	LINECALLINFO: 0x000102cc
	dwTotalSize=0x000001a8 dwNeededSize=0x0000015c dwUsedSize=0x0000015c
	hLine=0x00010311 dwLineDevieceID=0x00000006 dwAddressID=0x00000000
	dwTrunk=0xffffffff(-1)
	CallerID=201
	CalledID=202
	ConnectedID=201
09:24:57	LINECALLSTATE_CONNECTED received
09:24:57	LINECALLINFO on 0x000102cc received
09:24:57	LINECALLINFO: 0x000102cc
	dwTotalSize=0x000001a8 dwNeededSize=0x0000015c dwUsedSize=0x0000015c
	hLine=0x00010311 dwLineDevieceID=0x00000006 dwAddressID=0x00000000
	dwTrunk=0xffffffff(-1)
	CallerID=201
	CalledID=202
	ConnectedID=201
09:25:02	LINECALLSTATE_IDLE received
09:25:03	IDLE Call Handle (0x000102cc) deallocated

Figure 6.5 Messages of Call Status

2.5 Holding Calls in Progress

If an extension asks to hold a call in progress, the TAPI-compatible applications of both the caller and called party will receive the call status events as shown in the figure below:

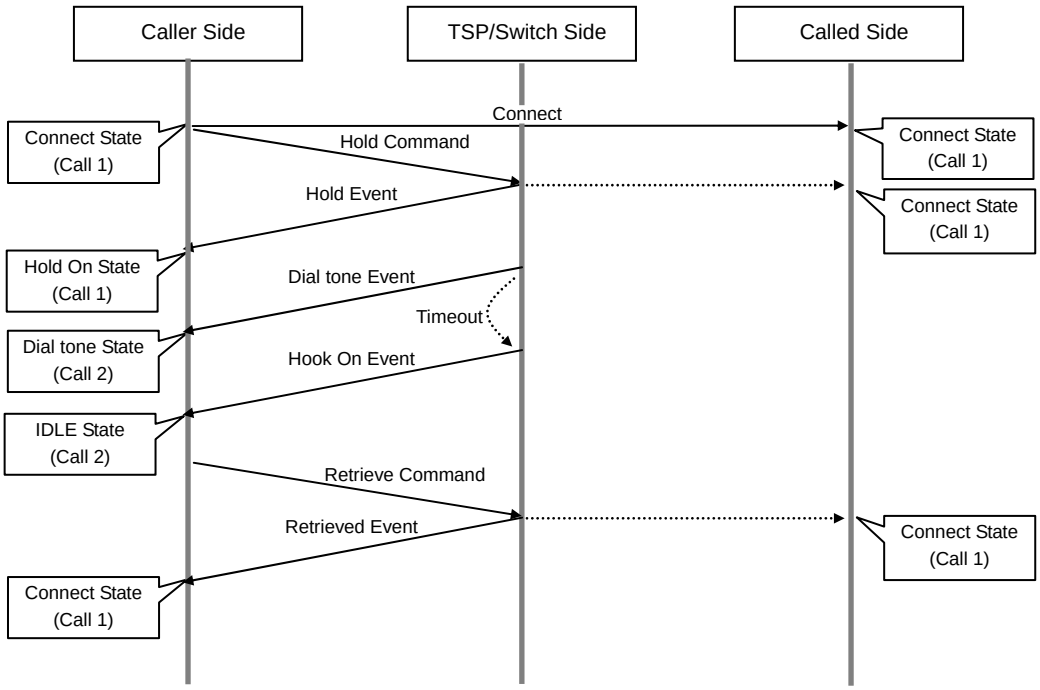


Figure 6.6 Flow Chart of Calls on Hold

The status message created when a call in progress is held is displayed on the TAPI Sampler program as shown below:

The Extension that has asked for holding a Call

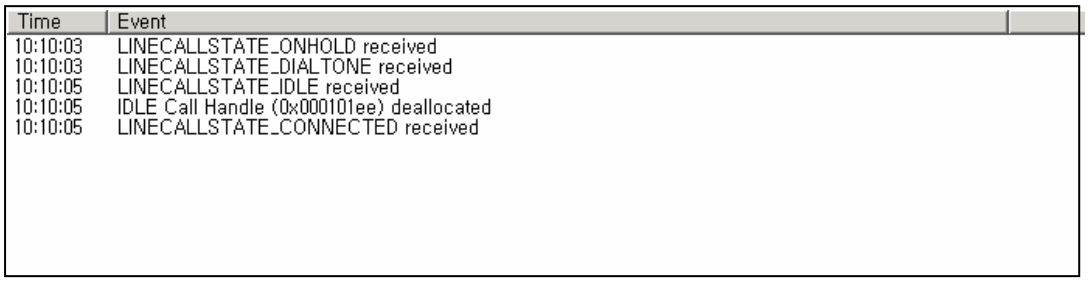
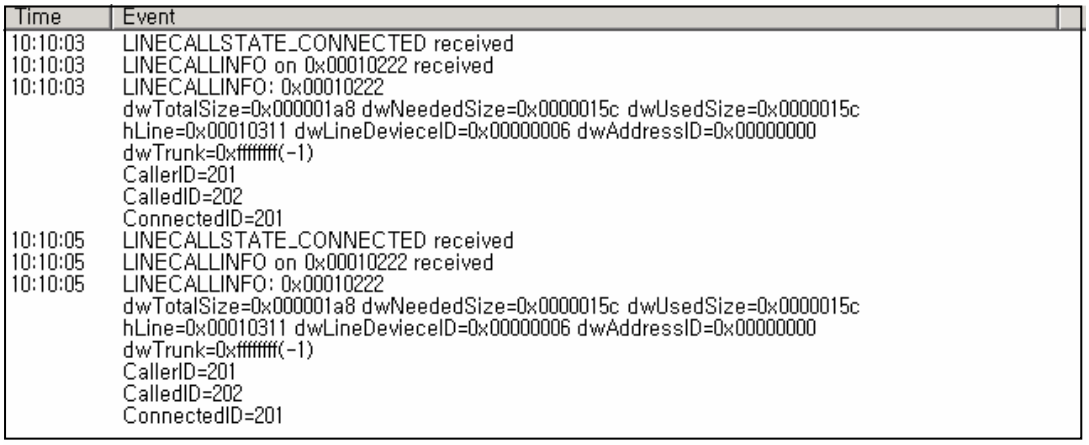


Figure 6.7 Messages of Call Status

The Extension where a Call has been Held



Time	Event
10:10:03	LINECALLSTATE_CONNECTED received
10:10:03	LINECALLINFO on 0x00010222 received
10:10:03	LINECALLINFO: 0x00010222 dwTotalSize=0x000001a8 dwNeededSize=0x0000015c dwUsedSize=0x0000015c hLine=0x00010311 dwLineDeviecelD=0x00000006 dwAddressID=0x00000000 dwTrunk=0xffffffff(-1) CallerID=201 CalledID=202 ConnectedID=201
10:10:05	LINECALLSTATE_CONNECTED received
10:10:05	LINECALLINFO on 0x00010222 received
10:10:05	LINECALLINFO: 0x00010222 dwTotalSize=0x000001a8 dwNeededSize=0x0000015c dwUsedSize=0x0000015c hLine=0x00010311 dwLineDeviecelD=0x00000006 dwAddressID=0x00000000 dwTrunk=0xffffffff(-1) CallerID=201 CalledID=202 ConnectedID=201

Figure 6.8 Messages of Call Status

2.6 Procedure for Consult Transfer

If either Extension 201 or Extension 202 forwards an extension call to another extension (203) while Extension 201 or Extension 202 is making the call, the TAPI-compatible applications of the caller, called party, and forwarded party will receive the call status events as shown in the figure below:

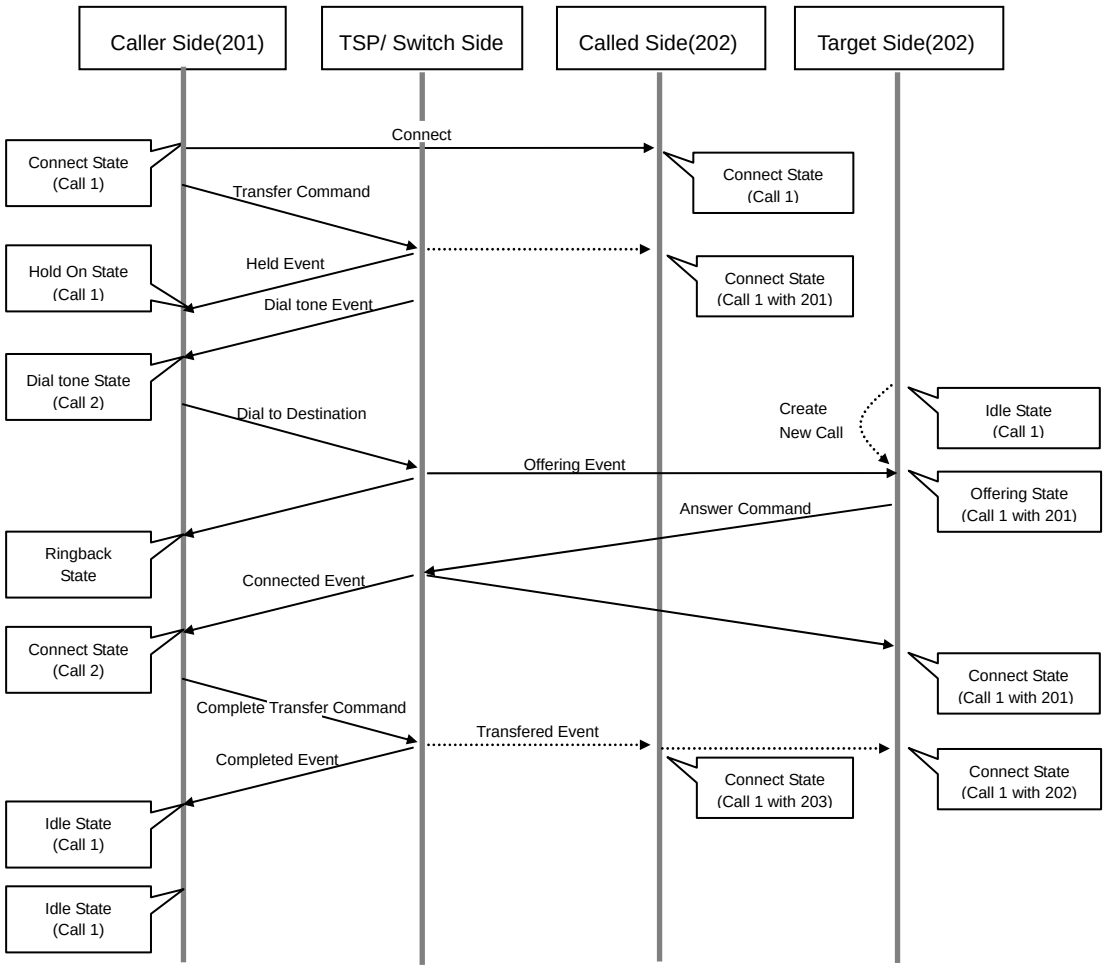


Figure 6.9 Flow Chart of Call Forwarding Status

The call status messages for the consult transfer are displayed on the TAPI Sampler program as shown below:

Extension 201

Time	Event
10:59:41	LINECALLSTATE_ONHOLDPENDTRANSFER received on 0x00010289
10:59:41	LINECALLSTATE_DIALTONE received on 0x000101ef
10:59:44	LINECALLSTATE_RINGBACK received on 0x000101ef
10:59:44	LINECALLINFO on 0x000101ef received
10:59:44	LINECALLINFO: 0x000101ef dwTotalSize=0x000001a8 dwNeededSize=0x00000154 dwUsedSize=0x00000154 hLine=0x00010044 dwLineDevieceID=0x00000005 dwAddressID=0x00000000 dwTrunk=0xffffffff(-1) CalledID=203 ConnectedID=203
10:59:46	LINECALLSTATE_CONNECTED received on 0x000101ef
10:59:46	LINECALLINFO on 0x000101ef received
10:59:46	LINECALLINFO: 0x000101ef dwTotalSize=0x000001a8 dwNeededSize=0x00000154 dwUsedSize=0x00000154 hLine=0x00010044 dwLineDevieceID=0x00000005 dwAddressID=0x00000000 dwTrunk=0xffffffff(-1) CalledID=203 ConnectedID=203
10:59:49	LINECALLSTATE_IDLE received on 0x000101ef
10:59:50	IDLE Call Handle (0x000101ef) deallocated
10:59:50	LINECALLSTATE_IDLE received on 0x00010289
10:59:50	IDLE Call Handle (0x00010289) deallocated

Figure 6.10 Call Status Messages for Consult Transfer of Extension 201

Extension 202

Time	Event
10:59:41	LINECALLSTATE_CONNECTED received on 0x00010212
10:59:41	LINECALLINFO on 0x00010212 received
10:59:41	LINECALLINFO: 0x00010212 dwTotalSize=0x000001a8 dwNeededSize=0x0000015c dwUsedSize=0x0000015c hLine=0x00010033 dwLineDevieceID=0x00000006 dwAddressID=0x00000000 dwTrunk=0xffffffff(-1) CallerID=201 CalledID=202 ConnectedID=201
10:59:49	LINECALLSTATE_CONNECTED received on 0x00010212
10:59:49	LINECALLINFO on 0x00010212 received
10:59:49	LINECALLINFO: 0x00010212 dwTotalSize=0x000001a8 dwNeededSize=0x0000016c dwUsedSize=0x0000016c hLine=0x00010033 dwLineDevieceID=0x00000006 dwAddressID=0x00000000 dwTrunk=0xffffffff(-1) CallerID=201 CalledID=202 ConnectedID=203 RedirectionID=203 RedirectingID=201

Figure 6.11 Call Status Messages for Consult Transfer of Extension 202

Extension 203

Time	Event
10:59:44	LINECALLSTATE_OFFERING received on 0x00010234
10:59:44	LINECALLINFO on 0x00010234 received
10:59:44	LINECALLINFO: 0x00010234 dwTotalSize=0x000001a8 dwNeededSize=0x0000015c dwUsedSize=0x0000015c hLine=0x00010278 dwLineDevieceID=0x00000007 dwAddressID=0x00000000 dwTrunk=0xffffffff(-1) CallerID=201 CalledID=203 ConnectedID=201
10:59:46	LINECALLSTATE_CONNECTED received on 0x00010234
10:59:46	LINECALLINFO on 0x00010234 received
10:59:46	LINECALLINFO: 0x00010234 dwTotalSize=0x000001a8 dwNeededSize=0x0000015c dwUsedSize=0x0000015c hLine=0x00010278 dwLineDevieceID=0x00000007 dwAddressID=0x00000000 dwTrunk=0xffffffff(-1) CallerID=201 CalledID=203 ConnectedID=201
10:59:49	LINECALLINFO on 0x00010234 received
10:59:49	LINECALLINFO: 0x00010234 dwTotalSize=0x000001a8 dwNeededSize=0x0000016c dwUsedSize=0x0000016c hLine=0x00010278 dwLineDevieceID=0x00000007 dwAddressID=0x00000000 dwTrunk=0xffffffff(-1) CallerID=202 CalledID=203 ConnectedID=202 RedirectionID=203 RedirectingID=201

Figure 6.12 Call Status Messages for Consult Transfer of Extension 203

NOTE

Type of Call Processing Messages in the TAPI Sampler Program

The messages of call progress such as Blind Transfer, Conference, PickUp, and Redirect can be checked from the TAPI Sampler program.

This page is intentionally left blank.

ABBREVIATION

A

API	Application Program Interface
ASP	Abstract Service Primitive

C

CD	Compact Disk
CTI	Computer Telephony Interface

D

DCS	Digital Cellular System
DND	Do Not Disturb
DNIS	Dialed Number Identification Service
DTMF	Dual Tone Multi-Frequency

H

HTTP	Hypertext Transfer Protocol

I

ID	Identification
iDCS	internet Digital Cellular System
IP	Internet Protocol

O

OHVA	Off-hook Voice Announce

P

PIDs	Process Identifications

R

ROM	Read Only Memory

T

TAPI	Telephony Application Programming Interface
TCP	Transmission Control Protocol
TSP	Telephony Service Provider
TSPI	Telephony Service Provider Interface

OfficeServ

OpenTSP Driver Description

Visit us at

http://www.samsungnetwork.com

SAMSUNG ELECTRONICS