CONVERSANT® System

Version 8.0

Communication Development

585-313-220
Issue 3
Comcode 108850272
December 2001

Copyright and Legal Notices

Acknowledgment

This document was prepared by the CRM development group of Avaya University. Offices are located in Denver CO, Columbus OH, Middletown NJ, and Basking Ridge NJ, USA.

Trademarks

Avaya has made every effort to supply the following trademark information about company names, products, and services mentioned in the CONVERSANT documentation library:

- Adobe Systems, Inc. — Trademarks: Adobe, Acrobat.
- AT&T — Registered trademarks: Truevoice.
- Avaya, Inc. — Registered trademarks: AUDIX, CONVERSANT, CentreVu, DEFINITY, Voice Power. Trademarks: FlexWord, Intuity, Avaya.
- CLEO Communications — Trademarks: LINKix.
- Hayes Microcomputer Products, Inc. — Trademarks: Hayes, Smartmodem.
- Intel Corporation — Registered trademarks: Pentium.
- Interface Systems, Inc. — Trademarks: CLEO.
- International Business Machines Corporation — Registered trademarks: IBM, VTAM.
- Lucent Technologies — Registered trademark: 5ESS. Trademark: Lucent.
- Microsoft Corporation — Registered trademarks: Excel, Internet Explorer, Microsoft, MS, MS-DOS, Windows, Windows NT.
- Minnesota Mining and Manufacturing — Trademarks: 3M.
- Netscape Communications — Trademarks: Netscape Navigator.
- Novell, Inc. — Registered trademarks: Novell.
- Oracle Corporation — Trademarks: OBJECT*SQL, ORACLE, ORACLE*Terminal, PRO*C, SQL*FORMS, SQL*Menu, SQL*Net, SQL*Plus, SQL*ReportWriter.
- Phillips Screw Co. — Registered trademarks: Phillips.
- Santa Cruz Operation, Inc. — Registered trademarks: UnixWare.

- UNIX System Laboratories, Inc. — Registered trademarks: UNIX.
- Xerox Corporation — Trademarks: Ethernet.

Limited Warranty

Avaya provides a limited warranty on this product. Refer to the “Limited Use Software License Agreement” card provided with your package.

Avaya has determined that use of this electronic data delivery system cannot cause harm to an end user's computing system and will not assume any responsibility for problems that may arise with a user's computer system while accessing the data in these document.

Every effort has been made to make sure that this document is complete and accurate at the time of release, but information is subject to change.

Product Safety Standards

This product complies with and conforms to the following international Product Safety standards as applicable:

- Safety of Information Technology Equipment, IEC 60950, 3rd Edition, including all relevant national deviations as listed in Compliance with IEC for Electrical Equipment (IECEE) CB-96A.
- Safety of Information Technology Equipment, CAN/CSA-C22.2 No. 60950-00 / UL 60950, 3rd Edition
- Safety Requirements for Customer Equipment, ACA Technical Standard (TS) 001 - 1997
- One or more of the following Mexican national standards, as.applicable: NOM 001 SCFI 1993, NOM SCFI 016 1993, NOM019 SCFI 1998.

United States FCC Compliance Information

Part 15: Class A statement. This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio-frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case the user will be required to correct the interference at his own expense.

Electromagnetic Compatibility (EMC) Standards

This product complies with and conforms to the following international EMC standards and all relevant national deviations:

- Limits and Methods of Measurement of Radio Interference of Information Technology Equipment, CISPR 22:1997 and EN55022:1998.
- Information Technology Equipment - Immunity Characteristics - Limits and Methods of Measurement, CISPR 24:1997 and EN55024:1998, including:
  - ~ Electrostatic Discharge (ESD) IEC 61000-4-2
  - ~ Radiated Immunity IEC 61000-4-3
  - ~ Electrical Fast Transient IEC 61000-4-4
  - ~ Lightning Effects IEC 61000-4-5
  - ~ Conducted Immunity IEC 61000-4-6
  - ~ Mains Frequency Magnetic Field IEC 61000-4-8
  - ~ Voltage Dips and Variations IEC 61000-4-11
  - ~ Powerline Harmonics IEC 61000-3-2
  - ~ Voltage Fluctuations and Flicker IEC 61000-3-3

Canadian Interference Information

This Class A digital apparatus complies with Canadian ICES-003.

Cet appareil numérique de la classe A est conforme à la norme NMB-003 du Canada.

European Union Declaration of Conformity

Avaya Inc. declares that the equipment specified in this document bearing the "CE" (Conformité Europeénne) mark conforms to the European Union Radio and Telecommunications Terminal Equipment Directive (1999/5/EC), including the Electromagnetic Compatibility Directive (89/336/EEC) and Low Voltage Directive (73/23/EEC). This equipment has been certified to meet and CTR4 Primary Rate Interface (PRI) and subsets thereof in CTR12 and CTR13, as applicable.

Copies of the Declaration of Conformity (DoC) can be obtained by contacting your local sales representative and are available on the following Web site:

http://support.avaya.com/elmodocs2/DoC/IDoC/index.jhtml

Telecom New Zealand Ltd Warning Notices

GENERAL WARNING: The grant of a Telepermit for any item of terminal equipment indicates that only Telecom has accepted that the item complies with minimum conditions for connection to its network. It indicates no endorsement of the product by Telecom, nor does it provide any sort of warranty. Above all, it provides no assurance that any item will work correctly in all respects with other items of Telepermitted equipment of a different make or model, nor does it imply that any product is compatible with all of Telecom’s network services.

IMPORTANT NOTICE: Under power failure conditions, this device may not operate. Please ensure that a separate telephone, not dependent on local power, is available for emergency use.

AUTOMATIC RE-ATTEMPTS TO THE SAME NUMBER: Some parameters required for compliance with Telecom’s Telepermit requirements are dependent on the equipment (PC) associated with this device. The associated equipment shall be set to operate within the following limits for compliance with Telecom specifications:

- There shall be no more than 10 call attempts to the same number within any 30 minute period for any single manual call initiation, and,
- The equipment shall go on-hook for a period of not less than 30 seconds between the end of one attempts and the beginning of the next attempt.

AUTOMATIC CALLS TO DIFFERENT NUMBERS: Some parameters required for compliance with Telecom’s Telepermit requirements are dependent on the equipment (PC) associated with this device. In order to operate within the limits for compliance with Telecom specifications, the associated equipment shall be set to ensure that automatic calls to different numbers are spaced such that there is not less than 5 seconds between the end of one call attempt and the beginning of the next attempt.

USER INSTRUCTIONS (AUTOMATIC CALL SETUP): This equipment shall not be set up to make automatic calls to the Telecom "111" emergency service.

CALL ANSWERING (AUTOMATIC ANSWERING EQUIPMENT): Some parameters required for compliance with Telecom’s Telepermit requirements are dependent on the equipment (PC) associated with this device. In order to operate within the limits for compliance with Telecom specifications, the associated equipment shall be set to ensure that calls are answered between 3 and 30 seconds of receipt of ringing.

Japan

This is a Class A product based on the standard of the Voluntary Control Council for Interference by Information Technology Equipment (VCCI). If this equipment is used in a domestic environment, radio disturbance may occur, in which case, the user may be required to take corrective actions.

この装置は、情報処理装置等電波障害自主規制協議会（ＶＣＣＩ）の基準に基づくクラスＡ情報技術装置です。この装置を家庭環境で使用すると電波妨害を引き起こすことがあります。この場合には使用者が適切な対策を講ずるよう要求されることがあります。

Preventing Toll Fraud

“Toll fraud” is the unauthorized use of your telecommunications system by an unauthorized party (for example, a person who is not a corporate employee, agent, subcontractor, or working on your company’s behalf). Be aware that there may be a risk of toll fraud associated with your system and that, if toll fraud occurs, it can result in substantial additional charges for your telecommunications services.

Avaya Fraud Intervention:

If you suspect that you are being victimized by toll fraud and you need technical assistance or support, call the Technical Service Center’s Toll Fraud Intervention Hotline at 1-800-643-2353.

Providing Telecommunications Security

Telecommunications security (of voice, data, and/or video communications) is the prevention of any type of intrusion to (that is, either unauthorized or malicious access to or use of your company's telecommunications equipment) by some party.

Your company's "telecommunications equipment" includes both this Avaya product and any other voice/data/video equipment that could be accessed via this Avaya product (that is, "networked equipment").

An "outside party" is anyone who is not a corporate employee, agent, subcontractor, or working on your company's behalf. Whereas, a "malicious party" is anyone (including someone who may be otherwise authorized) who accesses your telecommunications equipment with either malicious or mischievous intent.

Such intrusions may be either to/through synchronous (time-multiplexed and/or circuit-based) or asynchronous (character-, message-, or packet-based) equipment or interfaces for reasons of:

- Utilization (of capabilities special to the accessed equipment)
- Theft (such as, of intellectual property, financial assets, or toll-facility access)
- Eavesdropping (privacy invasions to humans)
- Mischief (troubling, but apparently innocuous, tampering)
- Harm (such as harmful tampering, data loss or alteration, regardless of motive or intent)

Be aware that there may be a risk of unauthorized intrusions associated with your system and/or its networked equipment. Also realize that, if such an intrusion should occur, it could result in a variety of losses to your company (including but not limited to, human/data privacy, intellectual property, material assets, financial resources, labor costs, and/or legal costs).

Your Responsibility for Your Company's Telecommunications Security

The final responsibility for securing both this system and its networked equipment rests with you – an Avaya customer's system administrator, your telecommunications peers, and your managers. Base the fulfillment of your responsibility on acquired knowledge and resources from a variety of sources including but not limited to:

- Installation documents
- System administration documents
- Security documents
- Hardware-/software-based security tools
- Shared information between you and your peers
- Telecommunications security experts

To prevent intrusions to your telecommunications equipment, you and your peers should carefully program and configure your:

- Avaya-provided telecommunications systems and their interfaces
- Avaya-provided software applications, as well as their underlying hardware/software platforms and interfaces
- Any other equipment networked to your Avaya products

Avaya does not warrant that this product or any of its networked equipment is either immune from or will prevent either unauthorized or malicious intrusions. Avaya will not be responsible for any charges, losses, or damages that result from such intrusions.

Documentation Ordering Information

To order a document, specify the 9-digit document number, the issue number, and the issue date.

Write, Call, or Fax

Voice	1 800 457-1235	International Voice	317 322-6791
FAX	1 800 457-1764	International FAX	317 322-6699

Write: Globalware Solutions
200 Ward Hill Avenue
Haverhill, MA 01835 USA
email: totalware@gwsmail.com

World Wide Web

Use a web browser to order documents:

http://support.avaya.com/elmodocs2/conversant/index.jhtml

Standing Orders

You can be placed on a standing order list for this and other documents you may need. Standing order will enable you to automatically receive updated versions of individual documents or document sets, billed to account information that you provide. For more information on standing orders, or to be put on a list to receive future issues of this document, call or write the Avaya Publications Center (see Write, Call, or Fax on page viii).

Contents

About This Book

Overview

This book is a reference manual for creating the necessary platform environment and applications to implement various communication interfaces between callers, administrators, and the CONVERSANT system.

How This Book Is Organized

This book is organized into the following sections:

- Chapter 1, Analog Telephony Interfaces — Describes the use of analog telephony as a communication arrangement, as well as the provisioning required to implement this interface. This includes the suggested administrative values to set on the system.

Note: Chapter 1 does not apply to the UCS 1000 as it is a digital only offer.

- Chapter 2, Digital Telephony Interfaces — Describes the use of digital telephony as a communication arrangement, as well as the provisioning required to implement this interface. This includes the suggested administrative values to set on the system.
- Chapter 3, Adjunct/Switch Application Interface — Describes the use of the Adjunct/Switch Application Interface (ASAI) as a communication arrangement, as well as the provisioning required to implement this interface. This includes the suggested analog or digital administrative values to set on the system.
- Chapter 5, Converse Vector Step Routing — Describes the use of the Converse Vector Step (CVS) routing as a communication arrangement, as well as the provisioning required to implement this interface. This includes the suggested administrative values to set on the system.
- Chapter 6, Call Classification Analysis — Describes the potential use and benefits of Call Classification Analysis (CCA) within analog and digital communication arrangements, as well as the provisioning required to implement this interface. This includes the suggested administrative values to set on the system.
- Chapter 7, Data Network Communications — Describes the potential uses of data network communications, discusses physical and logical protocol differences, and details what you must do on the system to implement this type of communication.
- Appendix A, Transmission Level Adjustment — Describes how to ensure that all speech heard by a caller is at a level that is appropriate for listening without causing oscillations or distortions in the network.
- Glossary — Defines the terms, abbreviations, and acronyms used in system documentation.
- Index — Alphabetically lists the principal subjects covered in the book.

How to Use This Book

Read Chapter 1, Analog Telephony Interfaces through Chapter 6, Call Classification Analysis, to learn more about the telephony interfaces used by the caller accessing the CONVERSANT system. Each of these chapters contains examples of how communication between the system and an external network is established. These examples are *not* the only methods of gaining this access, as actual network cabling varies on a site-by-site basis. These chapters also provide examples of using various features in an application whether it was developed using Script Builder, transaction state machine (TSM) script language, or the Intuity Response Application Programming Interface (IRAPI). Chapter 7, Data Network Communications, describes the following data network interfaces:

- TN3270E
- TCP/IP
- SQL*NET
- Physical asynchronous connections to the CONVERSANT platforms

Conventions Used in This Book

Understanding the typographical and other conventions used in this book is necessary to interpret the information.

Terminology

- The word "type" means to press the key or sequence of keys specified. For example, an instruction to type the letter "y" is shown as

  Type **y** to continue.

- The word "enter" means to type a value and then press the **ENTER** key on the keyboard. For example, an instruction to type the letter "y" and press **ENTER** is shown as

  Enter **y** to continue.

- The word "select" means to move the cursor to the desired item and then press **ENTER**. For example, an instruction to move the cursor to the start test option on the Network Loop-Around Test screen and then press **ENTER** is shown as

  Select **Start Test**.

- The system displays menus, screens, and windows. Menus allow you to select options or to choose to view another menu, screen, or window (Figure 1 on page xv). Screens and windows both show and request system information (Figure 2 on page xv through Figure 5 on page xvi).

Note: Screens shown in this book are examples only. The screens you see on your machine will be similar, but not exactly the same.

Figure 1. Example of a CONVERSANT Menu

```
      Voice System Administration

 Application Package Administration
 Configuration Management
 Feature Packages
 Reports
 Script Builder Applications
 Switch Interfaces
 System Monitor
 Exit
```

Figure 2. Example of a CONVERSANT Window Requesting Information

```
                         Replace Disk


Enter the logical name of disk:



Enter jumper id of the disk being replaced (0-7): _
```

Figure 3. Example of a CONVERSANT Window Showing Information

```
 6                         Define User Password Information

The password has been defined as follows:

   jd  PS  08/08/96  0  24      1
```

Figure 4. Example of a CONVERSANT Screen Showing Information

```
In order to install UnixWare, you must reserve a partition (a
portion of your hard disk's space) on your primary hard disk
for the UNIX System.After you press 'ENTER' you will be shown
a screen that will allow you to create new partitions, delete
existing partitions or change the active partition of your
primary hard disk (the partition that your computer will boot
from).

WARNING: All files in any partition(s) you delete will be
destroyed. If you wish to attempt to preserve any files from
an existing UNIX System, do not delete its partition(s).

The UNIX System partition that you intend to use on the
primary hard disk must be at least 120 MBs and labeled
'ACTIVE.'
```

Figure 5. Example of a CONVERSANT Screen Requesting Information

```
You may use a partition of your secondary hard disk.  If you
choose to use a partition of your secondary hard disk you
will be shown a screen that will allow you to partition your
secondary hard disk.

WARNING: All files in any partition(s) you delete will be
destroyed.

If you choose to create a UNIX System partition on your
secondary hard disk, it must be at least 40 MBs.

Your Options are:

1.  Do not use a partition of the secondary hard disk for
    the UNIX System.
2.  Use a partition of the secondary hard disk for the
    UNIX System.

Press '1' or '2' followed by 'ENTER'.
```

Keyboard and Telephone Keypad Representations

- Keys that you press on your terminal or PC are represented as small capitalized **BOLD** text. For example, an instruction to press the enter key is shown as

  Press **ENTER**.

- Two or three keys that you press at the same time on your terminal or PC (that is, you hold down the first key while pressing the second and/or third key) are represented in small capitalized **BOLD** text. For example, an instruction to press and hold the Alt key while typing the letter "d" is shown as

  Press **ALT + D**.

- Function keys on your terminal, PC, or system screens, also known as soft keys, are represented as small capitalized **BOLD** text followed by the function or value of that key enclosed in parentheses. For example, an instruction to press function key 3 is shown as

  Press **F3** (Choices).

- Keys that you press on your telephone keypad appear in small capitalized **BOLD** text. For example, an instruction to press the first key on your telephone keypad is shown as

  Press **1** to record a message.

Cross References and Hypertext

Blue underlined type indicates a cross reference or hypertext link that takes you to another location in the document when you click on it with your mouse.

Screen Displays

- Values, system messages, field names, prompts that appear on the screen, and simulated screen displays are shown in typewriter-style `constant width` type, as in the following examples:

  Enter the number of ports to be dedicated to outbound traffic in the `Maximum Simultaneous Ports` field.

  ```
  Alarm Form Update was successful.
  Press <Enter> to continue.
  ```

- The sequence of menu options that you must select to display a specific screen or submenu is shown as follows:

  Start at the Voice Administration menu and select:

  ```
  Configuration Management
      Database Administration
  ```

  In this example, you would access the CONVERSANT Voice Administration menu and select the Configuration Management menu. From the Configuration Management menu, you would then select the Database Administration option.

Other Typography

- Commands and text you type in or enter appear in **bold type**, as in the following examples:

  Enter **change-switch-time-zone** at the `Enter command:` prompt.

  Type **high** or **low** in the `Speed:` field.

- Command variables are shown in ***bold italic*** type when they are part of what you must type in, and in *blue italic type* when they are referred to, for example:

  Enter **ch ma *machine_name***, where *machine_name* is the name of the call delivery machine you just created.

- Command options are shown inside square brackets, for example:

  Enter **connect *switchname* [-d] [-b | -w]**

Safety and Security Alert Labels

This book uses the following symbols to call your attention to potential problems that could cause personal injury, damage to equipment, loss of data, service interruptions, or breaches of toll fraud security:

CAUTION:
Indicates the presence of a hazard that if not avoided can or will cause minor personal injury or property damage, including loss of data.

WARNING:
Indicates the presence of a hazard that if not avoided can cause death or severe personal injury.

DANGER:
Indicates the presence of a hazard that if not avoided will cause death or severe personal injury.

SECURITY ALERT:
Indicates the presence of a toll fraud security hazard. Toll fraud is the unauthorized use of a telecommunications system by an unauthorized party.

Getting Help

The CONVERSANT system provides online help to assist you during installation, administration, and application development tasks.

To use the online help:

- Press **F1** (Help) when you are in a menu or window.

  The first time you press **F1**, the system displays information about the currently active window or menu.

  ~ When you are in a window, the help explains the purpose of the window and describes its fields.

  ~ When you are in a menu, the help explains how to use menus.

  If you press **F1** again, the system displays a General Help screen that explains how to use the online help.

- Press **F2** (Choices) when you are in a field.

  The system displays valid field choices either in a pop-up window or on the status line directly above the function keys.

- Press **F6** (Cancel) to exit the online help.

Technical Assistance

Web Site

The following customer support web site contains resources where you can find solutions for technical problems:

http://support.avaya.com

Contact Numbers

Technical assistance on the CONVERSANT product is available through the following telephone contacts:

- In the United States, call 1-800-242-2121.
- In Canada, call one of the following numbers, depending on your location:
  - 1-800-363-1882 for assistance in Quebec and eastern Canada
  - 1-800-387-4268 for assistance in Ontario and western Canada
- In any other country, call your local distributor or check with your project manager or systems consultant.

Related Resources

Additional documentation and training material is available for you to learn more about the CONVERSANT product.

Training

To obtain training on the CONVERSANT product, contact Avaya University at one of the following numbers:

- Organizations within Avaya (904) 636-3261
- Avaya customers and all others (800) 255-8988

You can also view information on CONVERSANT training at the Avaya University web site at the following web link:

http://learning2.avaya.com

Click on *TRAINING SEARCH*, *Search Catalog by Keyword*, and search for CONVERSANT.

The courses listed below are recommended. Other courses are available.

- For technicians doing repairs on CONVERSANT systems
  - BTE501W, CONVERSANT Administration for Technicians
  - BTE502H, CONVERSANT Installation and Maintenance
- For technicians and administrators
  - BTC346M, CONVERSANT Administration Overview (CD-ROM)

- For application developers

Note: Courses listed below are instructor-led unless otherwise specified.

  - ~ BTC128H, Introduction to Script Builder
  - ~ BTC166H, Introduction to Voice@Work
  - ~ BTC204H, Intermediate Voice@Work
  - ~ BTC204W, Intermediate Voice@Work, interactive distance learning, using Bit-Room technology
  - ~ BTC301H, Advanced CONVERSANT Programming

Documentation

Appendix A, "Documentation Guide," in *CONVERSANT System Version 8.0 System Description*, 585-313-219, describes in detail all books included in CONVERSANT documentation library and referenced in this book.

Note: Always refer to the appropriate book for specific information on planning, installing, administering, or maintaining a CONVERSANT system.

Additional Suggested Documentation

It is suggested that you also obtain and use the following book for information on security and toll fraud issues:

- *GBCS Products Security Handbook*, 555-025-600

Obtaining Printed Versions of the Documentation

See Documentation Ordering Information on page viii of Copyright and Legal Notices for information on how to purchase CONVERSANT documentation in printed form. You can also print documentation locally from the CD-ROM (see Printing the Documentation on page xxi).

Using the CD-ROM Documentation

Avaya ships the documentation in electronic form. Using the Adobe Acrobat Reader application, you can read these documents on a Windows PC, on a Sun Solaris workstation, or on an HP-UX workstation. Acrobat Reader displays high-quality, print-like graphics on both UNIX and Windows platforms. It provides scrolling, zoom, and extensive search capabilities, along with online help. A copy of Acrobat Reader is included with the documents.

Note: When viewing documents online, it is recommended that you use a separate platform and not the CONVERSANT system.

Setting the Default Magnification

You can set your default magnification by selecting **File | Preferences | General**. We recommend the **Fit Page** option.

Adjusting the Window Size

On HP and Sun workstations, you can control the size of the reader window by using the **-geometry** argument. For example, the command string **acroread -geometry 900x900 mainmenu.pdf** opens the main menu with a window size of 900 pixels square.

Hiding and Displaying Bookmarks

By default, the document appears with bookmarks displayed on the left side of the screen. The bookmarks serve as a hypertext table of contents for the chapter you are viewing. You can control the appearance of bookmarks by selecting **View | Page Only** or **View | Bookmarks and Page.**

Using the Button Bar

The button bar can take you to the book's Index, table of contents, main menu, and glossary. It also lets you update your documents. Click the corresponding button to jump to the section you want to read.

Using Hypertext Links

Hypertext links appear in blue underlined text. These links are shortcuts to other sections or books.

Navigating with Double Arrow Keys

The double right and double left arrows (and) at the top of the Acrobat Reader window are the go-back and go-forward functions. The go-back button takes you to the last page you visited prior to the current page. Typically, you use to jump back to the main text from a cross reference or illustration.

Searching for Topics

Acrobat has a sophisticated search capability. From the main menu, select **Tools | Search.** Then select **Master Index.**

Displaying Figures

If lines in figures appear broken or absent, increase the magnification. You might also want to print a paper copy of the figure for better resolution.

Printing the Documentation

Note: For information on purchasing printed copies of the documents, see Obtaining Printed Versions of the Documentation on page xx.

If you would like to read the documentation in paper form rather than on a computer monitor, you can print all or portions of the online screens.

Printing an Entire Document

To print an entire document, do the following:

1. From the documentation main menu screen, select one of the print-optimized documents. Print-optimized documents print two screens to a side, both sides of the sheet on 8.5x11-inch or A4 paper.
2. Select **File | Print**.
3. Enter the page range you want to print, or select **All**. Note that the print page range is different from the page numbers on the documents (they print two to a page).

   The document prints.
4. Close the file. Do not leave this file open while viewing the electronic documents.

Printing Part of a Document

To print a single page or a short section, you can print directly from the online version of the document.

1. Select **File | Print**.
2. Enter the page range you want to print, or select **Current**.

   The document prints, one screen per side, two sides per sheet.

How To Comment on This Book

While we have tried to make this document fit your needs, we are interested in your suggestions for improving it and urge you to send your comments to us.

Comment Form

A comment form, available in paper and electronic versions, is available via the documentation CD-ROM. To use the comment form:

1. Select **Comments** from the Main Menu of the CD-ROM.
2. Follow the instructions provided on the CD-ROM to do one of the following:
   - Print the paper version of the form, complete it, and either fax or mail it to us.
   - Access a Avaya website where you can enter your comments electronically.

1 Analog Telephony Interfaces

Overview

Note: Analog communication is not available on the UCS 1000 platform. See Chapter 2, Digital Telephony Interfaces, for information on the T1, Line Side or Foreign eXchange Station (FXS), and Primary Rate Interface (PRI) digital telephony interfaces.

This chapter describes the tip/ring and FAX analog telephony interfaces available with the CONVERSANT system's base and optional software and the requirements that must be met to implement these interfaces.

This chapter also provides an overview of analog communications and examples of typical analog connections.

Introduction to Analog Communications

In its analog configuration, the system provides nearly universal connectivity to existing private branch exchange (PBX) and automatic call distribution (ACD) customer-premise equipment. It also allows standard interfaces to such widespread network services as Public Switched Telephone Networks (PSTNs) and Centrex service.

The following base analog telephony features make the CONVERSANT system compatible with a variety of domestic PBXs or ACDs (including the Avaya DEFINITY Communications Systems Generic 1, 2, and 3, System 85, System 75, System 25, DIMENSION 2000, and so on):

- The system can perform switch-hook-flash transfers (also known as *register recall*) using the functions of the PBX or ACD or Centrex service. It can also determine if the extension to which the call was transferred is busy or there is no answer and whether an alternative message or action should occur.
- In addition to switch-hook-flash transfers, the system supports transfer with a call bridge connection through the system. This bridging can be done with both digital and analog connections. Table 1 on page 2 lists the analog line capabilities supported by call bridging.

- The system is capable of far-end caller disconnect detection through "wink signal" detection or such alternatives as call progress tone detection, for instance, dial tone, busy tone, or reorder tone detection. (The wink signal is a momentary break in loop current, typically of a 600-millisecond duration.) Because these capabilities allow the system to know when a caller hangs up, the system rarely transfers a "ghost" call, but instead allows the voice script to terminate quickly and be ready for the next call.

Note: Far-end caller disconnect detection through a wink signal or a call progress tone must also be supported by the PBX or ACD. Avaya DEFINITY Communications Systems Generic 1, 2, and 3, System 85, System 75, System 25, and DIMENSION 2000 switches provide the signaling needed to notify the CONVERSANT system of far-end caller disconnect. Other PBX systems may not. In these cases, implement script timeouts to ensure script termination.

- Outdialing for call transfer can be done with either touchtone or dial pulse (sometimes called decadic dialing or loop disconnect signaling).
- With custom software, the system can be programmed to transfer calls using dial access codes (rather than switch-hook-flash) to support PBXs that use this method of call transfer.

Trainable dial tone, software-settable switch hook flash duration, and wink signal duration also add to the system's flexibility.

Table 1 details the maximum number of analog and digital lines supported with call bridging. Table 2 details the maximum number of analog and digital lines supported without call bridging.

Table 1. Maximum Digital Trunks/Analog Lines Supported by Call Bridge

MAP/40P	
Answer/Originate	**Outbound/Bridging**
24 digital T1 (linked)	24 digital T1
24 analog tip/ring	24 analog tip/ring
24 analog tip/ring	24 digital T1
24 digital T1	24 analog tip/ring

Table 2. Maximum Digital Trunks/Analog Lines Supported without Call Bridge

	MAP/40P
	Answer/Originate
Analog	48
Digital	48

Analog Connections to a 5ESS Switch

Analog lines from the local service provider supply the physical interface between the switch and the CONVERSANT system. The lines should be configured as a standard 2500 analog set on the switch. See Chapter 5, "Switch Interface Administration," of *CONVERSANT System Version 8.0 Administration*, 585-313-510, for an extended list of tunable parameters available with the various switch integration packages.

Analog Connections to Avaya PBXs

Analog connections between a CONVERSANT system and a PBX can be made to accommodate the needs for basic system connectivity. They can also be made to support optional feature packages that can make use of analog connections, such as the Adjunct/Switch Application Interface (ASAI).

The following settings and configuration data must be present on the PBX for analog tip/ring communication between the PBX and the CONVERSANT system. The CONVERSANT system is designed to accommodate switch integration with Avaya System 75/DEFINITY switches as a default. Integration with other PBXs may require that you set specific switch integration values through the Voice System Administration menu. See Chapter 5, "Switch Interface Administration," of *CONVERSANT System Version 8.0 Administration*, 585-313-510, for an extended list of tunable parameters and valid values for various PBXs.

- The domestic PBX must provide analog service using CCITT (International Telephone and Telegraph Consultative Committee) and LSSGR (LATA Switching Systems Generic Requirements) standards. All analog station packs on DEFINITY switches and DIMENSION 2000 meet these standards. However, the LC03 circuit card on the DIMENSION 2000 and the SN229 circuit card on the System 85/G2 are *not* recommended for connection to the CONVERSANT system.
- Each analog port on the switch must be configured to communicate as a standard 2500 analog set with the ability to transfer and conference calls. Each port requires a station number, an appropriate Class of Service (COS)/Class of Restriction (COR), and a hardware port location.

Note: On DEFINITY G1/G3 switches, ports routed to the Intuity CONVERSANT system must not have data restrictions in the COR, and "redirect notification" must be set to "y" if the CONVERSANT system is to transfer calls to ACD splits staffed by Auto Answer (zip tone) agents.

- The station numbers assigned to CONVERSANT system ports must be valid entries in the system dial plan.

- If you are using a MERLIN LEGEND communications system:
  - ~ All analog trunks receiving calls from and getting calls for the CONVERSANT system must provide reliable disconnect.
  - ~ All tip/ring lines originating from the MERLIN LEGEND switch connected to the CONVERSANT system must be set up in a MERLIN LEGEND calling group as type “Generic VMI.”
  - ~ You must administer the lines connected to the system with “outside line” dial tone. See "Inside Dial Tone” in the *MERLIN LEGEND Communications System Installation, Programming, and Maintenance* book that is applicable to your system for more information.

Analog Connections to Other Switches

The CONVERSANT system can interface with other switches if differences in communication protocols and parameter settings are taken into account. The proper setting of these parameters on both the switch and the CONVERSANT system is essential for establishing communications between the two devices. See Chapter 5, “Switch Interface Administration,” of *CONVERSANT System Version 8.0 Administration*, 585-313-510, for an extended list of tunable parameters. For specific values for each parameter, contact your local technical support organization.

Tip/Ring Interface

The tip/ring interface is provided through an analog (loop-start) tip/ring circuit card, with multiple 2-wire interfaces to the PBX, ACD, central office (CO), or foreign PSTN services. In addition to providing a physical network interface, the tip/ring circuit card provides speech encoding and playback, dual tone multifrequency (DTMF) recognition, call supervision, and intraswitch call classification for intelligent transfers. See Introduction to Analog Communications on page 1 for more information.

Tip/Ring Connectivity

Figure 6 through Figure 8 on page 6 show typical tip/ring connections from the CONVERSANT system. See the “Installing or Replacing Circuit Cards” chapter in the MAP/40P maintenance book for information on installing a tip/ring circuit card.

Note: The connectivity diagrams provide examples of tip/ring connections and are not the only methods of gaining connectivity to an external network. Actual network cabling varies by site, and the cabling techniques used in each installation are the responsibility of the system administrator or installation technician.

Figure 6. Analog Tip/Ring Interface Connection to a DSX Patch Panel

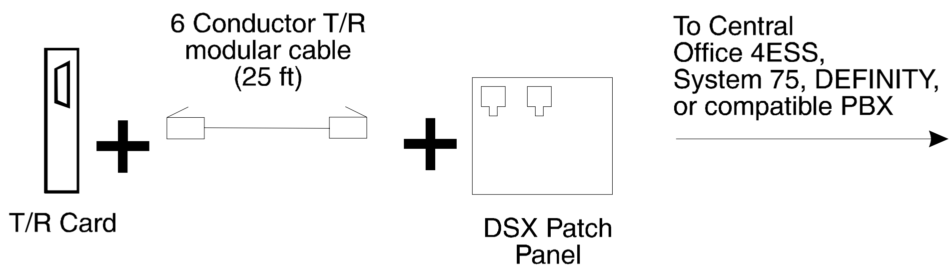
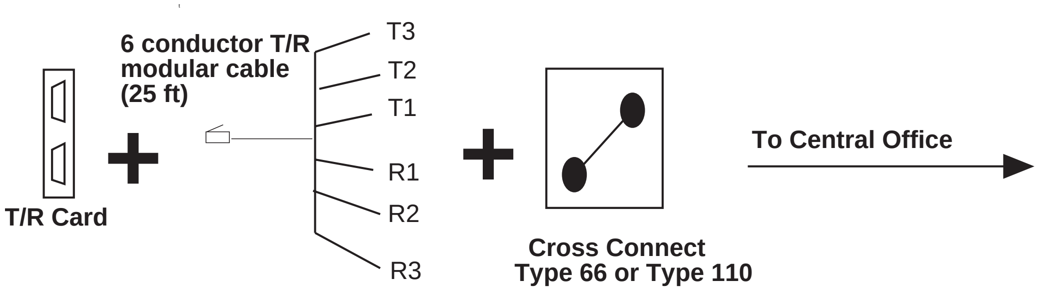
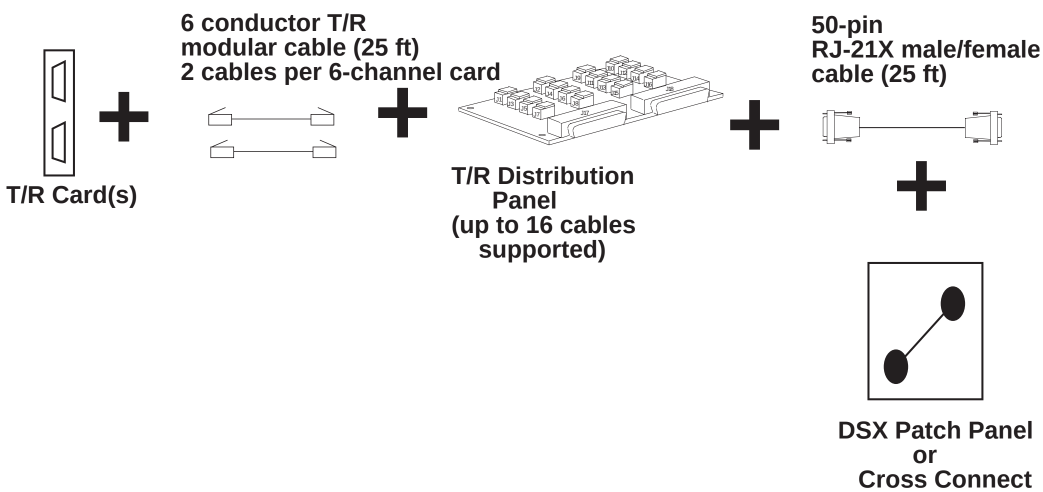


Figure 7. Analog Tip/Ring Interface Connection to a Type 66 or 110 Cross-Connect



The AYC30 circuit card has 8-pin jacks where the outside pins are available for a rarely used “Earth Recall” feature.

Figure 8. Analog Tip/Ring Interface Connection from Distribution Panel Using RJ21X Cable



Tip/Ring Telephony Interface Specifications

Tables from Table 3 through Table 6 on page 8 detail the various tip/ring telephony interface specifications.

Table 3. Tip/Ring Circuit Card General Specifications

Attribute	Value
Type of service	Loop-start POTS
Loop current detection	15 mA minimum
Ringing voltage detection	88 VRMS at 20 Hz (nominal)
Ringer equivalence for tip/ring	1.0 B for AYC10
Wink detection[1]	80–800 msec
Flash duration[1]	40–1550 msec
Register recall[1]	Timed break
Answer delay[1]	0–10 rings

[1] These attributes are adjustable through the Analog Switch Interface (ASI) packages.

Note: The wink detection and flash duration attributes can be changed through the Analog Interfaces screen described in Chapter 5, “Switch Interface Administration,” of *CONVERSANT System Version 8.0 Administration*, 585-313-510.

Table 4. Tip/Ring Circuit Card DTMF Tone-Detection Specifications

Attribute	Value
Digits	0–9, asterisk (*), pound sign (#), A–D
Amplitude[1]	+1 to -30 dBm total power (nominal tones)
On/off timing	80 msec minimum on, 23 msec off
Gaps bridged	23 msec
Signal/noise ratio	23 dB (nominal tones at -19 dBm total power)
Twist	+4 to -8 dB (high to low tone)
Frequency deviation	+/-1.5 %

[1] This attribute is adjustable through the Analog Switch Interface (ASI) package.

Table 5. Tip/Ring Circuit Card DTMF Addressing Specifications

Attribute	Default Value
Digits	0 – 9, asterisk (*), pound sign (#), A–D
On/off timing[1]	100 msec on, 60 msec off
Frequency	Precise tones
Twist[1]	0 dB
Amplitude[1]	-3 dBm per frequency
Dial Pulse Addressing Specifications	
Break Time[1]	60 mSec
Make Time[1]	40 mSec
Interdigit Time[1]	600 mSec

[1] These attributes are adjustable through the Analog Switch Interface (ASI) packages.

Table 6. Tip/Ring Circuit Card Default Call Progress Tone Detection Specifications

Tone	Frequency (Hz)[1]	Amplitude (dBm)[1]	S/N Ratio (dB)	Maximum Twist (dB)	Frequency Deviation (%)	Cadence[1]
Dial	350 + 440[1]	+1 to -24	55	+3	+/-0.5	Present for 1 sec
Stutter dial (recall dial tone)	350 + 440 [1]	+1 to -24	55	+3	+/-0.5	3 cycles of 120–150 msec on, 120–150 msec off followed by 1 sec on
Ringback	440 + 480	+1 to -24	55	+3	+/-0.5	1000–2000 msec on, 3000–4000 msec off
Busy	480 + 620	+1 to -24	55	+3	+/-0.5	60 IPM, 250–500 msec on, 500–650 msec off
Reorder (fast busy)	480 + 620	+1 to -24	55	+3	+/-0.5	120 IPM, 180–250 msec on, 250–350 msec off

1. These attributes are adjustable through the Analog Switch Interface (ASI) packages.

Tip/Ring Circuit Card Administration

Placing a card in the INSERV state allows it to be used for the purpose (play, code, and so on.) for which it is allocated in the application. You may need to *manually* place a tip/ring card into service if:

- After first installing the card or changing switch integration parameters, the voice system did not automatically place the card in the INSERV state.
- The card was placed in the MANOOS state.
- A diagnostic procedure failed (that is, placed that card in the MANOOS or BROKEN state).

To change the state of the tip/ring cards to INSERV, use the procedures described in Chapter 3, “Voice System Administration,” of *CONVERSANT System Version 8.0 Administration*, 585-313-510.

Transmission Level Plan

A Transmission Level Plan (TLP) for a piece of telecommunications equipment is a set of specifications dictating the incoming and outgoing speech volume levels that pass through the equipment. The TLP also includes the hardware and software tools for implementing those specifications. The specifications take into account the level plans of the various telephone network interfaces to which the equipment will connect. The goal of the plan is to ensure that all speech heard by a caller be at a level that is appropriate for listening without causing oscillations or distortions in the network.

Most switch designs implement a TLP with a “built-in” gain of -3 dB (often called *insertion loss*) in each tip/ring loop of a station-set-to-station-set connection, for a total gain of -6 dB from end to end (Figure 9). The CONVERSANT system default TLP implements this same strategy; that is, the system default TLP attempts to make the end-to-end gain of voice signals passing through it equal to -6 dB. (There are reasons to implement other strategies, however. See Reasons for Deviating from the Default IVOL and OVOL Settings on page 138 in Appendix A, Transmission Level Adjustment for more information.)

Figure 9. Typical Switch Transmission Level Plan for Station-Set-to-Station-Set Connection

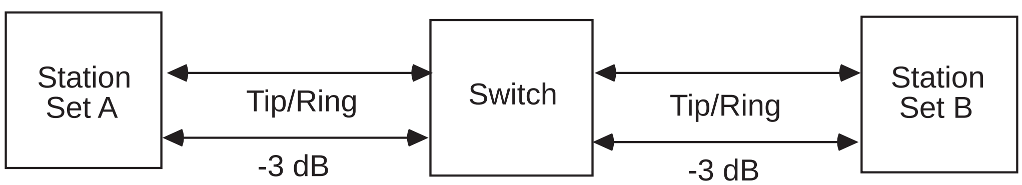


Table 7. Tip/Ring Circuit Card Transmission Level Plan

Attribute	Value
Input gain	0 dB fixed
Output gain	0 dB fixed
Incoming speech volume (IVOL) – card voice coding only	Selectable from -9 to +12 dB
Outgoing speech volume (OVOL) – card voice playback only	Selectable from -9 to +12 dB
TDM output gains	Selectable from -30 to +6 dB

Note: The IVOL, OVOL, and TDM output gains are system-wide parameters for analog interfaces and can be changed on a per-card basis for digital interfaces. These parameters can be modified via the Switch Interface Administration screens as described in Chapter 5, “Switch Interface Administration,” of *CONVERSANT System Version 8.0 Administration,* 585-313-510. Gains can also be overridden on a per-channel basis by an IRAPI application. However, even with IRAPI, the IVOL cannot be overridden for speech recording on a tip/ring channel. See *CONVERSANT System Version 8.0 Application Development with Advanced Methods*, 585-313-216, for the IRP_PLAYGAIN and IRP_RECORD_GAIN parameters under IrPARAMETERS(4IRAPI).

See Appendix A, Transmission Level Adjustment, for more information about adjustment of IVOL and OVOL levels.

Fax Interface

Fax communications involve transmitting graphic and text images between fax machines and other devices via standard telecommunications networks.

For a general discussion of the Script Builder FAX Actions feature package, see *CONVERSANT System Version 8.0 System Description*, 585-313-219.

FAX Provisioning

Applications that use the Script Builder FAX Actions can be assigned to tip/ring, T1, or Line Side (LST1/LSE1) channels. Fax processing can be done by the tip/ring circuit card or the speech and signal processor (SSP) circuit card.

FAX Application Development Issues

The CONVERSANT system can invoke fax services through Script Builder and Voice@Work applications.

FAX Actions

The FAX Actions allow you to include fax communications in any Script Builder or Voice@Work application. FAX Actions offer the following capabilities:

- Transmit a prestored graphic image to a caller
- Transmit a dynamically created text image (file) to a caller
- Create a text file dynamically for transmission to the caller
- Create customized cover pages

For a general discussion of the FAX Actions, see *CONVERSANT System Version 8.0 System Description*, 585-313-219. For detailed information about implementing FAX Actions in CONVERSANT system applications, see Chapter 8, “Using Optional Features with Script Builder,” of *CONVERSANT System Version 8.0 Application Development with Script Builder*, 585-313-217. For detailed information about implementing fax capabilities in Voice@Work applications, see "Standard External Functions" in the Voice@Work online help or Appendix C of *Using Voice@Work*, 585-313-207.

2 Digital Telephony Interfaces

Overview

This chapter describes the T1, E1, line side [Foreign eXchange Station (FXS)], and Primary Rate Interface (PRI) digital telephony interfaces available with the CONVERSANT system. It also describes optional software and the requirements that must be met to implement these interfaces.

This chapter also provides examples of typical digital connections and discusses application development issues you must address when using the various digital telephony interfaces and their parameters.

Introduction to Digital Communications

A digital T1 (E&M) or E1 (CAS) circuit (trunk) allows the system to connect to digital network facilities such as a central office (CO) switch. Digital connections between a switch and the CONVERSANT system can be through PRI, T1 (E&M), E1 (CAS), loop start FXS and ground start FXS. (Generally only E1 or T1 service is offered in a given area.)

Advantages of Digital Service

Analog configurations require one analog connection between the CONVERSANT system and a connected switch for each incoming channel, whereas several digital channels can be transmitted over a single connection. E1 requires only one cable to provide 30 channels of service. T1 requires only one cable to provide 24 channels of service.

Digital connections also significantly reduce the number of circuit cards required to support a CONVERSANT-to-switch interface. Analog configurations require five IVP6 circuit cards to support 30 incoming channels. E1 or line side E1 reduces the required hardware to only one E1 circuit card and part of a speech and signal processor (SSP) circuit card. T1 or line side T1 requires one T1 circuit card and part of an SSP circuit card per 24 channels of digital service. Two T1 circuit cards and one SSP circuit card provide 48 voice channels.

The digital circuit card can be used for either E1 or T1 services.

Advantages of PRI

PRI acts as a powerful interface between intelligent equipment such as PBXs and computers. Furthermore, PRI is widely used for access to features provided over the larger network such as automatic number identification (ANI).

See Primary Rate Interface on page 24 for a detailed discussion of features that accompany the use of PRI.

Note: PRI can be carried on either T1 or E1 lines. It provides 23 bearer (B) channels when carried over T1 lines, and 30 B channels when carried over E1 lines. In either case, calls are controlled from endpoint to endpoint by messages transferred over data (D) channels.

Network Communications

A T1 digital circuit carries information at 1.544 Mbps, and consists of 24 DS-0 channels. Each DS-0 channel operates at 64 Kbps, and is the equivalent of one incoming data line. An E1/T1 interface card has a (mechanical) switch that allows you to choose either the T1 or E1 interface. The E1 interface is very similar to the T1 except that an E1 digital circuit carries information at a rate of 2.048 Mbps and consists of 30 B channels and 2 signaling and framing channels. Each B channel is the equivalent of one incoming data line.

T1 connections also provide dialed number identification service (DNIS) information to further automate incoming calls for customers with multiple 800 or 900 numbers.

T1, E1, and Integrated Services Digital Network (ISDN) PRI support trunk interfaces. PRI can operate at either the T1 or E1 rate. A T1-PRI interface contains either 23 B+D channels or 24 B channels that are associated with the D channel on another 23 B+D card. An E1-PRI interface contains 30 B+D channels. The D channel does not provide normal telephony service, but is used to control the calls on the B channels. It provides information such as DNIS and ANI. Each B channel provides a 64-Kbps voice path.

Interconnection with PBX

Line-side connections between a switch and a CONVERSANT system can be made by means of either a T1 interface or an E1 interface. See FXS or “Line Side” Digital Interface on page 21 for more information.

The default gain in each B channel is 0 dB. Transmission levels are discussed under Transmission Level Plan on page 135 in Appendix A, Transmission Level Adjustment. Possible reasons for adjusting gain are given under Reasons for Deviating from the Default IVOL and OVOL Settings on page 138 also in Appendix A.

These line side T1 and line side E1 channels also support the Adjunct/Switch Application Interface (ASAI) feature when used with DEFINITY switches. (ASAI can be used for more advanced call control and to collect information such as ANI and DNIS.) See Advantages of Using the ASAI Feature on page 31 in Chapter 3, Adjunct/Switch Application Interface.

T1 (E&M), E1 (CAS), and PRI connections to a DEFINITY switch are supported as well as line side FXS T1 or line side FXS E1, but line side FXS T1 and line side FXS E1 are generally preferable. Line side FXS T1 and line side FXS E1 support switch-hook-flash transfers, but E1 (CAS), and PRI do not.

Note: T1 E&M supports switch-hook flash transfers if the switch supports it.

The system supports call bridging through a digital connection. Call bridges can also be used to simulate a transfer, but this consumes channel resources.

Digital Telephony Interface Specifications

Table 8 details the general digital telephony interface specifications for all T1/E1 protocols.

Table 8. Digital Telephony Interface General Specifications

Attribute	Specification for E1/T1 Circuit Card
Physical connector	BNC co-ax or 8-pin modular
FCC registration	AS5USA-24091-XD-E
Safety approval	• UL 1459 type approval for US markets • CSA 22.2 type approval for Canadian markets • EN 60950 type approval for European markets • AS3260 and TS-001 for Australian markets
T1 signal regeneration	CSU required over 200 meters (655 feet)
T1 loopback capability	CSU required for remote capability
Transmission level point (TLP) at DS-1 interface	0 ELP, 0 DLP
TLP at time-division multiplexed (TDM) interface	0 ELP, 0 DLP
Call progress tone frequency	Precise tone frequencies can be tuned to accommodate local standards
Call progress tone levels	-10 dBm total (nominal) This value is tunable through digital switch interface packages.
Call progress tone timing	• Ringing –on/off: 2 sec on, 4 sec off • Busy – on/off: 0.5 sec on, 0.5 sec off Values are tunable through digital switch interface packages
Call progress tone detection	Supported with Line Side FXS (loop start) protocol (either at T1 or E1 transmission rate)

1 of 2

Table 8. Digital Telephony Interface General Specifications

Attribute	Specification for E1/T1 Circuit Card
DS-1 timing source	Slave to DS-1 source (loop timed)
DS-1 timing (free running)	Stratum 4
Suggested channel service unit (CSU) types for use at T1 rate	• Paradyne (PEC 21581-ESF • Verilink 551VST List 2, or equivalent
Supported configurations	Tie trunk (robbed-bit E&M), E1 (CAS), ISDN-PRI (E1/T1), LSE1, LST1
Dual tone multifrequency (DTMF) output timing	70 msec on, 70 msec off This value is tunable through digital switch interface packages.
DTMF output levels	-8 dBm per frequency (nominal) This value is tunable through digital switch interface packages.
DTMF receivers	LATA Switching Systems Generic Requirements (LSSGR) compatible. Note: If DTMF muting is on for a call, the DTMF receiver’s minimum on time for detection is increased and may not meet LSSGR requirements. DTMF muting does not impact LSSGR compatibility of DTMF receivers during call setup, that is, S digits. This value is tunable through digital switch interface packages.
Number of receivers: T1	24 (1 per DS-0 channel)
Number of receivers: E1	30 (1 per B-channel)

2 of 2

Digital Connectivity

The system supports up to six T1 circuit cards. An SSP circuit card is required if you are using T1 circuit cards in coding and playback situations.

Note: Each SSP circuit card supports up to 120 channels of simultaneous speech playback using adaptive differential pulse code modulation (ADPCM) 32-Kbps coded speech.

See the “Installing or Replacing Circuit Cards” chapter in the maintenance book for your platform for information on installing digital and SSP circuit cards.

Figure 10 and Figure 11 show examples of typical digital connections to trunks and switches. Table 10 on page 19 details the digital telephony specification for the T1.5 Robbed-bit E&M protocol. Use Table 10 on page 19 in conjunction with Table 8 on page 13.

Figure 10. Example of E1/T1 Coaxial Connections to a DEFINITY G3 Switch

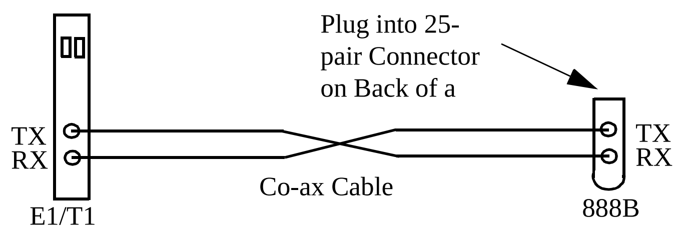
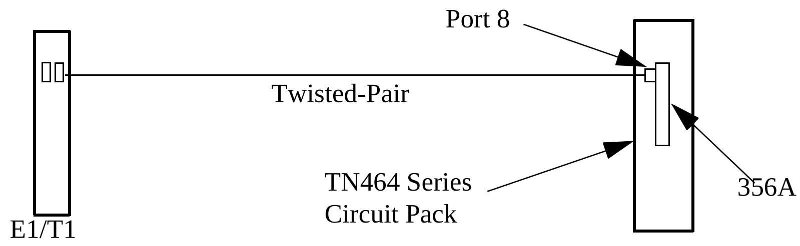


Figure 11. Example of E1/T1 Twisted-Pair Connections to a DEFINITY G3 Switch



Channel Service Unit Connectivity (T1 Only)

The T1 interface circuit card is connected to a CSU or directly to the DS-1 terminal block to establish T1 connections to a CO.

A CSU performs certain line-conditioning and equalization functions and responds to loopback commands from the CO. A CSU regenerates digital signals, monitors them for problems, and provides a way to test the digital circuit. A CSU is not always needed. However, a CSU is *required* if any of the following situations applies to the system setup:

- The CONVERSANT system is more than 200 meters (655 feet) from the signal source. The signal source may be a DSX or the last T1 repeater. Here, the CSU regenerates the received signal and properly attenuates the transmitted signal to prevent crosstalk.
- The CONVERSANT system is terminating the T1 trunk from outside the building. Here, the CSU provides the primary lightning and surge protection as required by FCC Rules Part 68.

- The T1 loop is not dry (that is, the loop is powered by either 110 VAC, +24 VDC or -48 VDC sources).
- You want to use the remote loopback and/or extended super frame (ESF) maintenance features. Here, the CSU recognizes the in-band bit patterns that signal it to loopback the incoming signal or to perform other maintenance functions.

On some types of CSUs, the connector on the T1 cable can plug into the E1/T1 circuit card and the cable terminates at a 15-pin D subminiature connector to the CSU (Figure 12). On other types, you must cut off the CSU connector and slide latch and strip and connect the wires (Figure 13).

Figure 12. Example of T1 Interface Connection to a CSU (From an E1/T1 Circuit Card

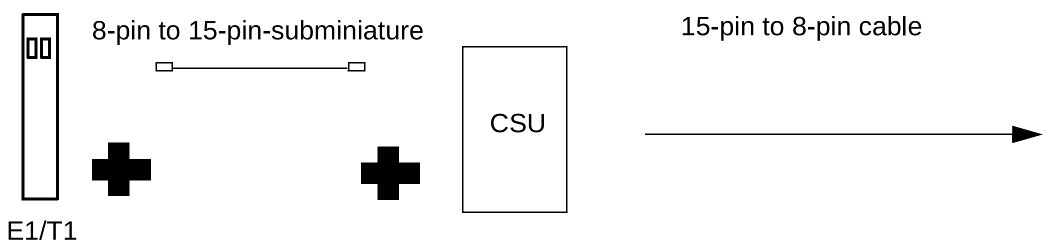
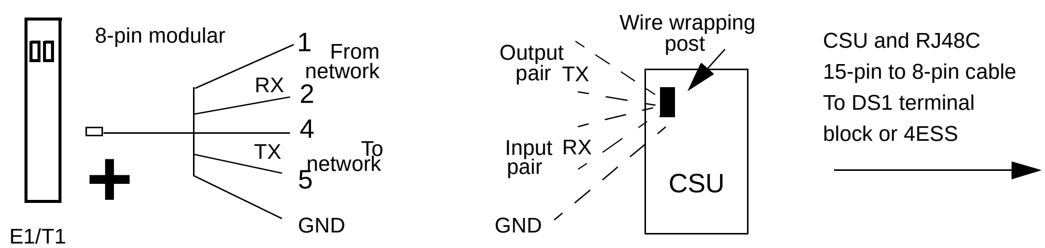


Figure 13. Example of E1/T1 Connection to a CSU with Wire Wrapping Posts



E1-CAS Interface

The E1/T1 circuit cards can operate with Channel Associated Signaling (CAS). This interface (at the E1 rate: 2.048 Mbits/sec) uses signaling bits associated with each channel to determine the state of the channel. Each link supports 30 voice channels.

Several country-specific signaling protocols have been developed using E1-CAS. Contact your local Avaya technical representative for more information about locally supported protocols.

Note: The system supports the R2-MFC for Mexico, Argentina, Brazil, and International Telecommunication Union (ITU).

Table 9. Digital Telephony Interface Specifications for E1-CAS Configurations

Attribute	Specification
DS1 rate	2.048 Mbits/sec (ITU G.703)
DS1 framing/line coding	HDB3 (ITU G.704, G.705)
Cyclic redundancy check (CRC)	(ITU G.706) May be set to YES or NO; must match the CRC setting of the network entity connected to the E1/T1 card
PCM companding rule	A-Law or Mu-Law (ITU G.711)
Line signaling	ITU System R2, Q.421 compliant; variations by specific protocol are supported
Address signaling options (register signaling) incoming and outgoing	DTMF (touchtone) ITU system MFC, Q.440, Q.441; variations for specific protocols are supported by table entries; dial pulse (slower than DTMF or MFC)
Outgoing destination number	15 digits maximum
Outgoing ANI number	15 digits maximum (if supported by protocol)
Incoming address: (DNIS)	15 digits maximum
Incoming ANI number	15 digits maximum (if supported by protocol)
Audible alerting tones on incoming calls	Ring, busy, reorder; variations by country supported
Call progress tone recognition on outbound calls	Not supported
Call transfer capability	Not supported

E1 Switch Integration and Administration

Switch Integration for E1-CAS is done using the Digital Interfaces screen. This screen is described in Chapter 5, “Switch Interface Administration,” of *CONVERSANT System Version 8.0 Administration,* 585-313-510. You must select one of the E1-CAS protocols that correspond to optional packages loaded on the CONVERSANT system.

Placing a card in the INSERV state allows it to be used for the purpose for which it is allocated in the application. After performing switch integration on the E1 circuit card for the CAS protocol, you may need to *manually* place an E1 circuit card into service if:

- After first installing the card or changing switch integration parameters, the voice system did not automatically place the card in the INSERV state.
- The card was placed in the MANOOS state.
- A diagnostic procedure failed (that is, placed that card in the MANOOS or BROKEN state).

To change the state of the E1 circuit cards to INSERV, use the procedure described in Chapter 3, “Voice System Administration,” of *CONVERSANT System Version 8.0 Administration*, 585-313-510.

E1 Connections

Because telephone network connections vary from country to country, no specific recommendation can be made concerning connection to the network entity. Consult your local Avaya technical representative to determine the proper physical connectivity.

T1 E&M Interface

The T1 circuit cards accept an ISDN PRI or DS-1 two-way digital trunk and convert it to two-way analog audio channels. Because of the bandwidth and transmission differences of each trunk, ISDN PRI and DS-1 offer different numbers of converted channels. A standard 1.544-Mbps DS-1 format trunk converts to 24 DS-0 channels. These 64-Kbps channels can provide 24 two-way audio channels.

Table 10. Digital Telephony Interface Specifications for T1 E&M Type Configurations

Attribute	Specification
DS-1 framing	D4 type only
DS-1 line coding	Zero code suppression (ZCS)
Protocol	Robbed-bit (4-wire) E&M
Alerting in/out	Wink/wink
Wink generation	230 msec default (selectable: 20–2500 msec)
Wink detection range	10–350 msec
Addressing (outgoing)	• DTMF (touchtone) • MF (multifrequency) • Dial pulse (slower than DTMF or MF)
Number of digits	15-digit maximum
Addressing (incoming)	• DTMF (touchtone) • MF (multifrequency) • Dial pulse (slower than DTMF or MF)
Number of digits (DNIS)	Will wait for up to 16 digits (selectable); can also be provisioned not to wait for digits
Initial digit timer	Will wait up to 4 seconds for first digit; can also be provisioned not to wait for digits
Interdigital timer	Will wait up to 2 seconds between digits
Audible ring starts	As soon as the selected number of digits is received or when one of the above timers expire, whichever occurs first
DNIS capacity	0–16 digits
ANI capacity	Not supported
Transfer capability	Supported

T1 Switch Integration and Administration

Switch integration for T1 is done using the Digital Interfaces screen. This screen is described in Chapter 5, “Switch Interface Administration,” of *CONVERSANT System Version 8.0 Administration*, 585-313-510. You must select T1 A/B robbed-bit E&M Protocol from the Digital Interfaces screen. See Switch Integration and Administration on page 22 and Primary Rate Interface on page 24 for information on performing switch integration for those types of protocols.

Placing a card in the INSERV state allows it to be used for the purpose for which it is allocated in the application. After performing switch integration on the T1 circuit card for the E&M protocol, you may need to *manually* place a T1 circuit card into service if:

- After first installing the card or changing switch integration parameters, the voice system did not automatically place the card in the INSERV state.
- The card was placed in the MANOOS state.
- A diagnostic procedure failed (that is, placed that card in the MANOOS or BROKEN state).

To change the state of the T1 circuit cards to INSERV, use the procedure described in Chapter 3, “Voice System Administration,” of *CONVERSANT System Version 8.0 Administration*, 585-313-510.

T1 Connections

See Channel Service Unit Connectivity (T1 Only) on page 15 for information on T1 connections through CSUs.

Digital Application Development Issues

The E1/T1 circuit card recognizes call progress tones and therefore supports flash transfers over LSE1 or LST1. The E1/T1 circuit card does not, however, support flash transfer over E1 (CAS) or PRI. See FXS or “Line Side” Digital Interface on page 21 for more information.

Simulated transfers using digital cards can be performed over call bridges. In the analog Tip/Ring or line-side digital environment, the switch-hook-flash transfer releases the call from the CONVERSANT system once the transfer is made. A call bridge, however, ties up an incoming port and an outgoing port until the call has concluded. Thus, with two ports being tied up simultaneously, more digital ports may be necessary.

Script Language

The **tic** instruction is used for basic control of incoming and outgoing calls on T1 and E1 lines. For more information about using the transaction state machine (TSM) script language on T1 lines, see the **tic** instruction in Chapter 3, “TAS Script Instructions,” and Appendix B, “Summary of Script Instructions,” of *CONVERSANT System Version 8.0 Application Development with Advanced Methods*, 585-313-216.

Response Application Programming Interface

The **irCall()**, **irAnswer()**, **irDial()**, and **irDisconnect()** functions provide the basic call control capabilities for T1 interfaces with the Response Application Programming Interface (IRAPI). The **irStartSpeechED()** function is supported, for LST1 or LSE1 interfaces over an E1/T1 circuit card. See Chapter 5, “IRAPI,” of *CONVERSANT System Version 8.0 Application Development with Advanced Methods*, 585-313-216, for more information about these functions when developing IRAPI applications.

FXS or “Line Side” Digital Interface

The line side digital protocol set has been expanded. The protocol formerly called “Line Side T1 for Definity” is a standard loop start Foreign eXchange Station (FXS) protocol specified in the ANSI generic PBX requirements (ANSI TIA/EIA-464-B-1996, section 6.2.3). In addition to the loop start version of the protocol, a ground start version of the protocol (TIA/EIA-464-B-1996, section 6.2.4) is also provided. These two protocols are widely supported by PBXs, central office switches, and channel banks under names like “line side digital,” “FXS,” or “off premise station (OPS).” These protocols will work over digital cards configured as T1 or E1. Switches known to support either or both of the loop start and ground start FXS protocols include Avaya Definity, Nortel Meridian, and Rockwell Spectrum.

Loop start and ground start FXS allows the use of a 24-channel, 1.544-Mbps digital interface between a switch and the CONVERSANT system. A T1 configuration uses T1 circuit card technology with special protocol-level software and CONVERSANT system user-interface modifications. This technology improves system connectivity and reduces the number of circuit cards and cables required (relative to tip/ring technology) to support 24 channels of service. An E1 configuration allows the use of a 30-channel, 2.048-Mbps digital interface between a switch and the CONVERSANT system V8 platform. E1 provides a similar improvement in connectivity relative to tip/ring cards.

Loop Start E1/T1 Provisioning

When either loop start FXS with E1 or T1 is used to provide an ASAI link between the CONVERSANT system and a switch, a separate path must be provided for communications between the two systems. The path must be provided by a MAPD connected to a local area network.

The following limitations apply when you use a line side digital interface:

- When a switch is excessively loaded and a timed delay is used prior to dialing, a call can be lost if the switch is not properly engineered and administered.
- Dial pulse is not supported on either T1 or E1 channels; however, dialing of DTMF tones is supported.
- The DEFINITY G2 does not provide forward disconnect.

Table 11 on page 22 details the digital telephony interface specifications for line side E1 and T1 configurations. Use Table 11 on page 22 in conjunction with Table 8 on page 13.

Table 11. Digital Telephony Interface Specifications for Line Side Configurations

Attribute	E1/T1 Circuit Card
DS-1 framing	D4 for T1 and CEPT for E1
DS-1 line coding	ZCS for T1 and HDB3 for E1
Wink-disconnect interval	300-msec default (selectable within a range of 10–2500 msec)
Dial-tone delay	1000-msec default (selectable within a range of 20–5100 msec)
Switch-hook-flash duration	700-msec default (selectable within a range of 10–2500 msec)
DNIS capacity	Not supported unless used with the converse vector step (CVS) or ASAI
ANI capacity	Not supported unless used with CVS or ASAI
Transfer capability	Flash transfers supported

Using the Converse Vector Step

The in-band DNIS capability is available when using the CVS feature of DEFINITY on loop start FXS E1 and T1 channels. See Chapter 5, Converse Vector Step Routing, for more information.

Switch Integration and Administration

Switch integration for loop start and ground start FXS is done using the Digital Interfaces screen. This screen is described in Chapter 5, “Switch Interface Administration,” of *CONVERSANT System Version 8.0 Administration*, 585-313-510. You must select appropriate protocol for the switch from the Digital Interfaces screen.

Placing a card in the INSERV state allows it to be used for the purpose (play, code, and so on) for which it is allocated in the application. After performing switch integration on the E1/T1 circuit card for the loop start or ground start FXS protocol, you may need to *manually* place the card into service if:

- After first installing the card or changing switch integration parameters, the voice system did not automatically place the card in the INSERV state.
- The card was placed in the MANOOS state.
- A diagnostic procedure failed (that is, placed that card in the MANOOS or BROKEN state).

To change the state of the E1/T1 circuit cards to INSERV, use the procedure described in Chapter 3, “Voice System Administration,” of *CONVERSANT System Version 8.0 Administration*, 585-313-510.

Loop Start and Ground Start FXS Connections

See Digital Connectivity on page 15 for examples of digital connection using loop start or ground start FXS on a digital switch.

Application Development Issues

The following are application development issues for Script Builder, script language, and the IRAPI.

Script Builder

Line side FXS T1/E1 supports blind call origination (outcalling) and blind call transfers for switches only as normally performed on tip/ring lines. Blind transfers mean that the CONVERSANT system does not detect call-progress tones or provide any form of answer supervision. Line side FXS T1/E1 can provide CPT detection only when used with Full CCA or when connected by means of an E1/T1 circuit card. Line side FXS T1/E1 does not support call progress tone detection for calls that go outside the DEFINITY switch.

Script Language

The following script instructions support FXS T1 and FXS E1 operations:

- **tic('C')** (only available with the use of Full CCA or when using E1/T1)
- **tic('o')**
- **tic ('O')** (only available with the use of Full CCA or when using E1/T1)
- **tic('f)**
- **tic('F')**
- **tic('d')**
- **tic('D')** (only available with the use of Full CCA or when using E1/T1)
- **tic('h')**

See the section on the **tic** instruction in Chapter 3, "TAS Script Instructions," and Appendix B, "Summary of TAS Script Instructions," of *CONVERSANT System Version 8.0 Application Development with Advanced Methods*, 585-313-216, for more information.

IRAPI

See Digital Application Development Issues on page 20 for details on the supported IRAPI functions for T1 interfaces. See Chapter 5, "IRAPI," of the *CONVERSANT System Version 8.0 Application Development with Advanced Methods*, 585-313-216, for more information about these functions when developing IRAPI applications using FXS T1 or FXS E1.

Primary Rate Interface

PRI is desirable for customers who need faster call-setup times, special signaling, or access to the information elements that are available with PRI. Such information elements as ANI, DNIS, redirecting number, and service type are available for incoming PRI calls. Outbound calls can provide information elements like outbound ANI, service type, and bearer capability.

CONVERSANT system V8 supports the following PRI protocols:

- National (with and without BCAS)
- AT&T
- ETSI
- Nortel

The CONVERSANT system supports the PRI protocols between itself and the digital telephone network or entity through the use of a special digital protocol, with the same physical connectivity as standard T1 digital communication. The PRI digital protocol uses either D4 or ESF framing. Standard T1 circuit card connectivity, as described in the previous sections, is used to implement the physical connection between the system and the remote network entity when using PRI.

PRI connectivity offers the ability to administer key protocol parameters through software interfaces. This parameter administration must be performed before the physical connectivity is established. Two key parameters are dependent on the framing protocol used. If D4 framing is used, line coding must be “ZCS” and D-channel inversion must be *inverted*. If ESF framing is used, line coding must be “B8ZS” and D-channel inversion must be *noninverted*. The PRI service provider determines the method of framing used. ESF/B8ZS is preferred.

When operating at the E1 rate, use CEPT framing and HDB3 line coding. CEPT/HDB3 are the only options allowed at the E1 rate.

The CONVERSANT system does not support switch-hook-flash transfers using PRI configurations. Simulated T1 transfers can be performed only over call bridges. In both the analog tip/ring and digital line side environments, the switch-hook-flash transfer releases the call from the CONVERSANT system once the transfer is made. A call bridge, however, ties up an incoming port and an outgoing port until the call is concluded. Thus, with two ports being tied up simultaneously, more ports may be necessary.

Table 12 on page 25 details the digital telephony interface specifications for PRI type configurations. Use Table 12 on page 25 in conjunction with Table 8 on page 13.

Table 12. Digital Telephony Interface Specifications for PRI Type Configurations on an E1/T1 Circuit Card

Attribute	Specification
DS-1 framing	D4 or ESF (selectable) for T1 rate, CEPT for E1
DS-1 line coding	• ZCS (with T1 D4 framing only) • B8ZS (with T1 ESF framing only) • HDB3 (with E1 CEPT framing only)
B-channel capacities	Up to 143 B+D when six T1 cards are used; up to 149 channels when 5 150 B+D E1 cards are used
	See the *CONVERSANT System Version 8.0 System Description*, 585-313-219, for a list of platform limitations. **Note:** These configurations are switch dependent. Not all switches support all configurations.
D-channel capacities	Multiple D-channels are supported up to the maximum number of T1/E1 cards (six channels for 6 T1 cards, five channels for 5 E1 cards)
Interface ID	• 1 (for a card with a D-channel, not selectable) • 2–6 (for a card without a D-channel)
DNIS capacity	0–15 digits
ANI capacity	0–15 digits
D-channel backup	Not supported
Transfer capability	Not supported

Using PRI in a DEFINITY Call Center

ISDN-PRI provides a Universal Call ID (UCID) capability for every call in a DEFINITY call center customer environment. UCID provides a unique identifier (8-byte binary or 20-character ASCII) to allow for uniform data-tracking for all call-related data in a call center, regardless of the system. Also, available is the User-to-User Information element (UUI), which allows the customer to specify additional information to be passed in external function arguments. For more information about these features, see *CONVERSANT System Version 8.0 Application Development with Advanced Methods*, 585-313-216.

PRI Provisioning

Supported B-channel capacities in PRI configurations are switch dependent (see Table 12 on page 25). Not all switches support all configurations. For example, the 5ESS switch only supports the 23 B+D configuration, but the 4ESS switch can support up to 119 B+D. See the *CONVERSANT System Version 8.0 System Description*, 585-313-219, for information on supported PRI configurations.

Special parameter provisioning of PRI is required on the switch, but is not part of the normal order process for AT&T PRI network services. Thus, give special attention to the determination and provisioning of these parameters when ordering and implementing this feature. In addition, the CONVERSANT system uses some parameters that must be correct and matching in both machines. Table 13 show how to set these parameters on the switch. See *CONVERSANT System Version 8.0 New System Installation*, 585-313-149, for additional provisioning information.

You should provision incoming calls to the CONVERSANT system so that the channel number is exclusive and not preferred. Also, if the switch is configured to deliver ANI on a subscription basis, it is not possible for the system to request a different type of ANI on a call-by-call basis.

Table 13. PRI Parameters and Values

Parameter	AT&T ISDN	ETSI PRI	National BCAS ISDN (with Service messages)	National ISDN (without Service messages)	Nortel PRI
DCHAN_DELAY	0	0	0	0	0
NPI_TOA	0x400	0x400	0x400	0x400	0x400
PROTOCOL	0	0	0	0	0
FLAGS	0x33609	0xc80	0x409a1	0x40980	0x80941
Timer T203[1] (in seconds)	30	10	30	30	10
Timer T302 (in seconds)	15	15	15	15	15
Timer T303[1] (in seconds)	4	4	4	4	4
Timer T304[1] (in seconds)	30	30	30	30	30
Timer T305[1] (in seconds)	4	30	30	30	30
Timer T308[1] (in seconds)	4	4	4	4	4
Timer T309[1] (in seconds)	30	90	90	90	90

1 of 2

Table 13. PRI Parameters and Values

Parameter	AT&T ISDN	ETSI PRI	National BCAS ISDN (with Service messages)	National ISDN (without Service messages)	Nortel PRI
Timer T310[1] (in seconds)	10	40	30	30	10
Timer T313[1] (in seconds)	4	4	4	4	4
Timer T316[1] (in seconds)	120	120	30	30	120
Timer T3M1[1] (in seconds)	120	120	120	120	120
Timer T372 (in seconds)	7	7	7	7	7
					2 of 2

[1] All timers are adjustable as described in the **/vs/man/cat4/pri.rc.4** file

PRI Switch Integration and Administration

Switch integration for the PRI feature is done using the Digital Interfaces screen. This screen is described in Chapter 5, “Switch Interface Administration,” of *CONVERSANT System Version 8.0 Administration*, 585-313-510. You must select ISDN-PRI Layer 1 Protocol from the Digital Interfaces screen.

To assign PRI functionality to an SSP circuit card, see Chapter 3, “Voice System Administration,” of *CONVERSANT System Version 8.0 Administration*, 585-313-510. To assign PRI functionality to a T1 or E1 circuit card, see Chapter 5, “Switch Interface Administration,” of *CONVERSANT System Version 8.0 Administration*, 585-313-510.

PRI Connections

As mentioned earlier, PRI can be connected through either a T1 or E1 circuit card. See Digital Connectivity on page 15 for examples.

Understanding B-Channel and D-Channel

Only one T1 circuit card can be configured with the D-channel. The D-channel is always the 24th channel of this circuit card. (See the information on assigning the PRI Layer 1 Protocol to a T1 circuit card in Chapter 5, “Switch Interface Administration,” of *CONVERSANT System Version 8.0 Administration*, 585-313-510, for more details). The D-channel cannot be used to run applications. It carries messages between the switch and the system. These messages are used to control the state of calls on all the other PRI channels.

All the other PRI channels are referred to as B (bearer) channels. The B-channels provide two-way audio channels to run applications. Therefore, on a PRI that is configured to have only one T1 circuit card, the first 23 channels (B-channels) on that card can be used to run applications. The 24th channel (D-channel) is reserved for call control. If your PRI is configured with more than one T1 card, the additional T1 cards (the ones configured without a D-channel) will have 24 B-channels on which to run applications. The system can run applications on a total of 143 B-channels (that is, six T1 cards).

Note: To provide acceptable performance, only 96 B-channels can be used for incoming calls. The rest of the channels must be used for outgoing bridged calls.

For E1 PRI, channel 0 is reserved for framing and channel 16 is reserved for the D channel, the remaining 30 channels are B-channels. Typically, each E1 PRI interface has its own D-channel (unlike T1 PRI where a single D-channel frequently controls more than one T1 interface).

Determining the D-Channel

If you do not know which channels have the D-channels, perform the following procedure. See Chapter 3, “Voice System Administration,” of *CONVERSANT System Version 8.0 Administration*, 585-313-510, for more information.

To determine which channels have the D-channel:

1 Start at the CONVERSANT Administration menu, and select:

```
> Voice System Administration
  >Configuration Management
    >Equipment
```

The system displays the Voice Equipment screen showing a list of all channels in the system.

2 Use the ▲ and ▼ keys to scroll through the list of channels.

The D-channels are the only channels that are labeled “PRID” in the **TYPE** column. B-channels are labeled *PRIB*.

Once you know which channels have the D-channel, you are ready to bring the PRI into service to allow it to begin taking calls. Change the state of all PRI channels to **INSERV** using the procedure described in Chapter 3, “Voice System Administration,” of *CONVERSANT System Version 8.0 Administration*, 585-313-510.

Display the Options field to see with which D-channel group the card is associated (PRI1 PRI2, and so on) and whether it has a D-channel (DCHAN).

PRI Application Development Issues

The following are PRI application development issues for Script Builder, script language, and the IRAPI.

Script Builder

The PRI feature provides the following Script Builder external actions and an external function for use in PRI applications:

- The ISDN_billing external action provides the billing number to incoming call applications.
- The Attr_ANI external function allows an application to request the billing number for incoming calls on a call-by-call basis.

Note: The Attr_ANI external function is not necessary for facilities that subscribe to ANI.

- The ISDN_service external action allows an application to choose Service Type for outgoing PRI calls.

In addition, PRI supports the following call-control Script Builder actions:

- Answer
- Disconnect
- Make Call
- Call Bridge

The Call Transfer action is not supported for PRI because the PRI protocol does not support the transfer function.

For more information about integrating the PRI feature in your CONVERSANT application, see Chapter 8, “Using Optional Features,” of *CONVERSANT System Version 8.0 Application Development with Script Builder*, 585-313-217.

Script Language

Several capabilities are available to implement the PRI feature in TSM script language applications.

- The **tic** instruction is used for basic control of incoming and outgoing calls on the PRI. The **tic(‘C’)** and **tic(‘O’)** instructions provide additional return code information over the T1 and analog interface implementations.

  The following additional script registers apply to PRI:

  ~ IE.ANI – Calling party number
  ~ IE.DNIS – Called party number
  ~ IE.REDIRECTING – Originally dialed number
  ~ IE.SERVICE – Incoming service type
- The **setattr** instruction can be used to request the Calling Party Number (CPN) from the network before starting the script.
- The **setstring** instruction can be used to send a CPN on an outbound call.
- The **setparam** instruction can be used to specify an outbound service type or bearer capability on an outbound call.

For more information about integrating the PRI feature using TSM script language, see Chapter 3, “TAS Script Instructions,” of *CONVERSANT System Version 8.0 Application Development with Advanced Methods*, 585-313-216.

Response API

The **irCall()**, **irAnswer()**, **irDial()**, and **irDisconnect()** functions provide the basic call control capabilities for T1 interfaces. The **irFlash()** and **irStartSpeechED()** function is not supported for PRI interfaces. The **irSetIE()** and **irGetIE()** can be used to set and get information elements available only with PRI. See Chapter 5, “IRAPI,” of *CONVERSANT System Version 8.0 Application Development with Advanced Methods*, 585-313-216, for more information about these functions when developing IRAPI applications.

Advanced PRI Capabilities

For more information about PRI and for sample applications of advanced PRI programming, see the *CONVERSANT System Version 8.0 Advanced PRI Developer’s Guide*, comcode 108199167. This document is for the use of independent software vendors (ISVs) and value added resellers (VARs) only, who use it to create packages that provide additional PRI capabilities for their customers, in other words, the end users.

This document is also for the use of the CONVERSANT application developer who wants to develop applications that go beyond the standard Commercial PRI capabilities that are documented above and elsewhere. It describes how to extend the capabilities of the standard Commercial PRI services and how to develop General Purpose PRI application processes that offer even more complete control of the ISDN PRI signaling.

3 Adjunct/Switch Application Interface

Overview

This chapter describes the use of the CONVERSANT system Adjunct/Switch Application Interface (ASAI) feature and the requirements that must be met to implement this interface. Also provided are ASAI application and call flow examples, a discussion of the use of ASAI versus the DEFINITY converse vector step (CVS), and a list of application development issues that must be addressed when using ASAI.

ASAI Overview

Briefly, ASAI provides an ISDN-based interface between switches and adjunct processors. The CONVERSANT system’s ASAI feature supports this application interface for communications with the Avaya DEFINITY Communications System, Generic 3 (hereafter referred to as the DEFINITY G3 switch). This digital signaling interface allows the CONVERSANT system to monitor and route calls on the DEFINITY G3 switch. When used in conjunction with tip/ring or digital Line Side E1 or T1 interfaces (LSE1/LST1), the ASAI interface allows the system to monitor and control the incoming calls it receives.

Advantages of Using the ASAI Feature

When using ASAI, caller-dependent and region-dependent treatment for incoming calls is possible in routing and voice response applications. In addition, the direct agent calling feature available with these applications allows calls to be delivered to specific agents while maintaining accurate split measurements. These capabilities help to ensure that calls are quickly and reliably directed to the call center resource best suited to handle them. This minimizes the number of transfers a caller experiences and allows callers to be serviced in a rapid, consistent, and personalized fashion.

In data screen delivery applications, information associated with a given call is available to each agent receiving the call. For example, a caller may be directed initially to a CONVERSANT tip/ring or LST1/LSE1 channel where the caller is prompted through an automated voice response application. At some point the caller may request to be transferred to a live agent to discuss a topic in more detail. With the ASAI feature, the identity of the caller and additional information collected from the caller by the voice response application is not lost. Pertinent information from the voice response application can be saved and presented in a data screen to the live agent who receives the transferred call, thereby eliminating the need for the customer to repeat information already collected. This ability reduces call holding time and eliminates the need for the caller to repeat the same information to each agent. This benefit holds true even when calls are transferred several times or are transferred between live agents.

The ASAI feature eliminates the need for multiple boxes with multiple interfaces to the host computer, thereby simplifying host application development. Access to ASAI capabilities using Script Builder minimizes the effort required to implement the CONVERSANT piece of the overall CONVERSANT/host application. Such ASAI information as automatic number identification (ANI) and dialed number identification service (DNIS) related to a particular call can be retrieved for use in the script that is handling the call. See ASAI Application Development Issues on page 39 for more information.

The use of data screen delivery applications reduces the time needed to service calls. This is because the host screen application is ready to provide or accept information at the same time the agent begins to speak with the caller. The reduction in per-call service time translates directly into reduced 800-network costs and reduced agent costs. Also, certain calls can be eliminated entirely via the use of routing applications, for example, call screening for the identification of fraudulent calls. In this case, no network costs are incurred for the call and no agent time is wasted on the call.

Call Center Features

The ASAI package provides the following capabilities for use in DEFINITY call center environments. See Chapter 8, “Using Optional Features,” of *CONVERSANT System Version 8.0 Application Development with Script Builder*, 585-313-217, for information on implementing these features into your applications.

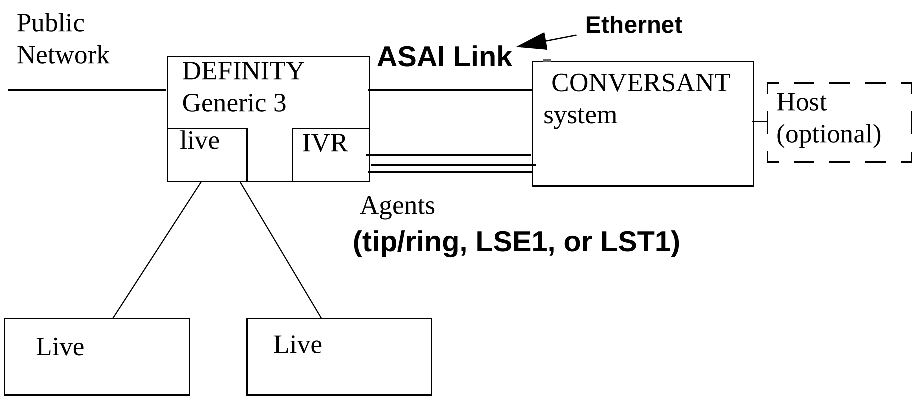
- Universal Call ID (UCID) — UCID provides a unique identifier (8-byte binary or 20-character ASCII) for every call in a DEFINITY call center customer environment. UCID allows for uniform data-tracking for all call-related data in a call center, regardless of the system. DEFINITY uses the ASAI interface to pass the UCID to the adjunct.
- ANI Information Indicator (ANI-II) — ANI-II provides a number that indicates the class of service of the customer who is calling, such as residential, coin, or wireless.
- User-to-User Information element (UUI) — UUI allows the customer to specify additional information to be passed in external function arguments, which can contain up to 32 bytes of information.

ASAI Connectivity

An ASAI link between the DEFINITY G3 switch and the CONVERSANT system delivers control and supervisory messages about each tip/ring, line side FXS E1 or line side FXS T1 channel. The link must be implemented as an ethernet connection. One ASAI link per CONVERSANT system is supported.

Generally, such a configuration looks like that shown in Figure 14.

Figure 14. Typical CONVERSANT System and DEFINITY G3 Configuration



Note: The public network must provide a PRI connection to the DEFINITY G3 switch for an application to receive calling number information.

Establishing an Ethernet ASAI Link

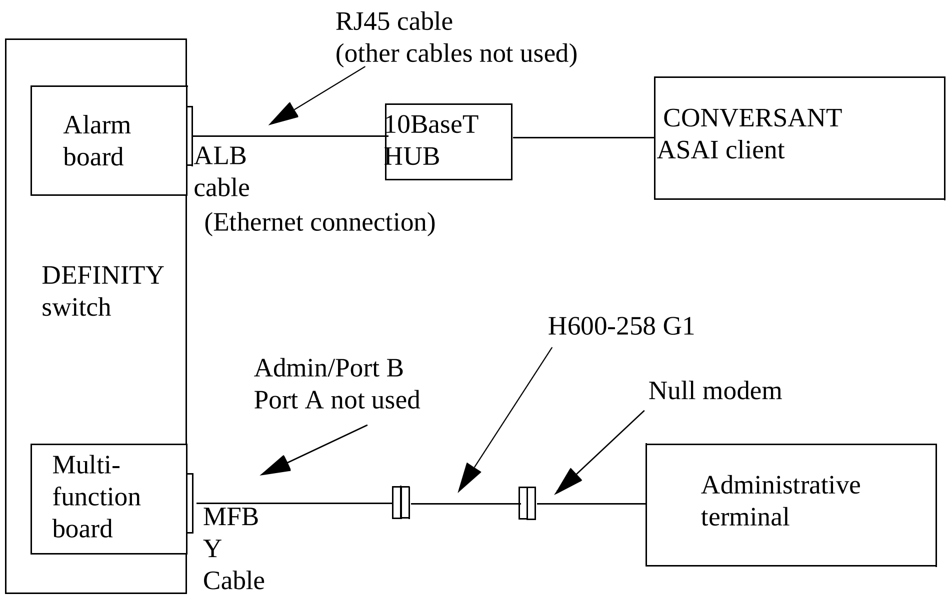
To establish an ASAI link between a DEFINITY G3 switch and a CONVERSANT system, one of the integrated LAN connections on the CONVERSANT system is connected to a MAPD circuit card in the DEFINITY G3 switch.

For more information about installation of the MAPD, see *Call Visor PC LAN over MAPD Installation, Administration, and Maintenance,* 555-230-113.

For information about connectivity to a DEFINITY switch, see *DEFINITY Communications System Generic 3 CallVisor ASAI Planning Guide,* Issue 4, 555-230-222.

Figure 15 on page 34 shows a typical LAN configuration.

Figure 15. Typical LAN Wiring for an ASAI Link



Connecting the CONVERSANT System Agents

The following information describes how to make tip/ring and LST1 connections from the CONVERSANT system to the switch.

Analog Tip/Ring Connections

Note: Analog connections are not supported on the UCS 1000.

ASAI can be provisioned using analog tip/ring lines between the switch and the CONVERSANT system. Analog tip/ring circuit cards must be installed in the CONVERSANT system with each line connected separately. See Chapter 2, “Hardware,” of *CONVERSANT System Version 8.0 System Description*, 585-313-219, for information on tip/ring circuit card capabilities for ASAI.

Line Side Digital Connections

ASAI can also be provisioned with line side FXS T1 or line side FXS E1, which allows digital connections between the CONVERSANT system and the line side of the switch. This type of connection allows the utilization of various switch features that are not compatible with an ordinary T1 trunk connected between the CONVERSANT system and switch. These features include call transfer and call progress tone (CPT) detection, either in conjunction with Full CCA or where an E1/T1 interface circuit card is used for communications.

Analog configurations require 24 separate connections to support an identical configuration provided by one T1 cable or 30 analog connections to compare to one E1 connection. There is also a significant reduction in the number of circuit cards required to support the interface: one E1 circuit card supports the same amount of traffic as five IVP6 circuit cards.

ASAI Administration

Administering the ASAI feature is a four-step process. The following example assumes that you are installing a voice response application with a configuration in which calls placed to an Automatic Call Distributor (ACD) on the switch are directed to (agent) lines on the CONVERSANT system. The CONVERSANT system is used to select a service for the incoming call based on the DNIS, or called number. The service requests the DNIS number and ANI, or calling number, from the ASAI interface and uses this information as part of the service being provided to the caller. To administer the ASAI feature, perform the following steps on the switch and the CONVERSANT system:

1. Administer the MAPD in CVLAN mode. Refer to the *CallVisor PC LAN over MAPD Installation, Administration, and Maintenance*, 555-230-113.
2. Install and administer the Ethernet circuit card. See the “Installing or Replacing Circuit Cards” chapter in the maintenance book for your platform for the procedure. (Station administration is the same for either. See Table 14 on page 36.)

Note: The UCS 1000 has dual, integrated LAN connections on the CPU Complex. Refer to "Installing Base System Software" in *CONVERSANT System Version 8.0 UCS 1000 Maintenance*, 585-313-150, for information on administer the LAN on the UCS 1000.

3. Administer the ACD domain (hunt group) on the CONVERSANT system and the DEFINITY G3 switch.
4. Administer the tip/ring, E1, or T1 telephone lines.
5. Administer the CONVERSANT system agent lines.

Once you have completed these steps, assign services to DNIS numbers. See Chapter 3, “Voice System Administration,” of the *CONVERSANT System Version 8.0 Administration*, 585-313-510, for information on how to assign these services.

Note: The following procedures assume that you have installed the necessary hardware on the CONVERSANT system and the DEFINITY G3 switch. See ASAI Connectivity on page 33 and Appendix B, “Cable Connectivity,” of *CONVERSANT System Version 8.0 New System Installation*, 585-313-149.

Note: The following procedures assume that you have completed the necessary administration on the switch. See the *DEFINITY Communications System Generic 3i Implementation*, 555-230-650, for more information.

Ethernet Administration

With either a new CONVERSANT system installation or an upgrade, you must administer the DEFINITY ACD split to be used for ASAI connectivity between the DEFINITY and the CONVERSANT system. Use the DEFINITY **add station** or **change station** commands to administer the ACD split. See Table 14 for the appropriate values.

Table 14. Administration Field Name and Requirements

Field Name	Required or Optional?	Valid Value
`Extension:`	Required	Whatever fits your dial plan
`Type:`[1]	Required	ASAI
`Port:`	Required	The port that connects to the ASAI line
`Name:`	Optional	Can be used as an identifier
`XID:`[1]	Required	y
`Fixed TEI:`[1]	Required	y
`TEI:`[1]	Required	3
`MIM Support:`[1]	Required	n
`CRV Length:`[2]	Required	2

[1] To match the built-in administration of the Ethernet circuit cards and the ASAI software, this field must have the value indicated.
[2] In some previous releases, the CRV Length field required a value of 1. You must use the value 2 for CONVERSANT system Version 8.0.

Administering the ACD Split Domain

The following information describes how to administer the ACD split domain on the CONVERSANT system and the DEFINITY switch.

On the CONVERSANT System

You must administer the ASAI feature to monitor the ACD hunt group extension and allow the CONVERSANT system to receive information on calls placed to its agent lines. In other words, you must administer the ASAI feature on the CONVERSANT system so that it requests call events (information) from a *domain* on the switch. In this case, the domain is the ACD hunt group or split, which is composed of the CONVERSANT system agent lines. This domain is referred to as the CONVERSANT system ACD domain. You can administer only one CONVERSANT system ACD split domain on the system. Therefore, all CONVERSANT system agent lines must be part of a single ACD split. Figure 14 on page 33 shows this configuration. See Chapter 4, “Feature Packages,” of *CONVERSANT System Version 8.0 Administration*, 585-313-510, to administer the CONVERSANT system ACD Split Domain.

On the DEFINITY G3 Switch

Use the DEFINITY **add hunt group** or **change hunt group** command to administer the Ethernet line. Table 14 on page 36 lists the values required for proper implementation of the DEFINITY G3 switch for the ASAI link. Table 15 shows a typical way of administering a DEFINITY hunt group to be an ACD. Use the following DEFINITY call center documentation to provide administration details: *DEFINITY Communications System Generic 3 Call Vectoring/Expert Agent Selection (EAS) Guide*, 555-230-520.

Table 15. DEFINITY Hunt Group Field Name and Values

Field Name	Contents: Non-EAS	Contents: EAS
`Group Number:`	The number of the hunt group	
`Group Extension:`	The extension to be used as the lead for the hunt group	
`Group Type:`	ucd	
`ACD?`	y	
`AAS?`	n	
`Vector?`	n	y
`Controlling Adjunct:`	none	

Administering the Tip/Ring, LSE1, and LST1 Lines

Note: The UCS 1000 does not support analog connections.

See Chapter 5, “Switch Interface Administration,” of *CONVERSANT System Version 8.0 Administration*, 585-313-510, to administer tip/ring, LSE1, and LST1 lines. To be certain that you select options that are compatible with the DEFINITY G3 switch (only certain versions), select **DEFINITY** in the PBX Defaults screen.

Note: **DEFINITY** is the default setting. Consequently, if you are administering a new system, the lines are configured correctly by default.

Place all the lines into service. To do so, see the information on changing maintenance state in Chapter 3, “Voice System Administration,” of *CONVERSANT System Version 8.0 Administration*, 585-313-510. These lines or channels are referred to in the following text as CONVERSANT system agent lines.

CAUTION:
Do not proceed until the lines are in the inserv state.

Administering the CONVERSANT System Agent Lines

After creating and bringing the ACD split or vector directory number (VDN) domain into service and administering the tip/ring and LST1/LSE1 lines, you must administer and log in to a CONVERSANT system. There are two ways to do this: as an agent extension in an ACD split, or by an agent ID (with optional password) in an EAS environment. This is required if your service is going to use DNIS or the **A_Callinfo** or the **A_Tran** actions described in Chapter 8, “Using Optional Features,” of *CONVERSANT System Version 8.0 Application Development with Script Builder*, 585-313-217. If you do not log in an agent line, the switch ACD does not route any calls to it. (Note that you can still dial the agent line directly, but no call information is available to the service that answers the call. In other words, the **A_Callinfo** action does not return any information for a call that is not routed to the CONVERSANT system by the ACD.) See Chapter 4, “Feature Packages,” of *CONVERSANT System Version 8.0 Administration*, 585-313-510, for how to log in to the CONVERSANT system agent lines. See Chapter 3, “Voice System Administration,” of *CONVERSANT System Version 8.0 Administration*, 585-313-510, to assign DNIS service to channels.

DEFINITY System Planning

DEFINITY system planning involves defining what changes you must make to the DEFINITY software configuration and ACD environment to support the planned applications. The following is a list of items to consider when planning for the changes.

- Call vectoring is strongly recommended for use in implementing all CONVERSANT system ASAI applications. This is especially true for data screen delivery applications that involve agent-to-agent transfers or DNIS service and for voice response applications that make use of DNIS service.
- Call vectoring is mandatory for routing applications. Call vectoring is also mandatory for data screen delivery applications that make use of call prompting information. Note that the call prompting capability of vectoring is an additional, optional feature over and above the optional call vectoring feature.
- If feasible, you might want to aggregate agents currently in multiple splits into a single split. This minimizes the number of domains that the CONVERSANT system monitors and allows agents to be used more efficiently. Since DNIS is available in call events, single split of agents can handle several applications. The host application can use DNIS to provide information screens that tell agents how to answer and handle calls.

ASAI Application Development Issues

Access to ASAI capabilities is provided through the high-level Script Builder application generation language. Subsets of the Notification, Third Party Call Control, and Routing capabilities of ASAI are integrated into Script Builder for use in ASAI applications.

Note: The CONVERSANT system ASAI feature does not provide access to the Set Value, Value Query, Request Feature, and Third Party Domain Control capabilities of ASAI. The Request Feature capability, however, is used internally by the CONVERSANT system ASAI feature to log tip/ring, LSE1, or LST1 channels in and out of an ACD split on the DEFINITY G3 switch.

The following application development issues must be considered when implementing the ASAI feature with the CONVERSANT system:

- Types of ASAI applications
- Using ASAI versus the converse vector step (CVS)
- Using ASAI in a call center
- CONVERSANT system script design
- Call-flow design
- Host-application design

ASAI Application Types

The capabilities provided by the ASAI feature support three classes of applications:

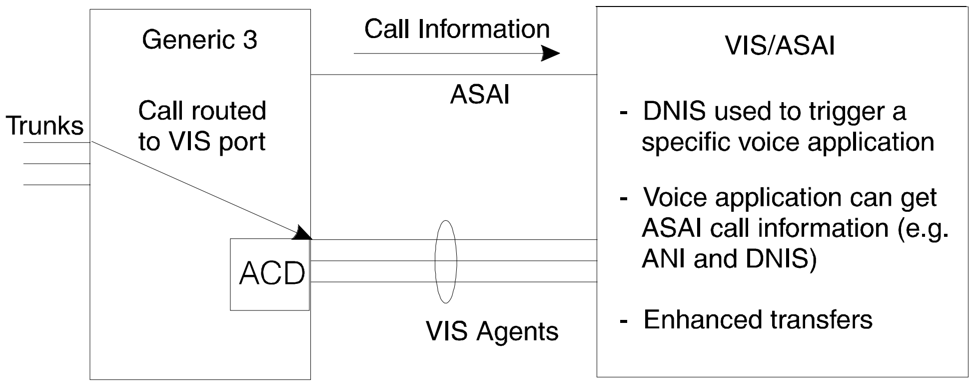
- Voice response applications
- Routing applications
- Data screen delivery applications

These classes of applications can all run simultaneously on a CONVERSANT system. This implies that a CONVERSANT ASAI system provides coresident voice response and DEFINITY G3 switch-to-host gateway capabilities. A single call, for instance, can first be routed by the CONVERSANT system, handled with a voice response application on the CONVERSANT system, and then be monitored by the same system as the call is ultimately delivered to a live agent. Furthermore, integration of the voice response and gateway capabilities allows agents to interact with callers based on the data collected in a voice response script through a host screen. The delivery of a data screen to an operator that contains information about the incoming caller is called a screen pop.

ASAI Voice Response Applications

In voice response applications using the ASAI feature, incoming calls can be routed to the CONVERSANT system over tip/ring, LSE1, or LST1 channels via an ACD split on the DEFINITY G3 switch. Figure 16 on page 40 shows this class of application.

Figure 16. ASAI Voice Response Applications



As a call is delivered to the CONVERSANT system, it receives ASAI information related to the call through the Ethernet LAN circuit card in the CONVERSANT system. ASAI allows it to receive the DNIS and/or ANI information of an incoming call to an analog tip/ring or digital LSE1 or LST1 line over this D-channel. The DNIS and ANI information can be used to control the voice application used for the call. The ASAI information related to the call is made available to the specific voice application that interacts with the caller. In addition, the call control capabilities of ASAI can be used to transfer the call away from the CONVERSANT system if the caller needs to speak to a live agent. The ASAI feature provides the following for voice response applications:

- Channel sharing — The DNIS and/or ANI information associated with the incoming call is used to select a particular Script Builder script to service the call. This allows tip/ring, LSE1, and LST1 ports to be shared across many applications. With port sharing, the same number of ports can handle more calls while maintaining the same grade of service. Alternatively, the same number of calls can be handled at a higher grade of service.
- ANI service — Providing this service allows scripts to be customized according to the calling party number or a range of numbers, for example, an area code.
- Call information — Once the call is answered by the CONVERSANT system, the ASAI information related to the call (such as ANI and DNIS) can be retrieved for use in the voice script that is handling the call.
- Enhanced transfer — The use of ASAI call control capabilities allows the transfer to be faster, quieter from the caller’s perspective, and more reliable. In addition, the G3 ASAI feature of direct agent calling can be used to transfer the call. This allows the call to be delivered to a specific agent while maintaining accurate ACD split statistics. Calls placed to specific agents without the direct agent calling feature do not count as ACD calls in calculating and reporting ACD split statistics. Finally, data that is captured in the voice script can be saved and associated with the transferred call. This enables a host application to deliver data screens to agents that are based on data collected by the voice script that

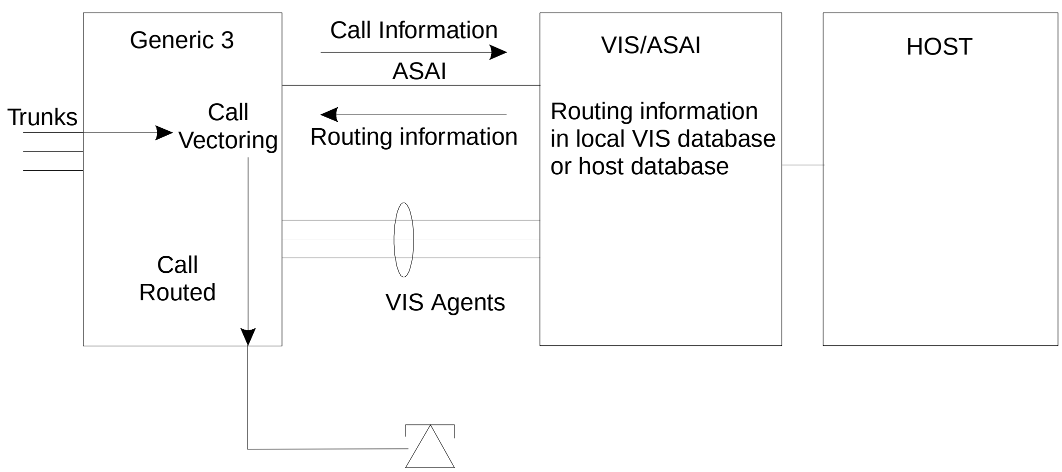
previously serviced the caller and any combination of ANI and/or DNIS information. See Data Screen Delivery Applications on page 42 for more information. The availability of ANI for script selection or within the voice script permits the design of unique voice response applications. Examples include:

- Locator service. A local or host database can be used to determine the closest car dealers, ATMs, stores, and so on.
- Weather reports. A weather report for the caller's area can be provided.
- Pay-per-view. A cable company can use ANI to automate customer selection and billing of pay-per-view programs.
- Caller-dependent transfers. The full 10-digit ANI can be used to identify callers and determine where they should be transferred if they need to speak to a live agent. This is desirable if, for instance, the caller is a preferred customer or is usually handled by a specific agent.
- Geographically based call transfers. The area code and/or exchange can be used to determine where callers should be transferred if they need to speak to a live agent. This is desirable if, for instance, agents handle calls from specific geographic regions.

Routing Applications

In routing applications using the ASAI feature, the CONVERSANT system is used as a routing server to support the routing capabilities of ASAI and the call vectoring feature on the DEFINITY G3 switch. Figure 17 shows how a routing application on the CONVERSANT system receives and responds to call routing requests that are sent by the DEFINITY G3 switch. The application uses routing information provided by the CONVERSANT system to direct the call to a live agent or to a CONVERSANT system agent via either a tip/ring, LSE1, or LST1 connection.

Figure 17. ASAI Routing Applications



The DEFINITY G3 switch generates these call-routing requests when a call is processed by specific call vectors on the switch.

Information as to where to route calls can reside on the CONVERSANT system in a local database or can be provided by a host to which the CONVERSANT system is connected. Call routing is typically based on ANI or call-prompting data collected by the DEFINITY G3 switch.

The use of routing capabilities can significantly improve the efficiency of a call center as shown in the following examples:

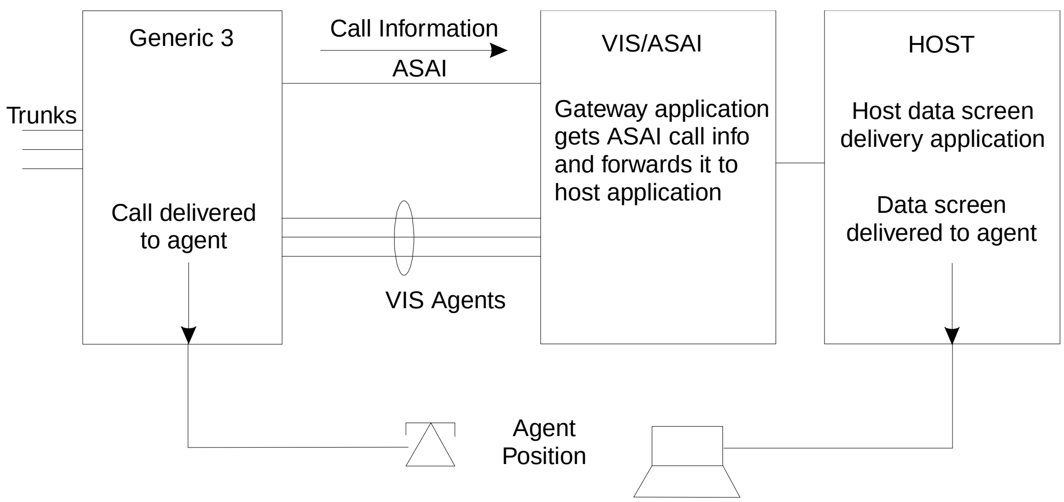
- Priority service — Important or “priority” callers such as major clients can be routed to a common agent group but queued at a higher priority so that they are serviced faster. These callers can also be routed to the specific agent who normally handles their transactions.
- Call redirection — Callers dialing into a particular call center application can be redirected to other call center applications. For example, callers who have delinquent accounts can be redirected to a collections department when they call a sales department.
- Call screening — Fraudulent callers can be disconnected before being connected to an agent so that no network costs are incurred and no agent time is wasted.
- Geographically-based service — Where service is provided on a regional basis, callers can be routed to the agent group that is responsible for their region.

Data Screen Delivery Applications

In data screen delivery applications, an application that resides on the host delivers a specified data screen related to a caller or dialed number to an agent at the same time that a voice call is delivered to the agent’s telephone. This reduces both the agent time and network time required to service the caller. Figure 18 shows a data screen delivery application.

Note: Data screen delivery applications are also known as *coordinated voice/data screen delivery* or *screen pop* applications.

Figure 18. Data Screen Delivery Applications



Note that the delivery of data screens is not a function of the CONVERSANT system itself. The system acts only as a communications gateway between the DEFINITY G3 switch and the host computer. A monitoring application on the CONVERSANT system provides the ability to track the status of calls on the switch. This monitoring application receives information about calls delivered to live agents and forwards this information to the application on the host. The host application in turn uses this information to deliver a data screen to the agent receiving the call.

The information made available to the host includes which agent receives a particular call and the ASAI information associated with the call, such as ANI, DNIS, and any DEFINITY G3 switch call-prompting information collected from the caller. In addition, the call may have been serviced by a CONVERSANT system voice script and then transferred to a live agent. In this case, information collected in the voice script can be saved and passed to the host at the time the call is delivered to the agent. Monitoring applications on the CONVERSANT system can therefore be used to support data screen delivery for three different call-flow scenarios:

- CONVERSANT system-to-agent transfers — In this scenario, calls are delivered to the system and then transferred to a live agent. As described previously, data screens delivered to agents in this scenario can be based on information collected in a voice script in addition to ASAI information such as ANI and DNIS and call-prompting information collected by the DEFINITY G3 switch.
- Incoming call directly to agent — In this scenario, incoming trunk calls are delivered directly to live agents. Data screens delivered to agents are based primarily on ANI and DNIS and/or call-prompting information. Data screens are not based on data collected in a voice script, since a voice script is not used to collect data from the caller.
- Agent-to-agent transfers — In this scenario, calls are transferred between live agents. Here, for example, “screening” agents can be used to collect information from the caller and handle simple transactions. The call can subsequently be transferred to “specialized” agents to handle more complex or detailed transactions. In these scenarios, data screens can be based on information keyed in to the host application by live agents. The host application can save data collected and entered by a screening agent and then use this data as the basis for data screens delivered to specialized agents who can receive the call. Note that the information available for the other two scenarios (that is, ANI, DNIS, call-prompting information, and voice-script data) is also available in this scenario. This information can be used in conjunction with data entered by a live agent to provide the basis for data screens.

Note: You must plan your call flows carefully if you are using multiple ASAI adjuncts with the same DEFINITY G3 switch. Once a call is monitored by a particular CONVERSANT system, the call cannot be redirected or transferred to a domain that is monitored by another system or ASAI adjunct. This is a consideration primarily for data screen delivery applications. For example, if you have agent-to-agent transfers for data screen delivery applications, agents must restrict transfers to domains monitored by the same CONVERSANT system that monitors calls delivered to them. Also, for example, you might have CONVERSANT system-to-agent transfers to support data screen delivery based on data collected by the CONVERSANT system. In this case, you should configure multiple CONVERSANT systems to “front end” mutually exclusive sets of live agents. These considerations do not apply if you are using only one CONVERSANT ASAI system and it is the only ASAI adjunct.

The CONVERSANT system-to-agent transfer scenario described above is supported using the enhanced-transfer capability provided for ASAI voice-response applications. The enhanced-transfer capability allows data collected in the voice script to be saved and associated with the transferred call. Data saved in this fashion can be included in the call-event information that is passed to the host at the time the transferred call is delivered to an agent.

The ability to save voice script data is useful in many ways. A voice script can be used to collect a variety of information such as account number, social security number, personal identification number, desired service, and so on. In many cases, this type of information is more useful than ASAI information such as ANI to both the host application and the live agents handling calls.

The ability to save voice script data with the enhanced transfer capability provides a useful bridge between voice response and data screen delivery applications. It provides true integration (in addition to coresidency) of the voice response and switch-to-host gateway capabilities offered with the CONVERSANT system’s ASAI feature. This mechanism for embedding voice script data in call event information for the transferred call can significantly reduce the complexity of the host application. Without this mechanism, the host application is typically required to associate information from two different physical interfaces (one interface from the voice response unit to receive data collected from the caller and another interface from the monitoring device over which call events are received). Also, the host application is typically required to track and associate multiple events for multiple calls (the initial incoming call to the voice response unit and the second, transferred call that is delivered to an agent). With the ASAI feature, a single message to the host over a single interface provides all the information needed to deliver a data screen based on data collected in a voice script.

ASAI Versus the Converse Vector Step

The CVS allows the switch to maintain control of a call while capabilities of the CONVERSANT system are being used. Whether to use ASAI or the CVS depends on several factors, including cost, traffic, and desired functionality. For example, the CVS feature, used in a script, could support a low-cost ANI routing application. Large volumes of traffic may require an ASAI-based solution due to the more efficient ASAI adjunct routing. See Chapter 5, Converse Vector Step Routing, for more information about the CVS.

The following provides a list of the capabilities and limitations of using the two features on tip/ring, LSE1, or LST1 lines.

- Both ASAI and CVS provide the delivery of ANI, DNIS, and switch call prompting digits for tip/ring, LSE1, or LST1 calls. The CVS provides this information on an in-band basis while ASAI makes the data available on an out-of-band basis. The ASAI out-of-band exchange of data is faster.

Note: CVS allows a maximum of two parameters to be delivered.

- The CONVERSANT system ASAI actions **A_Event** and **A_RouteSel** can be used in monitoring and routing scripts even if the calls are delivered via the CVS.

In addition, both ASAI and the CVS have some unique properties that may influence the decision as to which feature to use:

- ASAI properties
  - ~ When the CONVERSANT system is used as a gateway for switch-to-host applications, the **A_Tran** action simplifies call-flow development using screen pops based on data collected by the CONVERSANT system.
  - ~ Dynamic port allocation is simpler because ANI and DNIS service administration is supported. (Some script programming is necessary if you are using CVS for port allocation. For example, you could write an IRAPI-based start-up script to obtain ANI and DNIS for the CVS interface and then "exec" the appropriate script for that ANI/DNIS information. However, that IRAPI application is not provided with the generic software.
- CVS properties
  - ~ CVS allows a call to remain in a live agent queue while interacting with the CONVERSANT system.
  - ~ Queue position and administered digit string can be passed to the CONVERSANT system using the CVS. Queue position could be used as the basis for an anticipated delay announcement. An administered digit string could be used to identify specific annoucements to be played to callers.

Using ASAI in a Call Center

ASAI can significantly improve the operations in a call center. See also Call Center Features on page 32. This feature provides the following benefits:

- Enhanced customer service

  Caller-dependent and region-dependent treatment for incoming calls is possible in routing and voice response applications. In addition, the direct agent calling feature that is available with these applications allows calls to be delivered to specific agents while maintaining accurate split measurements. These capabilities help to ensure that calls are quickly and reliably directed to the call center resource that is best suited to handle them. This minimizes the number of transfers that a caller experiences and allows callers to be serviced in a rapid, consistent, and personalized fashion.

  In data screen delivery applications, information associated with a given call is available to each agent who receives the call. This eliminates the need for callers to repeat information to each agent. For example, a caller may be directed initially to a CONVERSANT system tip/ring, LSE1, or LST1 channel where the caller is prompted through an automated voice response application. At some point the caller may request to be transferred to a live agent to discuss a topic in more detail. With the CONVERSANT system's ASAI feature, the identity of the caller and additional information collected from the caller by the voice-response application can be saved and presented in a data screen to the live agent receiving the transferred call. This eliminates the need for the caller to repeat information already collected when calls are transferred multiple times or are transferred between live agents. Thus, call holding time is reduced.

- Improved price and performance

  The coresidency of voice response and switch-to-host gateway applications with the ASAI feature eliminates the need for multiple boxes with multiple interfaces to the host computer, thereby simplifying host application development. Access to ASAI capabilities using Script Builder minimizes the effort required to implement the CONVERSANT system's piece of the overall CONVERSANT system/host application. In addition, the use of DNIS in voice response applications to enable tip/ring, LSE1, or LST1 channel sharing means that the same number of CONVERSANT system channels can service more calls.

- Reduced cost of doing business

  Because the host screen application is ready to provide or accept information at the same time that the agent begins to speak with the caller, the use of data screen delivery applications reduces the time needed to service calls. Because calls are shorter, 800-network charges are lower. The same number of agents can handle an increase in call volume since per-call service time is reduced. Also, certain calls can be eliminated entirely via the use of routing applications, for example, call screening for the identification of fraudulent calls.

Specific agent tasks might change when you add an ASAI application such as data screen delivery to the call center. You should determine what agent training is needed before the new service begins. Agents should be trained on what new information will appear on their data terminal screens and how to use that information to interact with calling customers. Before implementing a data screen delivery application with the entire agent population, conduct a trial to compare old call center operations with the new call center operations using a data screen delivery application. Be sure to explain the benefits of the application so that agents can take advantage of them.

If data screen delivery is performed for agent-to-agent transfers, carefully read the information on Agent-to-Agent Transfers on page 56 . Agents must be trained to perform transfers properly so that the desired call events are passed to the host application. More specifically, for blind transfers, agents must transfer calls as follows:

1. Place the original call on hold by hitting the Transfer button once. This also causes a new call appearance to become active (dial tone is heard on this call appearance).

2. Dial the desired extension while hearing dial tone on the new, active call appearance.

3. Immediately press the Transfer button again after dialing the desired extension to complete the transfer.

In *consult* transfer scenarios, the agent may wait to talk to the second agent before completing the transfer. However, the agent must make sure that the original call is on *transfer* hold before completing the transfer. A call is said to be on transfer hold when the call is placed on hold by hitting the Transfer button. This is as opposed to *regular* hold where the call is placed on hold by hitting the Hold button.

For example, the agent may decide to return to the original caller before completing the transfer (for example, to say, “Please wait while I transfer you to Bill who can handle your question”). The agent must be sure to place the original call on transfer hold (not regular hold) before completing the transfer. If the agent used regular hold, the agent would be unable to return to the original caller.

Use the following procedure for consult transfer situations where the screening agent wants to go back and talk to the original caller before completing the transfer. In this procedure, Agent 1 is the screening agent who receives the original call from the calling customer. Agent 2 is the specialized agent who receives the transferred call. Although this procedure may seem cumbersome initially, it is the most natural set of steps to take in consult transfer scenarios where the screening agent wants to announce the transfer to the original caller after having talked to the specialized agent. This procedure also ensures that the CONVERSANT system can properly identify the original call when the two calls are merged. If agents do not follow this procedure, inaccurate call events are reported to the host application.

1. Agent 1 places the original caller on hold by hitting the Transfer button once. This also causes a new call appearance to become active (dial tone is heard on this call appearance).
2. Agent 1 dials Agent 2 while hearing dial tone on the new, active call appearance.
3. Agent 1 places the call to Agent 2 on regular hold by hitting the Hold button while the call to Agent 2 is still the active call.
4. Agent 1 returns to the original caller by pressing the call appearance for the original call. This makes the original call active once again. Agent 1 can now talk to the original caller.
5. After talking to the original caller for the second time, Agent 1 places the original caller on transfer hold again by pressing the Transfer button again. This is the second time Agent 1 has pressed the Transfer button. This causes a third, as yet unused, call appearance to become active. (Dial tone is heard on this call appearance, but this call appearance is not used for anything. Agent 1 goes to the next step and ignores the dial tone).
6. Agent 1 makes the call to Agent 2, which is currently on regular hold, the active call by pressing the call appearance for this call. At this point Agent 1 and Agent 2 are connected again and Agent 1 can inform Agent 2 that the transfer is about to be completed.
7. Agent 1 completes the transfer by hitting the Transfer button again. This is the third time Agent 1 has pressed the Transfer button.

CONVERSANT System Script Design

The CONVERSANT system ASAI feature provides four additional Script Builder actions that are used to access ASAI capabilities.

Note: These actions are discussed in further detail in Chapter 8, “Using Optional Features,” in *CONVERSANT System Version 8.0 Application Development with Script Builder*, 585-313-217.

- **A_Callinfo** — This action is used within a voice response script to retrieve ASAI information about a call delivered to a tip/ring, LSE1, or LST1 channel, for example, calling party number (ANI) and called party number (DNIS) for the call. This action therefore provides access to the Notification capability of ASAI for calls delivered to the CONVERSANT system.
- **A_Event** — This action is used within routing scripts to receive information about call routing requests sent by the DEFINITY G3 switch. This action is also used in monitoring scripts to receive information about calls delivered to an ACD agent. This action therefore serves a dual role by providing access to both the Routing and Notification capabilities of ASAI.
- **A_RouteSel** — This action is used within routing scripts to respond to call-routing requests previously received via the use of the **A_Event** action. This action therefore provides access to the Routing capability of ASAI and allows the CONVERSANT system to send ASAI call routing information to the switch.
- **A_Tran** — This action is used within a voice response script to transfer a call away from a tip/ring, LSE1, or LST1 channel on the CONVERSANT system. This action makes use of the Third Party Call Control capability of ASAI to effect the transfer.

ASAI Voice Script Design

ASAI voice response applications are designed using the **A_Callinfo** and **A_Tran** actions within voice response scripts. Other standard Script Builder actions are also used in the voice script to answer the call, greet the caller, collect data, and so on. See ASAI Application Examples on page 64 for an example of a voice script making use of the **A_Callinfo** and **A_Tran** events.

The **A_Callinfo** and **A_Tran** actions are used only in voice scripts that handle calls delivered to a CONVERSANT system tip/ring, LSE1, or LST1 channel. These two actions are not used in routing and monitoring scripts where, in contrast to voice scripts, a call is not present at a tip/ring, LSE1, or LST1 channel.

For ASAI voice response applications, incoming calls are routed to the CONVERSANT system over tip/ring, LSE1, or LST1 channels configured either as extensions in an ACD split or as agent ID’s under a VDN in an EAS environment on the DEFINITY G3 switch. The CONVERSANT system uses the Notification capability of ASAI to monitor the ACD split or VDN. As a call is offered, the CONVERSANT system receives event reports indicating the status of the call, for example, call offered, queued, alerting, and connected event reports. The CONVERSANT system uses the information contained in these event reports to provide the following capabilities:

- DNIS and ANI service — The DNIS and/or ANI information associated with the incoming call is used to select a particular Script Builder script to service the call. A unique dialed number can be provided for each unique voice response application. Each dialed number is typically represented by a unique VDN on the

DEFINITY G3 switch. Calls to these different VDNs can be routed to the same CONVERSANT system split. The DNIS and/or ANI information associated with an incoming call is then used to select a particular application. An administrative screen on the CONVERSANT system allows the different dialed numbers to be associated with a specific voice response application. This allows tip/ring, LSE1, or LST1 channels to be shared across many applications. Prior to this capability, channels had to be dedicated to specific Script Builder applications.

- Call information — Once the CONVERSANT system answers the call, the ASAI information related to the call can be retrieved for use in the voice script that is handling the call. In particular, the **A_Callinfo** action can be used to obtain ANI, DNIS, switch-collected user data (call prompting digits), call ID, and incoming trunk group ID if ANI is not available.

A user designing a voice script need not be concerned with processing the individual, lower-level ASAI event reports for incoming calls to the CONVERSANT system. Rather, special software is provided as part of the ASAI feature. This software processes the event reports and stores the information contained in these reports on a per-call basis. The DNIS and/or ANI information associated with a call is used to start a specific voice script on the channel that receives the call. The **A_Callinfo** action can then be used within the script to retrieve this information and use it in subsequent Script Builder actions.

A subset of the Third Party Call Control capability of ASAI is also supported for ASAI voice response applications. In particular, the **A_Tran** action uses Third Party Call Control to transfer a call away from the tip/ring, LSE1, or LST1 channel.

The use of the **A_Tran** action within a voice response script invokes the Third Party Call Control operations of third-party take control, third-party hold, third-party make call, and third-party merge. This sequence of ASAI operations invoked with **A_Tran** effects a transfer of the incoming tip/ring, LSE1, or LST1 call to the destination specified with the Destination Number field in **A_Tran**. Hence, the script designer is not required to program many individual ASAI operations. The use of a single action effects the transfer.

Standard switch-hook-flash transfers are still possible when the ASAI feature is used. The use of **A_Tran**, however, provides three significant enhancements over existing transfer mechanisms:

- Transfers are faster, quieter from the caller’s perspective, and more reliable since third-party call control is used rather than the standard switch-hook-flash mechanism.
- The transfer can be completed using direct agent calling. This is done by setting the Destination Number field in **A_Tran** to the desired agent extension and by setting the Split Extension field to the ACD split logged into by the agent. Direct agent calling allows the transfer to be completed to a specific agent while maintaining accurate ACD split measurements. The DEFINITY G3 switch direct agent calling feature can only be invoked via ASAI.
- Information captured in the voice script can be saved for subsequent use in a data screen delivery application. Information assigned to the CONVERSANT system Data field of **A_Tran** is saved by the CONVERSANT system even after the voice script terminates. The CONVERSANT system associates this data with the transferred call and makes this data available in call events passed to the monitoring script that monitors the transferred call.

The third enhancement is very useful for data screen delivery applications where the screens delivered to agents are based on data collected by the CONVERSANT system. Since data collected in a voice script can be saved and is included in call events that are made available to the monitoring script, the host application is simplified. For instance, a CONNECT event (described later) that is made available to the monitoring script contains both the extension of the agent receiving the transferred call and the CONVERSANT system data that was saved from the voice script that previously serviced the caller. This single event is then passed to the host, thereby providing all information needed by the host application in a single message.

Routing Script Design

Routing applications make use of the routing capability supported by ASAI and the call-vectoring feature on the DEFINITY G3 switch. In routing scenarios, calls are not physically delivered to tip/ring, LSE1, or LST1 channels on the CONVERSANT system. Instead, incoming calls to the DEFINITY G3 switch are directed to a vector containing an *adjunct route* step. The adjunct route step causes a *route request* message to be sent to the CONVERSANT system. The route request message contains information pertaining to the call, for example, ANI. The CONVERSANT system uses this information to determine where to route the call.

After the CONVERSANT system determines where to route the call, a *route select* message is sent back to the DEFINITY G3 switch. The route select message contains a destination address provided by the CONVERSANT system that the DEFINITY G3 switch uses to further direct the call. In routing scenarios, the CONVERSANT system can be viewed as a routing server which the DEFINITY G3 switch calls upon to route calls processed with a routing vector.

Note that as a result of routing, the call might be directed to a CONVERSANT system tip/ring, LSE1, or LST1 split to collect more information from the caller. This would be the case, for example, if the information contained in the route request is not sufficient to identify the caller, for example, ANI not recognized.

Routing applications on the CONVERSANT system are supported through the use of routing scripts that are designed using the **A_Event** and **A_RouteSel** actions. The **A_Event** action is used to bring information contained in a route-request message that is sent by the DEFINITY G3 switch up to the script level. The **A_Event** action returns a ROUTE REQUEST event when the DEFINITY G3 switch sends such a message. If no route-request messages are sent, the **A_Event** action waits until it receives one. When a ROUTE REQUEST event is made available to the script, it reflects information in an ASAI route-request message sent by the DEFINITY G3 switch. Note that the **A_Event** action is also used within monitoring scripts to retrieve other types of events as discussed later.

Once a ROUTE REQUEST event is received in a script and the script determines where the call should be routed, the **A_RouteSel** action is used to cause a route-select message to be passed back to the DEFINITY G3 switch. This in turn causes the call to be routed to the desired destination. Unlike voice response scripts, routing scripts are not associated with a particular call. A single routing script handles route requests for many calls. A routing script is designed to receive and process ROUTE REQUEST events. Because these events are controlled by vector processing on the DEFINITY G3 switch, they can arrive at any point in time. Hence, the primary difference between routing scripts and voice response scripts is that once they are activated, routing scripts run continuously. Routing scripts, therefore, have the following general structure:

1. An **A_Event** action to wait for and retrieve a ROUTE REQUEST event from lower-level ASAI software on the CONVERSANT system. Once the **A_Event** action retrieves a ROUTE REQUEST event, the actions below are executed.
2. Other standard Script Builder actions that make use of the data made available in the ROUTE REQUEST event to determine where to route the call. Examples include read table and get/send host screen actions to retrieve routing information from a local or host database.
3. An **A_RouteSel** action to pass the routing information (that is, desired destination) from the script to lower-level ASAI software on the CONVERSANT system. This causes an ASAI route-select message containing the routing information to be sent to the DEFINITY G3 switch.

Steps 1 through 3 are repeated by using additional Script Builder steps to create an infinite loop, that is, script labels and Goto actions. A sample routing script is provided in ASAI Application Examples on page 64.

A routing script may not contain any Script Builder actions that pertain to voice response capabilities (Announce, Prompt and Collect, and so on). A routing script is assigned by using the “RTE” domain designation as described in Chapter 4, “Feature Package Administration,” of *CONVERSANT System Version 8.0 Administration*, 585-313-510.

A routing script can use any of the information returned in the ROUTE REQUEST event. To route the call, see Chapter 8, “Using Optional Features,” of *CONVERSANT System Version 8.0 Application Development with Script Builder*, 585-313-217. Examples include the called-party number (for example, DNIS), calling party number (for example, ANI), and switch data (that is, call prompting information). Any one or combination of the data items returned in a ROUTE REQUEST event can be used as the basis for a routing decision.

The call is routed to the destination that is supplied in the Destination Number field of **A_RouteSel**. The destination can be on-switch (for example, station, ACD split, or VDN) or off-switch (for example, Direct Distance Dialing [DDD] number). Also, the call can be routed to a specific agent within an ACD split (direct agent routing). To do this, set the Destination Number field in **A_RouteSel** to the desired agent extension and the Split Extension field to the split logged into by the agent. Direct-agent routing is the preferred way to route calls to specific ACD agents since direct-agent calls are included in the calculations for ACD split statistics, for example, average speed of answer.

Monitoring Script Design

Monitoring scripts on the CONVERSANT system are used to support data screen delivery applications. The Notification capability of ASAI is used to track the progress of calls that are delivered to agents. A monitoring script on the CONVERSANT system receives information about these calls and forwards this information to a host application. The host application in turn uses the information to format a data screen that is presented to agents receiving calls. Note, therefore, that the delivery of data screens is not a function of the CONVERSANT system itself.

In data screen delivery applications, calls are not physically delivered to a tip/ring, LSE1, or LST1 channel on the CONVERSANT system. Rather, calls are delivered to ACD agents on the DEFINITY G3 switch. Note, however, that a call may have previously been delivered to a tip/ring, LSE1, or LST1 channel to collect information from the caller.

Events

Use the **A_Event** action to design a monitoring script. When used in monitoring scripts, the **A_Event** action returns the following types of call events:

- CONNECT Event — This event indicates that a monitored call is being delivered to an agent.
- ABANDON Event — This event indicates that a monitored call has been abandoned. ABANDON events are passed to a script whenever a caller hangs up before being connected to an agent.
- END Event — This event indicates that a monitored call has ended normally, that is, not abandoned.

Detailed information about the data made available in these events is discussed in Chapter 8, “Using Optional Features,” of *CONVERSANT System Version 8.0 Application Development with Script Builder*, 585-313-217. The three call event types passed to a monitoring script reflect information contained in ASAI event reports for the call.

General Structure

Unlike voice response scripts, monitoring scripts are not associated with a particular call. A single monitoring script handles call events for all of the calls that are delivered to a particular domain. A monitoring script is designed to receive and process call events that can arrive at any point in time as determined by how and when calls progress on the DEFINITY G3 switch. Hence, the primary difference between monitoring scripts and voice response scripts is that once activated, monitoring scripts run continuously. Monitoring scripts, therefore, have the following general structure:

1. An **A_Event** action to wait for and retrieve a call event from lower-level ASAI software on the CONVERSANT system. Once the **A_Event** action retrieves a call event, the actions below are executed.
2. Other Script Builder actions used to pass data in the event to a host. Examples include get/send host screen actions to send the data to an IBM host via the standard 3270 interface and a custom external function to pass the data to a custom DIP supporting an asynchronous interface.

Steps 1 and 2 are repeated by using additional Script Builder steps to create an infinite loop (that is, script labels and Goto actions). A sample monitoring script is provided in ASAI Application Examples on page 64.

A monitoring script cannot contain any Script Builder actions that pertain to voice response capabilities (Announce, Prompt and Collect, and so on). To assign a monitoring script, use the “VDN”, “ACD”, or “CTL” domain designation as described in Chapter 4, “Feature Package Administration,” of *CONVERSANT System Version 8.0 Administration*, 585-313-510.

A monitoring script can pass any combination of the three call-event types to a host. In addition, any combination of the data elements returned in a specific call event can be passed to a host. Examples include the called party number (for example, DNIS), calling party number (for example, ANI), and switch data (that is, call prompting information).

If you make changes to an existing monitoring script or add a new monitoring script, you must do one of the following:

1. Disable and then reenable the domain. See Chapter 4, “Feature Package Administration,” of *CONVERSANT System Version 8.0 Administration*, 585-313-510, for the procedure.
2. Stop and restart the voice system to activate the new script. See “Common System Procedures,” in *CONVERSANT System Version 8.0 System Reference*, 585-313-215, for the procedures.

Call-Flow Scenarios

Monitoring scripts on the CONVERSANT system can be used to support data screen delivery for the following three different call-flow scenarios:

- CONVERSANT system-to-agent transfers — In this scenario, calls are initially delivered to the system and then transferred from the CONVERSANT system to a live agent. The transferred call can be monitored with a VDN type or ACD type of monitoring script if the call is transferred to a monitored VDN or ACD split domain. The transferred call can also be monitored with a CTL type of monitoring script that allows the call to be transferred to a nonmonitored domain or individual station. If the Data field of **A_Tran** was used to save voice script data, this data is made available in the CONVERSANT system Data field of call events that are sent to the monitoring script. Hence, data screens that are delivered to agents in this scenario can be based on information that is collected in a voice script in addition to ASAI information such as ANI, DNIS, and call-prompting information that is collected by the DEFINITY G3 switch. See CONVERSANT System-to-Agent Transfers on page 54 for additional design considerations.
- Incoming call directly to agent — In this scenario, monitored VDNs or ACD splits deliver incoming trunk calls directly to live agents. Here, call events are passed to a VDN type or ACD type of monitoring script and contain only ASAI-related information such as ANI, DNIS, and/or call-prompting information. Data screens are not based on data that is collected in a voice script since a CONVERSANT system voice script is not used to collect data from the caller. Since the CONVERSANT system does not service calls in this scenario, no data is present in the CONVERSANT system Data field of call events.
- Agent-to-agent transfers — In this scenario, calls are transferred between live agents. For example, *screening* agents can be used to collect information from the caller and handle simple transactions. The call can subsequently be transferred to *specialized* agents who can handle more complex or detailed transactions. In these scenarios, data screens can be based on information that is keyed in to the host application by live agents. The host application can save the data that is collected and entered by a screening agent and then use this data as the basis for data screens that are delivered to other, specialized agents who can receive the call. The agent-to-agent transfer can be placed to a monitored domain or to an individual station and monitored with a VDN type or ACD type of monitoring script. Note that the call might have first been delivered to the CONVERSANT system and then transferred to an agent prior to the live agent-to-agent transfer. Hence, call events passed to the monitoring script in this scenario can contain the same information that is available for the other two call-flow scenarios. ASAI-related information such as ANI, DNIS, and call-prompting information and

CONVERSANT system Data can be present in call events. This information can be used in conjunction with data that is entered by a live agent to provide the basis for data screens. See Agent-to-Agent Transfers on page 56 for additional design considerations.

Call-Flow Design

CONVERSANT System-to-Agent Transfers

CONVERSANT system-to-agent transfers are accomplished by using the **A_Tran** action within a voice script that is servicing a caller. The use of **A_Tran** invokes ASAI Third Party Call Control operations to transfer a call away from the tip/ring, LSE1, or LST1 channel to which the caller is connected. The caller is transferred to the destination that is identified in the **Destination Number** field of the **A_Tran** action.

The transferred call can be monitored by a monitoring script so that data screen delivery applications can be supported for CONVERSANT system-to-agent transfers. The transferred call can be monitored in two different ways:

- The call can be transferred to a VDN or ACD split domain that is monitored by the CONVERSANT system with a monitoring script. Call events for the transferred call are passed to the script that is monitoring the domain to which the call is transferred.
- The call can be monitored using a CTL type monitoring script as described in Chapter 4, “Feature Package Administration,” of *CONVERSANT System Version 8.0 Administration*, 585-313-510. In this case, the call can be transferred to nonmonitored domains and individual stations. Here, only call events for calls that are transferred from the CONVERSANT system to agents are passed to monitoring scripts. Other direct calls to an ACD split, for example, are not monitored. Therefore, no call events for the direct calls are passed to monitoring scripts.

You can use a combination of the above two monitoring mechanisms on the same CONVERSANT system. See Chapter 4, “Feature Package Administration,” of *CONVERSANT System Version 8.0 Administration*, 585-313-510, for the rules for which a monitoring script receives call events when these two mechanisms are combined

In addition to monitoring the transferred call, the application developer can save data that is collected in the voice script for subsequent use in the data screen delivery application. To do this, use the CONVERSANT system Data field of **A_Tran**. Any data saved that is in this field when the transfer is initiated from the voice script is presented in call events that are passed to the monitoring script that monitors the transferred call. The CONVERSANT system Data field provides storage for 20 characters. Note that multiple data items can be stored in this field. A social security number and PIN number, for example, can be collected in the voice script, concatenated, and then saved in the CONVERSANT system Data field. The monitoring script that receives this data in call events can then unbundle the information for use in data screen delivery when the transferred call is delivered to an agent.

Typical Call Flow for CONVERSANT System-to-Agent Transfers

The following is a typical call flow for a CONVERSANT system-to-agent transfer:

1. A call arrives at a tip/ring, LSE1, or LST1 channel on the CONVERSANT system. The caller is prompted through a voice response script.
2. The caller decides to speak to a live agent after entering an account number. The voice script transfers the call to a live agent group using the **A_Tran** action. The account number that the caller input is saved by using the CONVERSANT system Data field of **A_Tran**. The voice script terminates after the transfer is complete and the tip/ring, LSE1, or LST1 channel is free to handle another call.
3. The transferred call is queued for an available agent. When the call is eventually delivered to an agent, a monitoring script on the CONVERSANT system receives a CONNECT event for the call. The CONVERSANT system Data field of this CONNECT event contains the account number that was previously saved by the voice script. The monitoring script passes the account number along with the connected agent information from the CONNECT event to the host.
4. The host application uses the account number to format a data screen concerning the caller and presents this data screen to the agent who receives the call. The host application does not need to associate multiple calls since all the necessary information for the transferred call is provided in a single CONNECT event.

One CONNECT event is generated for the entire scenario. This is the CONNECT event for the transferred call as it is delivered to the live agent. This CONNECT event contains the CONVERSANT system Data information in addition to ASAI information that is related to the original call to the CONVERSANT system. The ANI and DNIS for the original call prior to the transfer, for example, are reported in this CONNECT event. Also, the **Other Call ID** field contains the call ID of the call that was originally delivered to the CONVERSANT system's tip/ring, LSE1, or LST1 channel. Call events for calls to tip/ring, LSE1, or LST1 channels on the CONVERSANT system are not passed to monitoring scripts. Also, one END event is generated when the call eventually terminates. As with the CONNECT event, the END event contains data that is pertinent to the original call. See Call Flow Examples on page 69 for a detailed call flow example.

Considerations for CONVERSANT System-to-Agent Transfers

Additional considerations for CONVERSANT system-to-agent transfers are as follows:

- In some cases, you might want to collect more data in a voice script than can be stored in the CONVERSANT system Data field. The recommended method for handling this is to save the data collected by the voice script in the host application. Use **A_Callinfo** to retrieve the call ID of the call that is delivered to the tip/ring, LSE1, or LST1 channel. Pass the call ID along with the data to the host from the voice script itself. The host application must buffer the data until the CONNECT event for the transferred call is received. The call ID in the Other Call ID field of the CONNECT event can be used to correlate the two calls.

- The call can be transferred again after having been serviced by the live agent. In this case, an END event is not reported until all transferring is complete and the call terminates normally. As in the case of a single transfer, the END event contains information pertinent to the original call. Rules for how subsequent call events are reported are discussed in Agent-to-Agent Transfers on page 56.
- The discussions on blind and consult transfers (see Agent-to-Agent Transfers on page 56) do not apply to CONVERSANT system-to-agent transfers completed using the **A_Tran** action. Also, the delay needed for agent-to-agent transfers discussed later does not apply to CONVERSANT system-to-agent transfers completed using the **A_Tran** action.
- Transfers away from the CONVERSANT system can still be accomplished by using standard flash transfer mechanisms. This type of transfer, however, precludes the ability to use the CONVERSANT system Data field of the **A_Tran** screen to save voice script data for later use in data screen delivery applications. Also, the host application must view this type of transfer as an agent-to-agent transfer (see Agent-to-Agent Transfers on page 56). Hence, the discussions on blind transfer versus consult transfer and the need to introduce delay for blind transfers from the CONVERSANT system apply.

Agent-to-Agent Transfers

There are two options for call transfer in an agent-to-agent transfer scenario: blind transfer and consult transfer. These two options differ as to when the screening agent (the agent transferring the call) completes the transfer to the specialized agent (the agent receiving the transferred call) by pressing the Transfer button a second time.

- With a *blind transfer*, the screening agent presses the Transfer button a second time immediately after dialing. The screening agent does not talk to the specialized agent before completing the transfer. In addition, a delay is built into call handling so that the call is distributed to a specialized agent after the screening agent presses the Transfer button the second time.
- With a *consult transfer*, the screening agent waits until the specialized agent answers before pressing the Transfer button a second time.This allows the screening agent to talk to the specialized agent before completing the transfer.

Both of these call-transfer options are described in more detail later. To set up either a blind transfer or a consult transfer, it is important to understand two key concepts of how transferred calls are handled on the DEFINITY G3 switch.

Call Monitoring in Transfer Scenarios

The CONVERSANT system monitors VDN or ACD split domains by assigning monitoring scripts as described in Chapter 4, “Feature Package Administration,” of *CONVERSANT System Version 8.0 Administration*, 585-313-510. A call becomes monitored once it enters one of these monitored domains. The CONVERSANT system *must* also monitor all domains to which this call can be directed. Once monitored, therefore, a call remains monitored for the duration of the call even though it can be transferred several times. Once a call becomes monitored, call events are passed to the monitoring script that is assigned to the domain the call entered. A CONNECT event, for example, is passed to a monitoring script when a specific agent is selected to receive the call. The screening agent may transfer calls to other monitored VDN and ACD splits or to individual stations. The original call to the screening agent must be monitored and therefore delivered to the screening agent via a monitored VDN or ACD split.

Call ID Management in Transfer Scenarios

The DEFINITY G3 switch assigns a call ID to each call. The call ID is provided in the Call ID field of call events for the call. In agent-to-agent transfer scenarios, there are multiple calls and, therefore, multiple call IDs as described in the following transfer scenario:

1. The original call is delivered to an agent and is assigned a unique call ID. The agent talks with the caller and decides that the call needs to be transferred to another agent.
2. The first press of a Transfer button places the original call on hold and allows another call to be placed from the transferring telephone.
3. A second call, temporarily independent of the first call, is placed from the transferring telephone. This call is assigned a call ID that is different from that of the original call. If this second call is placed to a monitored domain, the call immediately becomes monitored by the CONVERSANT system and call events can be passed to a monitoring script. If this second call is placed to an individual station, the call does not become monitored until the transfer is complete as described in Step 4.
4. The second press of the Transfer button *merges* the original call which is on hold with the second call and drops the transferring telephone from the resultant call.

The CONVERSANT system is informed about the completed transfer immediately after the merge that occurs in Step 4. It is only after this merge, therefore, that the CONVERSANT system has the ability to associate the two calls.

With a blind transfer, this merge takes place *before* the merged call is delivered to the second, specialized agent. Hence, with blind transfer calls, the CONVERSANT system can include information in the CONNECT event for the merged call that relates to the original call. In particular, the CONVERSANT system retains the call ID of the original call and reports it in the **Other Call ID** field of the call events for the transferred call. This mechanism allows the host application to use call ID to associate the transferred call with the original call.

With a consult transfer, the merge takes place *after* the second call is delivered to the second, specialized agent. In this case, the original call is still on hold at the first agent's telephone when the second call is delivered to the second agent. Hence, for consult transfers, the CONVERSANT system can only provide information that is related to the second call in the CONNECT event for the second call. In particular, the call ID of the original call is *not* reported in the Other Call ID field of the CONNECT event for the second call. The host application must use a mechanism other than call ID to associate the original call with the second call. The alternate mechanism is the CPN information as discussed below.

Blind Transfer

With a *blind transfer*, the screening agent does not talk to the specialized agent before completing the transfer. With this type of transfer, the CONVERSANT system retains the call ID of the original call and reports it in the Other Call ID field of call events for the transferred call. Also, other ASAI information such as ANI and DNIS that is related to the original call is reported in the call events for the transferred call.

A typical call flow for blind transfers is described below. In this call flow, Agent 1 is a live agent in a screening split who transfers calls to specialized agents. Agent 2 is a specialized agent who can either receive calls via a monitored VDN or ACD split or via a regular extension. Calls to Agent 1 in the screening split must be delivered via a monitored VDN or ACD split.

1. A call arrives for Agent 1.
2. Agent 1 answers the call and enters pertinent information about the caller.
3. Agent 1 transfers the call to Agent 2 by pressing the transfer button, dialing the VDN, ACD split, or individual extension and pressing the transfer button again.
4. Agent 1 is finished with the call.
5. The host application uses call ID information reported in CONNECT events to determine which data to display on Agent 2's data-terminal screen. The call ID from the Call ID field of the CONNECT event for the original call matches the call ID provided in the **Other Call ID** field of the CONNECT event for the transferred call.

Two CONNECT events are passed to monitoring scripts for the entire scenario, that is, one for the original call to the screening agent and one for the transfer to the specialized agent. One END event is generated when the call eventually terminates. See Call Flow Examples on page 69 for detailed examples that include complete descriptions of call flows and call-event contents.

The following conditions should be noted for blind transfers:

- The domain that receives the original call and any domains that receive the transferred call must be monitored.
- Calls can be transferred either to a monitored domain or to a station. For a blind transfer to a monitored domain, the following must be considered:
  - ~ The agent must complete the transfer immediately after initiating it by pressing the Transfer button a second time.
  - ~ A delay must be built into the flow of the transfer so that the communications system can recognize the completion of the transfer before the receiving agent is selected for the call. You can create this built-in delay by transferring calls to a VDN. This VDN is associated with a vector that has a "wait" step in it. The vector directs the call to the desired split with a *route to* or *queue to* step.

  For blind transfer to a station, the following must be considered:

  - ~ When an agent in a monitored domain completes a transfer to a station rather than to an ACD split, a CONNECT event is passed to a monitoring script. The agent must initiate and complete the transfer by pressing the Transfer button a second time for the CONNECT event to be passed to the script. The CONNECT event therefore only becomes available to the host application when the agent pushes the Transfer button the second time.
- In call-center operations that use blind transfer, the host application can tag current call data by call ID. The call ID allows the application to determine which data is associated with the call as the call is transferred to a monitored domain or station.

- If for some reason calls are transferred to nonmonitored domains, unexpected operation can result. When the call placed by Agent 1 is not initially monitored, the CONVERSANT system assumes that a transfer to a station is taking place. Hence, two CONNECT events for the transferred call are generated. One CONNECT event is generated when the transfer is completed by Agent 1 and another is generated when the call is actually delivered to Agent 2. Also, the Connected Party Number field of the first CONNECT event for the transferred call identifies the ACD split or VDN extension that was dialed by Agent 1, rather than identifying the extension of Agent 2. Note that the Connected Party Number field of the second CONNECT event for the transferred call identifies the extension of Agent 2.
- The END event that is reported for the transferred call contains information pertinent to the original call. For example, the original ANI for the caller is reported in the Calling Party Number field and the call ID for the original call is reported in the Other Call ID field. Also, an END event is reported for a call only when the call ultimately terminates. An END event is not reported when a call is transferred.
- The call can be transferred again after it is serviced by Agent 2. In this case, an END event is not reported until all transferring is completed and the call terminates normally. As in the case of a single transfer, the END event contains information that is pertinent to the original call. Rules for how subsequent CONNECT events are reported are as described in this chapter and depend on whether the call is transferred to a monitored domain or to a station and whether consult or blind transfer operation is used.

Consult Transfer

With a *consult transfer*, the screening agent talks to the specialized agent before completing the transfer. With this type of transfer, the call ID for the original call is not retained by the CONVERSANT system and is not reported in the **Other Call ID** field of call events for the transferred call.

A typical call flow for consult transfers is described below. In this call flow, Agent 1 is a live agent in a screening split who transfers calls to specialized agents. Agent 2 is a specialized agent who can receive calls via a monitored VDN or ACD split or via an individual station. Calls to Agent 1 in the screening split must be delivered via a monitored VDN or ACD split.

1. A call arrives for Agent 1.
2. Agent 1 answers the call and enters pertinent information about the caller.
3. Agent 1 presses the Transfer button.
4. Agent 1 dials the extension of the monitored domain or station to which the call will be transferred.
5. Agent 1 waits for Agent 2 to answer.
6. Agent 1 and Agent 2 consult privately about the caller.
7. Agent 1 presses the Transfer button again to complete the transfer.
8. Agent 1 is finished with the call.

9 The host application uses calling party information to determine which data to display on Agent 2’s data terminal screen. The extension for Agent 1 is reported in the Calling Party Number field of the CONNECT event for the second call.

Two CONNECT events are passed to monitoring scripts for the entire scenario, one for the original call to the screening agent and one for the call to the specialized agent. One END event is generated when the call eventually terminates.

See Call Flow Examples on page 69 for detailed examples that include complete descriptions of call flows and call-event contents.

The following conditions should be noted for consult transfers:

- With a consult transfer, Agent 1 and Agent 2 generally both view the call data in a private consultation while the caller is on soft hold.
- Calls can be transferred either to a monitored domain or to an individual station. For a consult transfer to a monitored domain, the following must be considered:
  - ~ When Agent 2 is selected to receive the call from Agent 1, a CONNECT event is made available to a monitoring script. Since Agent 1 stays on the line until Agent 2 answers, the two calls have not yet been merged. This implies that the CONNECT event for the second call does not contain information that is pertinent to the first call. The Other Call ID field for the second CONNECT event, for example, is NULL and does not contain the call ID of the first call. Also, for example, the Calling Party Number field contains the extension for Agent 1 and not the ANI for the caller. This is because the second call is viewed as a new, direct call to Agent 2 from Agent 1. The CONVERSANT system cannot assume that the two calls will eventually be merged since in some cases, they will not be. Hence, the two calls cannot be correlated by using call ID from CONNECT events.
  - ~ In this case, the Calling Party Number field of the second CONNECT event should be used to correlate the two calls. This field contains the extension for Agent 1. The host application can assume that Agent 1 is performing a consult transfer. The host application can then retrieve the appropriate data from Agent 1’s data-terminal screen and deliver it to Agent 2’s data-terminal screen. After the two agents consult, Agent 1 can complete the transfer by pressing the Transfer button a second time. No additional CONNECT event is passed to a monitoring script when the transfer is completed.

  For consult transfer to a station, the following must be considered:

  - ~ A CONNECT event for the second call is passed to a monitoring script only after the transfer is completed when Agent 1 presses the Transfer button the second time. This means that when Agent 1 and Agent 2 are talking privately, the host application will not have been notified about the second call to Agent 2. This is because the second call is placed to a station and not to a monitored domain. The CONVERSANT system, therefore, does not receive events for the second call until the two calls are merged. The host application can be programmed to allow the receiving station to query for the data. After the transfer is complete, a CONNECT event is passed to a monitoring script. This CONNECT event contains information that is pertinent to the first call. The Other Call ID field of this CONNECT event, for example, contains the call ID of the original call that was delivered to Agent 1. Also, for example, the **Calling Party Number** field of this CONNECT event contains the ANI of the original caller.

- If for some reason calls are transferred to nonmonitored domains, an unexpected operation can result. When the call to Agent 2 from Agent 1 is not initially monitored, the CONVERSANT system assumes that a transfer to a station is taking place. Hence, the **Connected Party Number** field of the CONNECT event generated when the transfer is completed by Agent 1 identifies the ACD split or VDN extension dialed by Agent 1, rather than the extension of Agent 2.
- The END event reported for the transferred call contains information that is pertinent to the original call. For example, the original ANI for the caller is reported in the **Calling Party Number** field and the call ID for the original call is reported in the **Other Call ID** field. This is true even though the second CONNECT event for consult transfers to monitored domains does not contain information that is pertinent to the original call. This is because the END event is reported after consult transfers to monitored domains are completed; that is, after the two calls are merged and can be associated by the internal software on the CONVERSANT system. Also, an END event is reported for a call only when the call ultimately terminates (that is, an END event is not reported when a call is transferred). These properties for END events allow the host application to consistently use the **Other Call ID** field of END events to identify when and which calls have left the DEFINITY G3 switch entirely.
- The call can be transferred again after it is serviced by Agent 2. In this case, an END event is not reported until all transferring is completed and the call terminates normally. As in the case of a single transfer, the END event contains information that is pertinent to the original call. Rules for how subsequent CONNECT events are reported are as described in this chapter and depend on whether the call is transferred to a monitored domain or to a station and whether consult or blind transfer operation is used.

Host Application Planning and Design

In certain call center environments, the CONVERSANT ASAI system is integrated with a host computer. An application must be provided on the host that works with the CONVERSANT ASAI system. This host software application is not part of the CONVERSANT ASAI product. The host application can use the information it receives from the CONVERSANT ASAI system to do certain functions such as display call information on agent screens or route calls. The host application can also be called upon to provide the basis for an automated voice response application.

In some cases, particularly for voice response applications, the CONVERSANT ASAI system integrates well with an embedded application and hence no changes are required. For routing and data screen delivery applications, however, you will probably have to modify an existing application or provide a new one to accommodate new functionality.

You may have several options for providing this host application. For example, you can develop your own application or modify an existing application to work with the CONVERSANT ASAI system. This is typically done by the customer’s data processing or information systems department. Alternatively, you can purchase a third-party software vendor application that is designed to work with the CONVERSANT ASAI system.

Application development may require significant planning and coordination between different organizations within your company. The telecommunications, call center operations, and data processing organizations are all typically involved in the planning process. Schedules for application development or customization must be coordinated closely with plans to implement the CONVERSANT ASAI system, ISDN services, and any additional communications system ACD features.

The voice response, routing, and data screen delivery applications enabled by a CONVERSANT ASAI system can all potentially make use of ANI information delivered by the network. The use of ANI generates several considerations.

- You should allow for the possibility that the same caller will call from different telephone numbers. The same person, for example, might sometimes call from home and sometimes call from the office. The same database record should be used in both cases. Calls generated from a switch will probably have more than one ANI assignment. This is based on the different trunk groups used to generate the call and the fact that individual trunk circuits sometimes carry different ANI identities.
- You should allow for situations when ANI information is not delivered for a call.
  - ~ In voice response applications, the voice script should provide some sort of default call handling for cases where no ANI is available.
  - ~ In routing applications, the caller could be routed to a CONVERSANT system tip/ring, LSE1, or LST1 split so that additional information can be collected.
  - ~ In data screen delivery applications, an agent can ask the caller for this information.
- You might want to write an ANI learning module to automatically associate new ANI information with existing customer records. Agents and voice response scripts can verify ANI information that is passed by the DEFINITY G3 switch to the CONVERSANT system.
- You should allow for situations where a single ANI is associated with multiple calling customers. More than one customer, for example, can call from the same switch. Examples of how to handle such situations include bringing up a menu from which the agent can choose the appropriate customer and switching to traditional methods for bringing up customer data.

ASAI Voice Response Application Considerations

- Voice response applications can make use of direct agent calling. Calls can be transferred to specific agents within ACD splits after being serviced by a voice response script. In this case, your database must maintain the ACD split extensions that agents are logged into as well as the extensions for the agents themselves.
- If your voice response application involves transfers to live agents, see CONVERSANT System-to-Agent Transfers on page 54 for additional design considerations.

Routing Application Considerations

- Unlike data screen delivery applications, routing applications make use of the host application in an *inquiry/response* fashion. This implies that the addition of a CONVERSANT ASAI routing application to your call center may have little or no impact on the high-level operation of the application. The most significant change to the host application will probably be the information stored in the database. Information as to how calls should be routed must be added to the database if it is not already present. An example is ANI-to-agent and/or ACD split mappings. If feasible, consider using a local CONVERSANT system database to store routing information.
- Routing applications can make use of direct-agent calling. Calls can be routed to specific agents within ACD splits. In this case, your database must maintain the ACD split extensions that agents are logged into as well as the extensions for the agents themselves.

Data Screen Delivery Application Considerations

- Prior to the use of data screen delivery applications, a host application typically waits for input from agents before it performs an operation. Thus, the agent's input generally controls the application. With data screen delivery applications, a new input to the application is provided. While this input enables agents to work more quickly, it means that the host application must be modified. In particular, the host must use the call events provided by a monitoring script on the CONVERSANT system to drive the screens on the agents' terminals. The interface to the system serves as a *control* link while the interface to the agent operates traditionally as an *inquiry/response* link. The interactions between these two properties of the application must be considered carefully.

  Suppose, for example, that a call arrives for an agent before that agent is finished entering data from the previous call. This scenario can be handled in one of two ways:

  ~ Agents can be trained to use Aux Work or After Call Work feature buttons on their telephones to make themselves unavailable for calls until they are finished entering data from the previous call.

  ~ There is typically a point in the application's sequence of operations (for example, base transaction screen) where the agent is waiting for a new call and begins interaction with the application. The application could look for information from the CONVERSANT system only at this point. The agent's telephone will alert the agent to the new call, and the agent can quickly finish work on the previous call. You may want to provide a quick way for the agent to move to this place in the interactions with the application.
- In data screen delivery applications, telephone extensions are used to identify agents who are receiving calls. The host application must therefore be able to associate extensions with particular data terminals. There are three possible ways to do this:

  ~ The agent can be queried for the telephone extension when the application is started. This is the most flexible arrangement and handles the situation where data terminals and terminal IDs are not permanently associated with the same telephone. Agents do make mistakes in providing the telephone extension to the system. You should plan for these occasional mistakes and make sure that agents understand how to use the system properly. Discuss this issue with the person responsible for the company's agent operations.

  - ~ If an agent is always assigned to the same work position and hence, to the same extension, the extension information could be added to an agent profile.
  - ~ If the relationship between data terminals, terminal IDs, and telephones is relatively stable, administration of the host application can maintain a fixed mapping between telephones and terminals.
- The agent screen application should be able to operate even if the CONVERSANT system is not delivering call events. If call information is not being delivered, the appropriate person or the application itself should notify agents that there is a problem and that they should operate in manual mode. The DEFINITY G3 switch continues to deliver calls to agents even if the ASAI link to the CONVERSANT system is down.
- If your call center application involves data screen delivery for CONVERSANT system-to-agent transfers, see CONVERSANT System-to-Agent Transfers on page 54 for additional design considerations.
- If your call center application involves data screen delivery for agent-to-agent transfers, see Agent-to-Agent Transfers on page 56 for additional design considerations.
- Your application should be able to accommodate cases where multiple CONNECT events are received for the same call. This can occur, for example, where direct-agent calling is used. A call might first ring at the initial agent's telephone and then at the telephone of a covering agent if the call is not answered by the initial agent. In this case, two CONNECT events are sent to a monitoring script when CONNECT events are triggered on an ASAI-alerting event report.
- Your application should be able to accommodate cases where the connected party identified in call events is not a known ACD agent. Depending on switch administration and the design of call vectors, calls can be redirected to domains (VDNs or ACD splits) other than the domain to which the call is originally offered. If calls cover or are redirected from a live-agent split to an AUDIX split, for example, call events can identify an AUDIX channel extension as the connected party.

ASAI Application Examples

This section provides the following examples of scripts developed using the ASAI feature on the CONVERSANT system:

- An ASAI voice script that is developed with the **A_Callinfo** and **A_Tran** actions
- A routing script that is developed with the **A_Event** and **A_RouteSel** actions
- A monitoring script that is developed with the **A_Event** action

Sample ASAI Voice Script

The following is an example of an ASAI voice script that is developed with the **A_Callinfo** and **A_Tran** actions.

```
   start:
   # This is a sample voice script making use of the A_Tran
   # action. This script would be used to handle calls at a
   # Tip/Ring channel.
   #
   # In steps 1 through 3, standard Script Builder actions can
   # be used to greet the caller, collect information, etc. In
   # particular, it is assumed that a Prompt and Collect is
   # used to collect an account number which is stored in
   # account_num. A local database is read in an attempt to
   # match the account number the caller provided and the ANI
   # supplied with A_Callinfo. If a match is found, the table
   # provides an agent extension and a split extension which
   # are used to route the call to a specific agent within a
   # split (direct agent routing). If no match is found, the
   # call is routed to a default live agent split.
   #
   # Fields dest_num (agent extension) and split_num (split
   # extension) for direct agent routing are returned from
   # the table when a match is found.
   #
4. External Action: A_Callinfo
       calling: calling_num
       called: called_num
       switchdata: switch_data
       trunkid: trunk_num
       callid: call_id
       cause: callinfo_cause
       Return Field: callinfo_return
5. Read Table
       Table Name: account_db  Search From Beginning
           Field: account = account_num
           Field: ani = calling_num
   #
   # Set defaults in case no match is found in the table:
   # dest_num is set to the default live agent split (split
   # 5678). split_num is set to NULL so that direct agent
   # calling is not invoked.
   #
6. Evaluate
   If $MATCH_FOUND = 0
7.     Set Field Value
           Field: dest_num = “5678”
           Field: split_num = ““
   End Evaluate
   #
   # Transfer the call. Place the account number
   # (account_num) in the visdata field. The ASAI DIP on the
   # VIS saves this data and associates it with the
   # transferred call.  A subsequent CONNECT event reported
   # for the transferred call will contain this data.
   #
8. External Action: A_Tran
       destination: dest_num
       split: split_num
       priority: No
       visdata: account_num
       state: call_state
```

```
        cause: tran_cause
        Return Field: tran_return
    #
    # Note that the CONNECT event is not received in this voice
    # script. Rather, a monitoring script is used to monitor
    # the transferred call and receive the CONNECT event when
    # the transferred call is delivered to an agent.  This
    # allows the Tip/Ring channel to service other calls while
the
    # first, transferred call is queued for an available
    # agent.
    #
9.  Quit
```

Sample Routing Script

The following is an example of an ASAI routing script that is developed with the **A_Event** and **A_RouteSel** actions.

```
    start:
    # This is a sample routing script making use of the
    # A_Event action. This script would be given, via
    # administration, an "RTE" type designation and therefore
    # would receive only route requests (that is, no CONNECT,
    # ABANDON, or END messages would be received or processed
    # by this script).  A local database is used to route the
    # call based on ANI.  A local database is read in an
    # attempt to match the ANI for the call.  If a match
    # is found, the table provides an agent extension and a
    # split extension which are used to route the call to a
    # specific agent within a split (direct agent routing).
    # If no match is found, the call is routed to a default
    # split (for example, to a VIS Tip/Ring split to collect
    # additional information).
    #
    # Fields dest_num (agent extension) and split_num (split
    # extension) for direct agent routing are returned from
    # the table when a match is found.
    #
    begin_loop:
    #
1.  External Action: A_Event
        connected: connect_num
        calling: calling_num
        called: called_num
        switchdata: switch_data
        trunkid: trunk_num
        callid: call_id
        otherid: other_id
        laidisplay: lai_info
        visdata: vis_data
        routingid: routing_id
        cause value: cause
        Return Field: event_return
    End External Action
    #
    # Check to make sure a ROUTE REQUEST was received.
    # If a ROUTE REQUEST was not received, go back and get
    # the next event.
    #
2.  Evaluate
            If event_return != "'R'"
3.      Evaluate
```

```
          If event_return = "'r'"
4.        Modify Table
          Table Name : rtg_err Operation: Add
          Field: clg_num = calling_num
          Field: cld_num = called_num
          Field: err_cause = cause
          Field callid_value = call_id
   #
   #
   #
   #
      Else
5.        Goto begin_loop
      End Evaluate
   End Evaluate
6. Read Table
      Table Name: ani_db Search From Beginning
          Field: ani = calling_num
   #
   # Set defaults in case no match is found in the table:
   # dest_num is set to the default destination (split 1234).
   # split_num is set to NULL so that direct agent calling is
   # not invoked.
   #
7. Evaluate
   If $MATCH_FOUND = 0
8.    Set Field Value
          Field: dest_num = "1234"
          Field: split_num = ""
   End Evaluate
9. External Action: A_Routesel
      destination: dest_num
      split: split_num
      priority: No
      routingid: routing_id
      cause: cause
      Return Field: route_return
   #
   # Repeat the process - go back and get the next event.
   #
10.Goto begin_loop
```

Sample Monitoring Script

The following is an example of an ASAI monitoring script that is developed with the **A_Event** action.

```
start:
# This is a sample monitoring script making use of the
# A_Event action. This script would be given, via
# administration, a "VDN", "ACD", or "CTL" type
# designation.  This script would be used to receive
# information about monitored calls and pass this
# information to a host.  In this type of scenario, the
# A_Event action can be used to receive CONNECT, ABANDON,
# and END events (no ROUTE REQUEST events are received).
# In this example, a subset of the information available
# in CONNECT events is passed to a host via the 3270
# interface.
#
# It is assumed here that the Transaction Base Screen for
# the host application is called "connect_data". This
# screen is assumed to contain fields that are used for
```

```
    # transmitting data obtained through A_Event. When the
    # host receives the filled screen, it responds by sending
    # a different screen, conveniently named the “verify”
    # screen. The “verify” screen is then sent back with the
    # key, PF3, to obtain the Transaction Base Screen,
    # “connect_data”, again.
    #
    begin_loop:
    #
    HOST_UP:
    Event_start:
1.   External Action: A_Event
        Connected_Number: connect_num
        Calling_Party_Number: calling_num
        Called_Party_Number: called_num
        Switch_Data: switch_data
        Call_Id: call_id
        Other_Call_Id: ocall_id
        LAI_Display_Info: lai_info
        VIS_Data: vis_data
        Routing_ID: route_id
        Return Field: event_ret
    #
    # Check to make sure a CONNECT was received since we
    # don’t care about ABANDON’s and END’s in this example
    # application.  If a CONNECT was not received, go back and
    # get the next event.
    #
2.  Evaluate
       If event_ret   != “‘C’”
3.  Goto Event_start
       End Evaluate
    #
    # Send data to the host.  Only connected agent, ANI, DNIS,
    # and VIS data are used in this example application.
    #
    # It is assumed that Aid Key for sending the data to the
    # host is PF3. Note that you have to investigate what Aid
    # Key is appropriate for your host environment.
    #
4.  Send Host Screen
    Send Screen Name: connect_data Use Aid Key: PF3
       Field: connect_number = connect_num
       Field: ani = calling_num
       Field: dnis = called_num
    voice_data = vis_data
5.  Get Host Screen
    For Screen Name: verify
    End Get Host Screen
6.  Send Host Screen
    Send Screen Name: verify Use Aid Key: PF3
7.  Get Host Screen
    For Screen Name: connect_data
    End Get Host Screen
    #
    # Repeat the process - go back and get the next event.
    #
8.  Goto Event_start
    HOST_DOWN:
9.  Goto start
```

Call Flow Examples

This section provides the following examples of data screen delivery call flows and the contents of the call events that result from these call flows:

- Call to agent via ACD split
- Call to agent via VDNs with call prompting
- Call to VDN and abandoned in queue
- Call to VDN and abandoned after agent selection
- Agent-to-agent transfer via VDN and blind transfer
- Agent-to-agent transfer to a station via blind transfer
- Agent-to-agent transfer via VDN and consult transfer
- Agent-to-agent transfer to a station via consult transfer
- CONVERSANT system-to-agent transfer via ACD split

In all call-flow scenarios, it is assumed that CONNECT events are triggered on ASAI *alerting* event reports. Hence, as shown in the scenarios, a CONNECT event is passed to a monitoring script when an agent is selected for a monitored call. An agent is considered to be selected for a call when the agent’s telephone begins ringing or the agent hears a zip tone. CONNECT events can also be triggered on ASAI *connected* event reports. In this case, CONNECT events are passed to monitoring scripts when agents actually answer monitored calls.

In all call-flow scenarios, it is assumed that the incoming call is delivered via an ISDN facility. This implies that the ANI is available in the ISDN SETUP message for the incoming call. If ANI is available, it is reported in call events as depicted in the call-flow scenarios. If ANI is not available, the incoming trunk group ID is reported instead.

Also, since it is assumed that incoming calls are delivered via an ISDN facility, a 10-digit called party number (CPN) is reported in call events. This number corresponds to the CPN that is provided in the ISDN SETUP message for the incoming call. Note that, as depicted in the call-flow scenarios, this number identifies a billing number and not the 800 number that is dialed by the caller. The use of switch administration to modify DNIS digits does not affect the reporting of the CPN for incoming ISDN calls.

Incoming calls can also be delivered via non-ISDN facilities. In this case, ANI is not available, so the trunk group ID is always reported instead. Also, for non-ISDN calls, the CPN identifies the ACD split or VDN extension to which the call is initially directed. Hence, for non-ISDN calls, the use of switch administration to modify DNIS digits can affect the reporting of the CPN. If modified by switch administration, the DNIS digits, as provided by the network, are not reported in the CPN. Rather, the ACD split or VDN extension that results from the modification is provided in the CPN.

Scenarios 6 through 9 discuss agent-to-agent *transfer* calls. Note that the call events generated for agent-to-agent *conference* calls are the same as described in the transfer scenarios. The three functional differences for conference calls are:

- The screening agent uses the Conference button instead of the Transfer button.
- The screening agent stays on the call instead of being dropped off.
- The END event for the call is not generated until all parties disconnect from the call.

Call to an Agent via an ACD Split

A call arrives at the DEFINITY G3 switch and is delivered directly to a monitored ACD split (no vector processing takes place for the call). An agent is assigned to the call, interacts with the caller, and then terminates the call.

Example:

1. A caller calling from the telephone number 303-555-1726 calls a toll-free 800 number that is associated with customer service.

   Calls to that 800 number are presented to a monitored ACD split with the extension 7777.

2. The call is queued to the monitored ACD split 7777.
3. The call is assigned to an agent in that split with the extension 1234.
4. A CONNECT event is passed to the monitoring script for ACD split 7777 with the following information:

Connected Party Number	1234
Calling Party Number	3035551726
Called Party Number	9085557777
Switch Data	
Call Id	101
Other Call Id	
LAI Display Info	
VIS Data	
Routing ID	
Return Field	C

5 When the selected agent completes and disconnects the call, an END event is passed to the monitoring script for ACD split 7777 with the following information:

Connected Party Number	1234
Calling Party Number	3035551726
Called Party Number	9085557777
Switch Data	
Trunk Group Id	
Call Id	101
Other Call Id	
LAI Display Info	
VIS Data	
Routing ID	
Return Field	E

Call to an Agent via VDNs with Call Prompting

A call arrives at the DEFINITY G3 switch and is handled with call vectoring. The initial VDN/vector that processes the call makes use of the call-prompting feature on the DEFINITY G3 switch to collect information from the caller. In particular, the caller is asked to request a service, for example, “press 1 for gizmo service or press 2 for widget service.” The call is then routed unconditionally to a second VDN that is monitored. The vector that is associated with the second VDN queues the call to an ACD split. Agents in this split can handle service calls for both products. The call-prompting information that is collected on the DEFINITY G3 switch can be used to determine which application to start up when the call is delivered to an agent in the common agent group. This allows a single 800 number to be advertised for both products.

Example:

1 A caller calling from the telephone number 303-555-1726 calls a toll-free 800 number that is associated with customer service. Calls to that 800 number are initially handled with a vector that is associated with VDN 7771. VDN 7771 is not monitored. This vector prompts the user to enter a 1 or 2 and then routes the call to VDN 7772 with a “route to” step. In this example, it is assumed that the caller inputs a 1.

2 The call is routed to the monitored VDN 7772. The vector that is associated with VDN 7772 queues the call to an ACD split with a “queue to” step.

3 The call is assigned to an agent in the split with the extension 1234.

4 A CONNECT event is passed to the monitoring script for VDN 7772 with the following information:

Connected Party Number	1234
Calling Party Number	3035551726
Called Party Number	9085557771
Switch Data	1
Trunk Group Id	
Call Id	101
Other Call Id	
LAI Display Info	
VIS Data	
Routing ID	
Return Field	C

5 When the selected agent completes and disconnects the call, an END event is passed to the monitoring script for VDN 7772 with the following information:

Connected Party Number	1234
Calling Party Number	3035551726
Called Party Number	9085557771
Switch Data	1
Trunk Group Id	
Call Id	101
Other Call Id	
LAI Display Info	
VIS Data	
Routing ID	
Return Field	E

Call to a VDN and Abandoned in Queue

A call arrives at the DEFINITY G3 switch and is handled with a VDN or vector. The vector queues the call to an ACD split. The caller abandons the call while it is in the queue and before it is assigned to an agent.

Example:

1. A caller calling from the telephone number 303-555-1726 calls a toll-free 800 number that is associated with customer service. Calls to that 800 number are processed by a vector that is associated with VDN 7771. VDN 7771 is monitored.
2. The vector that is associated with VDN 7771 queues the call to a vector-controlled split with a “queue to” step.
3. The caller abandons the call before an agent is assigned to the call.
4. An ABANDON event is passed to the monitoring script for VDN 7771 with the following information:

Connected Party Number	
Calling Party Number	3035551726
Called Party Number	9085557771
Switch Data	
Trunk Group Id	
Call Id	101
Other Call Id	
LAI Display Info	
VIS Data	
Routing ID	
Return Field	A

Call to a VDN and Abandoned After Agent Selection

A call arrives at the DEFINITY G3 switch and is handled with a VDN or vector. The vector queues the call to an ACD split. The caller abandons the call after is was assigned to an agent but before the agent could answer it.

Example:

1. A caller calling from the telephone number 303-555-1726 calls a toll-free 800 number that is associated with customer service. Calls to that 800 number are processed by a vector that is associated with VDN 7771. VDN 7771 is monitored.
2. The vector that is associated with VDN 7771 queues the call to an ACD split with a “queue to” step.
3. An agent at extension 1234 is selected for the call.
4. The caller abandons the call before the agent at extension 1234 can answer.

5 An ABANDON event is passed to the monitoring script for VDN 7771 with the following information:

Connected Party Number	1234
Calling Party Number	3035551726
Called Party Number	9085557771
Switch Data	
Trunk Group Id	
Call Id	101
Other Call Id	
LAI Display Info	
VIS Data	
Routing ID	
Return Field	A

Note that this is different from the previous scenario where the caller abandons the call while in the queue. Since an agent was selected for the call before it was abandoned, a CONNECT event is passed to the monitoring script. In the previous case where the caller abandons the call while it is in the queue, no agent was selected for the call; therefore, no CONNECT event is passed to the monitoring script before the ABANDON event. In this scenario, where the caller abandons the call after agent selection, the ABANDON event contains the extension of the agent who was selected for the call. This information can be used to cancel the CONNECT event for the call to the agent since the call terminates before the agent can interact with the caller. Alternatively, the host application could simply let the next CONNECT event for the same agent "overwrite" the previous CONNECT event for the call that was abandoned. The next CONNECT event comes when the next call is delivered to the agent.

Note also that this scenario only applies when CONNECT events are triggered on ASAI alerting event reports. If CONNECT events are triggered on ASAI CONNECT event reports, CONNECT events are passed to monitoring scripts only when agents actually answer calls. Consequently, for cases where CONNECT events are triggered on ASAI CONNECT event reports, only an abandon while in the queue situation is possible. An abandon after agent selection situation will never occur or be reported.

Agent-to-Agent Transfer via a VDN and Blind Transfer

A call is delivered to an agent within a screening split. The screening agent transfers the call using a blind transfer to a group of specialized agents. A delay is built into the transfer by having the screening agent place the transfer call to a VDN. The vector associated with this VDN queues the call to the specialized agent group after delaying the call. This delay allows the transfer to be completed before the transfer call is delivered to a specialized agent.

Example

1 A caller calling from the telephone number 303-555-1726 calls a toll-free 800 number that is associated with customer service. Calls to that 800 number are processed with a vector that is associated with VDN 7771. VDN 7771 is monitored.

2 The vector that is associated with VDN 7771 queues the call to the split of screening agents.

3 The call is assigned to an agent in the screening split with the extension 1234.

4 A CONNECT event is passed to the monitoring script for VDN 7771 with the following information:

Connected Party Number	1234
Calling Party Number	3035551726
Called Party Number	9085557771
Switch Data	
Trunk Group Id	
Call Id	101
Other Call Id	
LAI Display Info	
VIS Data	
Routing ID	
Return Field	C

5 The screening agent talks with the caller and determines that a transfer is necessary.

6 The screening agent at extension 1234 presses the Transfer button and dials 7770, which is the extension of a monitored VDN.

7 The vector associated with VDN 7770 delays the call placed by the agent at extension 1234 for 2 seconds with a “wait” step.

8 The screening agent at extension 1234 then presses the Transfer button again. This merges the two calls at the screening agent’s telephone and drops the screening agent from the resultant call. Note that no END event is reported at this time.

9 The vector associated with VDN 7770 queues the resultant call to the group of specialized agents with a “queue to” step.

10 A specialized agent at extension 4681 is selected for the transferred call.

11 A CONNECT event is passed to the monitoring script for VDN 7770 with the following information:

Connected Party Number	4681
Calling Party Number	3035551726
Called Party Number	9085557771
Switch Data	
Trunk Group Id	
Call Id	105
Other Call Id	101
LAI Display Info	
VIS Data	
Routing ID	
Return Field	C

12 The specialized agent at 4681 completes the call and disconnects.

13 An END event is passed to the monitoring script for VDN 7770 with the following information:

Connected Party Number	4681
Calling Party Number	3035551726
Called Party Number	9085557771
Switch Data	
Trunk Group Id	
Call Id	105
Other Call Id	101
LAI Display Info	
VIS Data	
Routing ID	
Return Field	E

Note that for blind transfers to monitored domains as described in this scenario, the second CONNECT event identifies the original call in the **Other Call Id** field. Note also that this CONNECT event contains ASAI information that pertains to the original call, for example, original ANI and DNIS in the **Calling Party Number** and **Called Party Number** fields, respectively. Any LAI display information, VIS data, or switch data associated with the original call is also carried forward.

Agent-to-Agent Transfer to a Station via Blind Transfer

A call is delivered to an agent within a screening split. The screening agent transfers the call using blind transfer to a specialized agent at an individual station.

Example

1. A caller calling from the telephone number 303-555-1726 calls a toll-free 800 number that is associated with customer service. Calls to that 800 number are processed with a vector that is associated with VDN 7771. VDN 7771 is monitored.
2. The vector that is associated with VDN 7771 queues the call to the split of screening agents.
3. The call is assigned to an agent in the screening split with the extension 1234.
4. A CONNECT event is passed to the monitoring script for VDN 7771 with the following information:

`Connected Party Number`	1234
`Calling Party Number`	3035551726
`Called Party Number`	9085557771
`Switch Data`	
`Trunk Group Id`	
`Call Id`	101
`Other Call Id`	
`LAI Display Info`	
`VIS Data`	
`Routing ID`	
`Return Field`	C

5. The screening agent talks with the caller and determines that a transfer is necessary.
6. The screening agent at extension 1234 presses the Transfer button and dials 2022. Extension 2022 identifies an individual station that is associated with a single, specialized agent.
7. The call initiated by the agent at extension 1234 begins ringing at station 2022. Note that no CONNECT event is reported for this call at this time since the CONVERSANT system is not yet monitoring this call.
8. The screening agent at extension 1234 then presses the Transfer button again. This merges the two calls at the screening agent's telephone and drops the screening agent from the resultant call. Note that no END event is reported at this time.

9 A CONNECT event is passed to the monitoring script for VDN 7771 with the following information:

Connected Party Number	2022
Calling Party Number	3035551726
Called Party Number	9085557771
Switch Data	
Trunk Group Id	
Call Id	105
Other Call Id	101
LAI Display Info	
VIS Data	
Routing ID	
Return Field	C

Note that this CONNECT event for blind transfers to stations is not passed to a monitoring script until the screening agent completes the transfer by pressing the Transfer button a second time.

10 The specialized agent at 2022 answers the transferred call and begins interacting with the original caller.

11 The specialized agent at 2022 completes the call and disconnects.

12 An END event is passed to the monitoring script for VDN 7771 with the following information:

Connected Party Number	2022
Calling Party Number	3035551726
Called Party Number	9085557771
Switch Data	
Trunk Group Id	
Call Id	105
Other Call Id	101
LAI Display Info	
VIS Data	
Routing ID	
Return Field	E

Note that for blind transfers to stations as described in this scenario, the second CONNECT event identifies the original call in the **Other Call Id** field. Note also that this CONNECT event contains ASAI information that pertains to the original call, for example, original ANI and DNIS in the **Calling Party Number** and **Called Party Number** fields, respectively. Any LAI display information, VIS data, or switch data that is associated with the original call is also carried forward.

Agent-to-Agent Transfer via a VDN and Consult Transfer

A call is delivered to an agent within a screening split. The screening agent transfers the call using consult transfer to a group of specialized agents.

Example

1. A caller calling from the telephone number 303-555-1726 calls a toll-free 800 number that is associated with customer service. Calls to that 800 number are processed with a vector that is associated with VDN 7771. VDN 7771 is monitored.
2. The vector that is associated with VDN 7771 queues the call to the split of screening agents.
3. The call is assigned to an agent in the screening split with the extension 1234.
4. A CONNECT event is passed to the monitoring script for VDN 7771 with the following information:

`Connected Party Number`	1234
`Calling Party Number`	3035551726
`Called Party Number`	9085557771
`Switch Data`	
`Trunk Group Id`	
`Call Id`	101
`Other Call Id`	
`LAI Display Info`	
`VIS Data`	
`Routing ID`	
`Return Field`	C

5. The screening agent talks with the caller and determines that a transfer is necessary.
6. The screening agent at extension 1234 presses the Transfer button and dials 7772, which is the extension of a monitored VDN.
7. The vector associated with VDN 7772 queues the call to the group of specialized agents.
8. A specialized agent at extension 4440 is selected for the call that was placed by the agent at extension 1234.

9 A CONNECT event is passed to the monitoring script for VDN 7772 with the following information:

Connected Party Number	4440
Calling Party Number	1234
Called Party Number	7772
Switch Data	
Trunk Group Id	
Call Id	105
Other Call Id	
LAI Display Info	
VIS Data	
Routing ID	
Return Field	C

10 The screening agent and the specialized agent talk privately while the original caller is on hold.

11 The screening agent at extension 1234 then presses the Transfer button again. This merges the two calls at the screening agent's telephone and drops the screening agent from the resultant call. The original caller is now connected to the specialized agent at extension 4440. Note that no END event is reported at this time.

12 The specialized agent at extension 4440 interacts with the original caller.

13 The specialized agent at 4440 completes the call and disconnects.

14 An END event is passed to the monitoring script for VDN 7772 with the following information:

Connected Party Number	4440
Calling Party Number	3035551726
Called Party Number	9085557771
Switch Data	
Trunk Group Id	
Call Id	105
Other Call Id	101
LAI Display Info	
VIS Data	
Routing ID	
Return Field	E

Note that for consult transfers to monitored domains as described in this scenario, the second CONNECT event does not identify the original call in the Other Call Id field. Note also that this CONNECT event does not contain ASAI information that pertains to the original call. Only call events passed to a monitoring script after the transfer is completed contain this information, for example, the END event or a CONNECT event for a subsequent blind transfer. Any LAI display information, CONVERSANT system data, or switch data that is associated with the original call is also carried forward and reported in call events that are reported after the transfer is complete.

Agent-to-Agent Transfer to a Station via a Consult Transfer

A call is delivered to an agent within a screening split. The screening agent transfers the call using consult transfer to a specialized agent at an individual station.

Example

1 A caller calling from the telephone number 303-555-1726 calls a toll-free 800 number that is associated with customer service. Calls to that 800 number are processed with a vector that is associated with VDN 7771. VDN 7771 is monitored.

2 The vector that is associated with VDN 7771 queues the call to the split of screening agents.

3 The call is assigned to an agent in the screening split with the extension 1234.

4 A CONNECT event is passed to the monitoring script for VDN 7771 with the following information:

Connected Party Number	1234
Calling Party Number	3035551726
Called Party Number	9085557771
Switch Data	
Trunk Group Id	
Call Id	101
Other Call Id	
LAI Display Info	
VIS Data	
Routing ID	
Return Field	C

5 The screening agent talks with the caller and determines that a transfer is necessary.

6 The screening agent at extension 1234 presses the Transfer button and dials 2022. Extension 2022 identifies an individual station that is associated with a single, specialized agent.

7 The second call that was initiated by the agent at extension 1234 begins ringing at station 2022. Note that no CONNECT event is reported for this call at this time since the CONVERSANT system is not yet monitoring this call.

8 The specialized agent at extension 2022 answers the call from the screening agent at extension 1234.

9 The screening agent and the specialized agent talk privately while the original caller is on hold.

10 The screening agent at extension 1234 then presses the Transfer button again. This merges the two calls at the screening agent’s telephone and drops the screening agent from the resultant call. The original caller is now connected to the specialized agent at extension 2022. Note that no END event is reported at this time.

11 A CONNECT event is passed to the monitoring script for VDN 7771 with the following information:

`Connected Party Number`	2022
`Calling Party Number`	3035551726
`Called Party Number`	9085557771
`Switch Data`	
`Trunk Group Id`	
`Call Id`	105
`Other Call Id`	101
`LAI Display Info`	
`VIS Data`	
`Routing ID`	
`Return Field`	C

Note that this CONNECT event for consult transfers to stations is not passed to a monitoring script until the screening agent completes the transfer by pressing the Transfer button a second time.

12 The specialized agent at 2022 interacts with the original caller.

13 The specialized agent at 2022 completes the call and disconnects.

14 An END event is passed to the monitoring script for VDN 7771 with the following information:

Connected Party Number	2022
Calling Party Number	3035551726
Called Party Number	9085557771
Switch Data	
Trunk Group Id	
Call Id	105
Other Call Id	101
LAI Display Info	
VIS Data	
Routing ID	
Return Field	E

Note that for consult transfers to stations as described in this scenario, the second CONNECT event identifies the original call in the **Other Call Id** field. Note also that this CONNECT event contains ASAI information that pertains to the original call, for example, original ANI and DNIS in the **Calling Party Number** and **Called Party Number** fields, respectively. Any LAI display information, CONVERSANT system data, or switch data that is associated with the original call is also carried forward.

CONVERSANT System-to-Agent Transfer Via an ACD Split

A call is delivered to a CONVERSANT system tip/ring, LSE1, or LST1 channel and serviced by an ASAI voice response application. An account number is collected in the voice script in preparation for a data screen delivery application based on this account number. The call is then transferred to a live agent group.

Example:

1 A caller calling from the telephone number 303-555-1726 calls a toll-free 800 number that is associated with customer service. Calls to that 800 number are processed with a vector that is associated with VDN 7771. VDN 7771 is not monitored.

2 The vector that is associated with VDN 7771 routes the call to the CONVERSANT system tip/ring, LSE1, or LST1 split with a “route to” step. Note that this tip/ring, LSE1, or LST1 split is monitored, but not by a monitoring script used to retrieve call events. This split is monitored internally by the CONVERSANT system to support ASAI voice response applications that make use of the **A_Callinfo** and **A_Tran** actions.

3 The call is answered by a tip/ring, LSE1, or LST1 channel and serviced by a voice response script. No CONNECT event is passed to a monitoring script for the call at this point. Assume, however, that this call is assigned call ID 101. This call ID would be available within the voice script by using the **A_Callinfo** action.

4 The voice script collects an account number from the caller. In this example, it is assumed that the account number is 987654321.

5 The A_Tran action is used within the voice script to transfer the call to the monitored ACD split 7777. The Destination Number field of **A_Tran** is set to 7777 and the CONVERSANT system Data field of **A_Tran** is set to 987654321.

6 When the transfer is executed, the voice script terminates. This allows the tip/ring, LSE1, or LST1 channel to service additional calls.

7 The call queues to the monitored ACD split 7777.

8 An agent at extension 1234 within ACD split 7777 is selected for the call.

9 A CONNECT event is passed to the monitoring script for ACD split 7777 with the following information:

`Connected Party Number`	1234
`Calling Party Number`	3035551726
`Called Party Number`	9085557771
`Switch Data`	
`Trunk Group Id`	
`Call Id`	105
`Other Call Id`	101
`LAI Display Info`	
`VIS Data`	987654321
`Routing ID`	
`Return Field`	C

10 The agent at extension 1234 answers the call and interacts with the caller.

11 The agent at extension 1234 completes the call and disconnects.

12 An END event is passed to the monitoring script for ACD split 7777 with the following information:

Connected Party Number	1234
Calling Party Number	3035551726
Called Party Number	9085557771
Switch Data	
Trunk Group Id	
Call Id	105
Other Call Id	101
LAI Display Info	
VIS Data	987654321
Routing ID	
Return Field	E

Note that for CONVERSANT system-to-agent transfers as described in this scenario, only one CONNECT event is reported to a monitoring script. This CONNECT event is reported when a live agent answers the transferred call. Not also that this CONNECT event contains data in the CONVERSANT system Data field if such data was saved in the voice script by the use of the **A_Tran** action. The CONNECT event also identifies the original call in the **Other Call Id** field and contains ASAI information that pertains to the original call, for example, original ANI and DNIS in the **Calling Party Number** and **Called Party Number** fields, respectively. Any LAI display information or switch data that is associated with the original call is also carried forward.

Note that the call can also be transferred from the CONVERSANT system to a nonmonitored domain or individual station. In this case, the call events are the same as those described in this scenario. The call events, however, are passed to a CTL-type of monitoring script instead of a VDN-type or ACD-split-type of monitoring script. Also, **A_Tran** must be used to ensure that the CTL-type monitoring script receives the call events for the transferred call. See Chapter 8, “Using Optional Features,” of *CONVERSANT System Version 8.0 Application Development with Script Builder*, 585-313-217, for more information.

4 Computer Telephony Integration

Overview

This chapter provides the following information about the Computer Telephony Integration (CTI) feature:

- Description of CTI
- Setup and Administration
- Application Development

Description of CTI

Applications that make function calls to the CTI data integration process (CTI DIP) on CONVERSANT can use a CentreVu Computer Telephony (CVCT) server to control ports on a PBX and to interact with Siebel eBusiness applications that are located on a client connected to the server. CTI is primarily an alternative to the manipulation of a PBX via a direct ASAI connection to the CONVERSANT (for a description of ASAI, see Chapter 3, “Adjunct/Switch Application Interface”, in *CONVERSANT System Version 8.0 Communication Development*, 585-313-220).

One of the main benefits of CTI is redundancy. Up to two extra CVCT servers can be connected to a CONVERSANT as backups to the server handling calls. If connection to the primary CVCT server is lost, then connection is made to a second server; if this connection is lost, then the third server is used.

Full CTI support is for:

- DEFINITY 6.3 or later
- CentreVu CT Release 9.1 Version 1 or later
- Private Data Version 6
- TSAPI Version 2

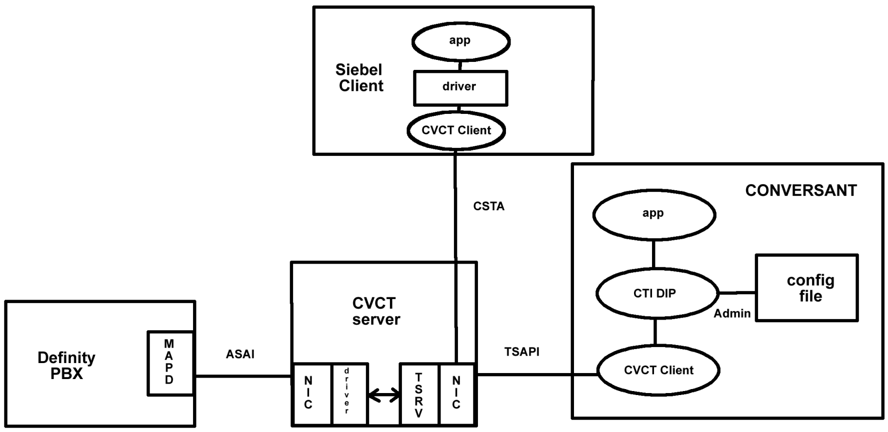
Principles of Operation

CTI operation is shown schematically in "CTI Connectivity" below.

An application on CONVERSANT uses the CTI DIP (administered via a configuration file) to interact with CVCT Client for UnixWare. The CVCT Client sends and receives TSAPI messages over a LAN to a network interface card (NIC) on the CVCT server. This same NIC can be used to communicate with a Siebel eBusiness application residing on a client that uses CVCT Client software and a special driver (CentreVu CT for Siebel Client).

The CVCT server utilizes a telephony server (TSRV) to route messages to the PBX with the aid of a special driver and another NIC. Messages at the Definity PBX are handled by its MAPD circuit card.

Figure 19. CTI Connectivity



Functionality

Function calls to the CTI DIP are designed to do the following on the active ports of a CONVERSANT:

- Put calls on hold
- Retrieve calls from hold
- Disconnect calls
- Transfer two calls
- Conference two calls
- Report information to the application about a call (port extension, call ID, ANI, Called Number)
- Report the state of each call to the application
- Provide answer notification to a voice script
- Dial a call
- Get and report private data (UUI and UCID information) to the application

Call Center Features

Like ASAI, the CTI provides capabilities for use in DEFINITY call center environments.

- Universal Call ID (UCID) — UCID provides a unique identifier (8-byte binary or 20-character ASCII) for every call in a DEFINITY call center customer environment. UCID allows for uniform data-tracking for all call-related data in a call center, regardless of the system. DEFINITY uses the ASAI interface to pass the UCID to the CVCT Server.
- User-to-User Information element (UUI) — UUI allows the customer to specify additional information to be passed in external function arguments, which can contain up to 96 bytes of information (compare with the 32 bytes available using ASAI).

Setup and Administration

Before CTI can be used, the following must be done:

1. "CTI DIP Setup"
2. "Telephony Setup"
3. "CONVERSANT and PBX Administration"
4. CVCT Administration
5. "Setup and Administration of a Siebel Client"

CTI DIP Setup

Setup of the CTI DIP has the following steps:

1. "Installing the CentreVu CT Client"
2. "Installing the CTI DIP"
3. "Set Up the Configuration File"
4. "Verify External Function Files"

Installing the CentreVu CT Client

To learn how to install the CentreVu CT Client for UnixWare, see Chapter 4, "Installing CVCT TSAPI Client Software," in *CentreVu Computer Telephony Release 9.1, Version 1, Telephony Services and CallVisor PC Installation*.

Installing the CTI DIP

The CTI DIP is installed as the optional feature package **ctidip**. If the package has already been installed, there will be a **/vs/data/cti** directory. If the directory does not exist, install the package as follows:

1. Insert the diskette into the diskette drive.
2. At the UNIX prompt, type **pkgadd -d diskette1** and press **ENTER**.

   The screen displays the default, which is to install everything.

3 Press **ENTER** to accept the default.

The installation continues.

4 Remove the diskette from the diskette drive.

For more information about installing packages, see Chapter 7 of *CONVERSANT System Version 8.0 UCS 1000 Maintenance*, 585-313-150.

Set Up the Configuration File

The configuration file, **dialer_cfg.dat**, is in the **/vs/data/cti** directory. It consists of eight required fields which must be in the order listed in the following table. Field values should be on separate lines that do not include comments.

Table 16. CTI DIP configuration file fields

Field	Description
`Debug Flag`	This Boolean flag indicates that the CTI DIP is to provide message tracing in addition to error logging. If this setting is turned off, only error logging will be provided. It is recommended that this field be set to off while the CTI DIP is handling a high volume of traffic. 0 denotes the OFF setting, while 1 denotes the ON setting. The default is 0, OFF.
`System Status Timer Interval`	The value in this field determines how frequently (in seconds) the CTI DIP sends a heartbeat to the switch. This is vital to check if the CVCT server is still in service. Note, however, that system performance may be affected if this value is set to less than 10 seconds. The default value is 10 seconds.
`Error Log File`	This field contains the path to the log file where the CTI DIP will record error and debugging messages. The default value for the error log file is **/vs/data/cti/cdial.log**.
`Login Name`	This field contains the CVCT login id for the CONVERSANT user application. The CTI DIP uses this login, along with the password to access to the CVCT server. The initial value for the login id is `<login name>`.
	1 of 2

Table 16. CTI DIP configuration file fields

Field	Description
`Password`	This field contains the CVCT password for the CONVERSANT user application. The CTI DIP uses this login and password to access to the CVCT server. The initial value for the password is <`password`>.
`Advertised Telephony Service for CVCT server #1`	This field contains a string that identifies Telephony server #1 and the service on that telephony server that the CTI DIP is instructed to use. The Telephony server name or IP address must be in the **/usr/lib/tslibrc file**. The initial value for this field is <`Tserver1`>.
`Advertised Telephony Service for CVCT server #2`	This field contains a string that identifies Telephony server #2 and the service on that telephony server that the CTI DIP is instructed to use. The Telephony server name or IP address must be in the **/usr/lib/tslibrc** file. The initial value for this field is <`Tserver2`>.
`Advertised Telephony Service for CVCT server #3`	This field contains a string that identifies Telephony Server #3 and the service on that telephony server that the CTI DIP is instructed to use. The Telephony server name or IP address must be in the **/usr/lib/tslibrc** file. The initial value for this field is <`Tserver3`>.
	2 of 2

CAUTION:
There can be no empty lines in the file (note that the “#” symbol may be used to turn unused lines into comments).

The following figure shows a sample configuration file.

Figure 20. Sample CTI DIP Configuration File

```
###########################################
# Dialer Client Configuration File
###########################################
#
# must be located in "/vs/data/cti/dialer_cfg.dat"
#
###########################################
# Debug Flag, off = 0, on = 1
#
0
###########################################
# System Status Timer interval (seconds)
#
10
###########################################
# Error Log File
#
/vs/data/cti/cdial.log
###########################################
# CVCT server user login name for application
#
joeblow
###########################################
# CVCT server user password for application
#
125p96
###########################################
# Advertised telephony service for Tserver #1
#
LUCENT#G3_SWITCH#CSTA#IVRINT15
###########################################
# Advertised telephony service for Tserver #2
#
LUCENT#G3_SWITCH#CSTA#WILBUR
###########################################
# Advertised telephony service for Tserver #3
#
LUCENT#G3_SWITCH#CSTA#ROSINANTE
```

Verify External Function Files

Verify that the following external function files are in the **/vs/bin/ag/lib** directory:

- **ctiCallInfo.t**
- **ctiCallState.t**
- **ctiConfer.t**
- **ctiDial.t**
- **ctiDiscon.t**
- **ctiHold.t**
- **ctiNotify.t**
- **ctiPrivData.t**

- **ctiTransfer.t**
- **ctiRetrieve.t**

If any of the files are missing, then reinstall the CTI DIP (see <Link>Installing the CTI DIP on page 89).

Telephony Setup

You must make digital and/or analog telephony connections to the PBX. They are the same as for ASAI.

Analog Tip/Ring Connections

You must install analog tip/ring circuit cards in the CONVERSANT system with each line connected separately. For information on tip/ring circuit card capabilities for ASAI, see Chapter 2, "Hardware," of *CONVERSANT System Version 8.0 System Description,* 585-313-219.

Note: Analog connections are not supported on the UCS 1000.

Line Side Digital Connections

Digital connections between the CONVERSANT system and the line side of the switch are made with either line side FXS T1 or line side FXS E1.

This type of connection allows the use of various switch features that are not compatible with an ordinary T1 trunk connected between the CONVERSANT system and switch. These features include call transfer and call progress tone (CPT) detection, either in conjunction with Full CCA or where an E1/T1 interface circuit card is used for communications.

CONVERSANT and PBX Administration

You must administer the CONVERSANT and the DEFINITY PBX as follows:

1. Administer the MAPD in DLG mode. See *CallVisor PC LAN over MAPD Installation, Administration, and Maintenance*, 555-230-113.
2. Install and administer the CONVERSANT Ethernet circuit card. See the "Installing or Replacing Circuit Cards" chapter in the maintenance book for your platform. Station administration is the same for either.

Note: The UCS 1000 has dual, integrated LAN connections on the CPU Complex. Refer to "Installing Base System Software" in *CONVERSANT System Version 8.0 UCS 1000 Maintenance*, 585-313-150, for information on administer the LAN on the UCS 1000.

3. Administer the tip/ring, E1, or T1 telephone lines on the PBX.

Once you have completed these steps, assign telephone numbers (or extensions) to the ports of the CONVERSANT that will use CTI. See Chapter 3, "Voice System Administration," of the *CONVERSANT System Version 8.0 Administration*, 585-313-510, for information on how to assign these services.

Ethernet Administration

You must administer the DEFINITY for ASAI connectivity between the DEFINITY and the CVCT server.

Use the DEFINITY **add station** or **change station** commands. See the following table for the appropriate values.

Table 17. Administration field name and requirements

Field Name	Required or Optional?	Valid Value
`Extension:`	Required	Whatever fits your dial plan
`Type:`[1]	Required	ADJLK
`Port:`	Required	The port that connects to the ASAI line
`Name:`	Optional	Can be used as an identifier
`XID:`[1]	Required	y
`Fixed TEI:`[1]	Required	y
`TEI:`[1]	Required	3
`MIM Support:`[1]	Required	n
`CRV Length:`[2]	Required	2

[1] To match the built-in administration of the Ethernet circuit cards with the ASAI software, this field must have the value indicated.
[2] In some previous releases,the CRV Length field required a value of 1. You must use the value 2 for CONVERSANT System Version 8.0.

Administering the Tip/Ring, LSE1, and LST1 Lines

See Chapter 5, "Switch Interface Administration," of *CONVERSANT System Version 8.0 Administration*, 585-313-510, to administer tip/ring, LSE1, and LST1 lines. To be certain that you select options that are compatible with the DEFINITY G3 switch (only certain versions), select ***DEFINITY*** in the PBX defaults screen.

Note: The UCS 1000 does not support analog connections.

Note: ***DEFINITY*** is the default setting. Consequently, if you are administering a new system, the lines are configured correctly by default.

Place all the lines into service. To do so, see the information on changing maintenance state in Chapter 3, "Voice System Administration," of *CONVERSANT System Version 8.0 Administration*, 585-313-510.

⚠ CAUTION:
Do not proceed until the lines are in the inserv state.

CVCT Administration

You must administer the CONVERSANT and the CVCT servers to make the CONVERSANT a CVCT client.

CVCT Administration of CONVERSANT

On the CONVERSANT, you must modify the **tslibrc** file located under **/usr/lib/** to include the IP addresses of the CVCT servers.

See the sample file in <Link>Sample tslibrc file on page 95.

Note: Unlike the **dialer_cfg.dat** file, the **tslibrc** file may contain blank lines.

Figure 21. Sample tslibrc file

```
#       /usr/lib/tslibrc

# Blank lines and text beginning with "#" are ignored.

# This is a list of one or more hosts offering Telephony Services via
TCP/IP.
# Either domain name or IP address may be used; default port number is
450
# The form is: host_name [port_number]  For example:

ivrint15.dr.avaya.com   450     # domain name style
135.9.84.33             450     # dotted-decimal IP address

wilbur.dr.avaya.com     450
135.9.84.195            450

rosinante.dr.avaya.com  450
135.9.84.194            450

# replace the above samples with the actual Telephony Server
address(es).

# Individual users may override the contents of this file by setting
# the TSLIBRC environment variable to the pathname of an alternate
# server list (in this same format) or by creating a ".tslibrc" file
```

Administration of the CVCT Servers

Do the following steps on each CVCT server:

1. Create a new user name for the CONVERSANT.
2. Register the user.
3. Create a device for each port of the CONVERSANT that will use CTI.

See *CentreVu Computer Telephony Release 9.1, Version 1, Telephony Services and CallVisor PC Installation* for details on how to administer the CVCT server.

Setup and Administration of a Siebel Client

If a Siebel client will be used, see *CentreVu ® CT Integration for SIEBEL ® eBusiness Applications (Release 1.1, Version 1.2.205) Client Installation Guide*. That guide will provide information and references regarding the related setup and administration of both the client and the CVCT server.

Application Development

This section demonstrates the use of CTI external functions.

Note: For information about using CTI with Siebel eBusiness applications, see *CONVERSANT Version 8.0 Application Development with Siebel eBusiness*, 585-310-784.

External functions

You must invoke CTI external functions to either change the state of a call or to move information about a call.

There are ten external functions available to Script Builder and Voice@Work:

- ctiCallInfo
- ctiCallState
- ctiConfer
- ctiDial
- ctiDiscon
- ctiHold
- ctiNotify
- ctiPrivData
- ctiRetrieve
- ctiTransfer

For details about each of these functions, see *CONVERSANT System Version 8.0 Application Development with Script Builder*, 585-313-217.

Sample application

The use of the CTI DIP will now be illustrated by a very simple, sample application, written in Script Builder. Comments are provided to identify what the application is doing.

```
start:
1.   Answer Phone
2.   External Function
        Function Name: ctiCallInfo
        Use Arguments: cust_callid  ani  calledNum  station_id
skill
        Return Field:  retcode

    #==================================================
    # IF no CTI info, application cannot service call
    # then recall the error message in the log table
    # and quit.
    #==================================================
```

```
3.   Evaluate
     If retcode   <= 0
4.      Set Field Value
              Field: szDescription = "cannot get info on
ctiCallInfo"
              Field: log_type = "CTICALLINFO_ERROR"
5.      External Function
           Function Name: u_datetime
           Use Arguments: date  time  $UNIX_TIME
6.      Modify Table
           Table Name: LOG Operation: Add
              Field: DATE = date
              Field: TIME = time
              Field: CHANNEL = $CHANNEL_NUMBER
              Field: DESCRIPTION = szDescription
              Field: ERRORCODE = log_type

7.      Goto QUIT
     End Evaluate

     #Answer Phone if getting CTI info...

    #================================
    #....Initial Customer Greeting
    #================================

     GREET_CUSTOMER:
8.   Announce
        Speak Without Interrupt
          Text: "Welcome to CTI test Call Center "
          Text: "Please wait while we transfer to our agent "
    #==================================================
    # Hold the current call while waiting to transfer.
    #==================================================

9.  External Function
        Function Name: ctiHold
        Use Arguments: cust_callid  error
        Return Field:  retcode
    #================================================
    # If no CTI hold failed, then recall the error
    # message in the log table
    # and quit.
    #================================================
10.  Evaluate
     If retcode   <= 0
11.     Set Field Value
              Field: szDescription = "cannot place call on
ctiHold"
              Field: log_type = "CTIHOLD_ERROR"
12.     External Function
           Function Name: u_datetime
           Use Arguments: date  time  $UNIX_TIME
13.     Modify Table
           Table Name: LOG Operation: Add
              Field: DATE = date
              Field: TIME = time
              Field: CHANNEL = $CHANNEL_NUMBER
              Field: DESCRIPTION = szDescription
              Field: ERRORCODE = log_type
```

```
14.     Goto QUIT
     End Evaluate

    #=======================
    # Set UUI information
    #=======================

15.  External Action: Concat5
        Destination:  UUI
        String1:  "This is CTI UUI Information."
        String2:  "Your account number is 23456."
        String3:  ""
        String4:  ""
        String5:  ""
        Max_Destination_Length:  255
        Return Field:  retcode
     End External Action

    #=======================================
    # Use ctiDial to pass UUI information
    #=======================================

16.  External Function
        Function Name: ctiDial
        Use Arguments: "5552499"  UUI
        Return Field:  retcode
17.  Evaluate
     If retcode   <= 0
18.     Set Field Value
              Field: szDescription = "cannot place call on
ctiDial"
              Field: log_type = "CTIDIAL_ERROR"
19.     External Function
           Function Name: u_datetime
           Use Arguments: date  time  $UNIX_TIME
20.     Modify Table
           Table Name: LOG Operation: Add
              Field: DATE = date
              Field: TIME = time
              Field: CHANNEL = $CHANNEL_NUMBER
              Field: DESCRIPTION = szDescription
              Field: ERRORCODE = log_type

21.     Goto QUIT
     End Evaluate

    #=========================================
    # Use ctiNotify to get the second callid
    #=========================================

22.  External Function
        Function Name: ctiNotify
        Return Field:  interp_callid

    #=============================================
    # Decide if the call has been delivered
    # If not delivered or originated, then:
    # (1) disconnect the second call,
    # (2) retrieve the first call and try again
    #=============================================
```

```
23.  Evaluate
     If interp_callid   < 0
24.     Set Field Value
              Field: szDescription = "cannot get callid on
ctiNotify"
              Field: log_type = "CTINOTIFY_ERROR"
25.     External Function
           Function Name: u_datetime
           Use Arguments: date  time  $UNIX_TIME
26.     Modify Table
           Table Name: LOG Operation: Add
              Field: DATE = date
              Field: TIME = time
              Field: CHANNEL = $CHANNEL_NUMBER
              Field: DESCRIPTION = szDescription
              Field: ERRORCODE = log_type

   #==============================================
   # Use ctiCallState to get the second call id
   #==============================================

27.     External Function
           Function Name: ctiCallState
           Use Arguments: call2id  call3id  call1state
call2state  call3state
           Return Field:  retcode
28.     Evaluate
     If retcode   <= 0
29.        Set Field Value
                 Field: szDescription = "Cannot get call state
info"
                 Field: log_type = "CTICALLSTATE_ERROR"
30.        External Function
              Function Name: u_datetime
              Use Arguments: date  time  $UNIX_TIME
31.        Modify Table
              Table Name: LOG Operation: Add
                 Field: DATE = date
                 Field: TIME = time
                 Field: CHANNEL = $CHANNEL_NUMBER
                 Field: DESCRIPTION = szDescription
                 Field: ERRORCODE = log_type

32.        Goto QUIT
        End Evaluate

        #Disconnect the dialed call first

33.     External Function
           Function Name: ctiDiscon
           Use Arguments: call2id
           Return Field:  retcode
34.     Evaluate
     If retcode   <= 0
35.        Set Field Value
                 Field: szDescription = "Cannot disconnect the
call"
                 Field: log_type = "CTIDISCONN_ERROR"
36.        External Function
              Function Name: u_datetime
```

```
            Use Arguments: date  time  $UNIX_TIME
37.       Modify Table
            Table Name: LOG Operation: Add
               Field: DATE = date
               Field: TIME = time
               Field: CHANNEL = $CHANNEL_NUMBER
               Field: DESCRIPTION = szDescription
               Field: ERRORCODE = log_type

38.       Goto QUIT
       End Evaluate

   #===========================
   # Retrieve the first call
   #===========================

39.    External Function
          Function Name: ctiRetrieve
          Use Arguments: cust_callid  error
          Return Field:  retcode
40.    Evaluate
     If retcode   <= 0
41.       Set Field Value
                Field: szDescription = "cannot retrieve on
ctiretrieve"
                Field: log_type = "CTIRETRIEVE_ERROR"
42.       External Function
             Function Name: u_datetime
             Use Arguments: date  time  $UNIX_TIME
43.       Modify Table
             Table Name: LOG Operation: Add
                Field: DATE = date
                Field: TIME = time
                Field: CHANNEL = $CHANNEL_NUMBER
                Field: DESCRIPTION = szDescription
                Field: ERRORCODE = log_type
44.       Goto QUIT
       End Evaluate
45.    Announce
          Speak With Interrupt
             Text: "Sorry, all agents are busy now. "
             Text: "Please wait while we transfer to the next
available agent."

   #=============================================
   # Retrieve the first call as an active call
   # Hold the first call again.
   #=============================================

46.    External Function
          Function Name: ctiHold
          Use Arguments: cust_callid  error
          Return Field:  retcode
47.    Evaluate
     If retcode   <= 0
48.       Set Field Value
                Field: szDescription = "cannot place call on
ctiHold"
                Field: log_type = "CTIHOLD2_ERROR"
49.       External Function
             Function Name: u_datetime
```

```
              Use Arguments: date  time  $UNIX_TIME
50.        Modify Table
              Table Name: LOG Operation: Add
                 Field: DATE = date
                 Field: TIME = time
                 Field: CHANNEL = $CHANNEL_NUMBER
                 Field: DESCRIPTION = szDescription
                 Field: ERRORCODE = log_type

51.        Goto QUIT
        End Evaluate
52.     External Function
           Function Name: ctiDial
           Use Arguments: "5552499"  UUI
           Return Field:  retcode
53.     Evaluate
     If retcode   <= 0
54.        Set Field Value
                 Field: szDescription = "cannot place call on
ctiDial"
                 Field: log_type = "CTIDIAL2_ERROR"
55.        External Function
              Function Name: u_datetime
              Use Arguments: date  time  $UNIX_TIME
56.        Modify Table
              Table Name: LOG Operation: Add
                 Field: DATE = date
                 Field: TIME = time
                 Field: CHANNEL = $CHANNEL_NUMBER
                 Field: DESCRIPTION = szDescription
                 Field: ERRORCODE = log_type

57.        Goto QUIT
        End Evaluate
58.     External Function
           Function Name: sleep
           Use Arguments: 2

   #===========================================
   # Use ctiCallState to get second call id
   #===========================================

59.     External Function
           Function Name: ctiCallState
           Use Arguments: interp_callid  call3id  call1state
call2state  call3state
           Return Field:  cust_callid
60.     Evaluate
     If retcode   <= 0
61.        Set Field Value
                 Field: szDescription = "Cannot get call state
info"
                 Field: log_type = "CTICALLSTATE2_ERROR"
62.        External Function
              Function Name: u_datetime
              Use Arguments: date  time  $UNIX_TIME
63.        Modify Table
              Table Name: LOG Operation: Add
                 Field: DATE = date
                 Field: TIME = time
                 Field: CHANNEL = $CHANNEL_NUMBER
```

```
                Field: DESCRIPTION = szDescription
                Field: ERRORCODE = log_type

64.          Goto QUIT
         End Evaluate

     End Evaluate

   #======================================================
   # Use ctiTransfer to merge the first and second calls
   #======================================================

65.  External Function
        Function Name: ctiTransfer
        Use Arguments: cust_callid  interp_callid
        Return Field:  retcode
66.  Evaluate
     If retcode   <= 0
67.     Set Field Value
              Field: szDescription = "cannot transfer on
ctiTransfer"
              Field: log_type = "CTITRANSFER_ERROR"
68.     External Function
           Function Name: u_datetime
           Use Arguments: date  time  $UNIX_TIME
69.     Modify Table
           Table Name: LOG Operation: Add
              Field: DATE = date
              Field: TIME = time
              Field: CHANNEL = $CHANNEL_NUMBER
              Field: DESCRIPTION = szDescription
              Field: ERRORCODE = log_type

70.     Goto QUIT
     End Evaluate
QUIT:

   #================================================
   # Use the ctiCallState to get the state of all
   # active calls on the current CONVERSANT port
   # If any active calls exist, then use ctiDiscon
   # to disconnect the call
   #================================================

71.  External Function
        Function Name: ctiCallState
        Use Arguments: CallID2  CallID3  State1  State2  State3
        Return Field:  CallID1
72.  Evaluate
     If CallID1   != 0
73.     External Function
           Function Name: ctiDiscon
           Use Arguments: CallID1
           Return Field:  RC
     End Evaluate
74.  Evaluate
     If CallID2   != 0
75.     External Function
           Function Name: ctiDiscon
           Use Arguments: CallID2
           Return Field:  RC
```

```
     End Evaluate
76.  Evaluate
     If CallID3   != 0
77.     External Function
           Function Name: ctiDiscon
           Use Arguments: CallID3
           Return Field:  RC
     End Evaluate
78.  Disconnect
79.  Quit
```

5 Converse Vector Step Routing

Overview

This chapter describes the use of the converse vector step (CVS) and the requirements that must be met to implement this interface. It also provides a list of application development issues that you must address when using the CVS.

What is the Converse Vector Step?

The converse vector step (CVS) allows the switch to maintain control of a call while capabilities of the CONVERSANT system are being used. To facilitate this control, the Script Builder conv_data external action supports the DEFINITY G3V2 Voice Response Integration feature (load 04.2.0.096 or greater) on tip/ring, line side T1 (LST1), and line side E1 (LSE1) lines.

Without the use of the CVS, once the call terminates on the CONVERSANT system channel, it is no longer under the control of the switch. The system must process the transaction further and then route the response back to the switch using the Transfer Call action. With the CVS, control over call routing is retained by the switch.

At the beginning of the script, the CVS allows touchtone signals to be passed to the CONVERSANT system. These signals contain information such as the automatic number identification (ANI). At the end of the script, the system can also use touchtone signals to pass back information relevant to further call vectoring, such as a customer account number.

For additional information about the DEFINITY Voice Response Integration feature, see *DEFINITY Communications System Generic 3 Call Vectoring and Expert Agent Selection (EAS) Guide*, 555-230-620.

CVS Provisioning

The following information describes the necessary provisioning for the CVS on the CONVERSANT system and the switch.

Provisioning within the CONVERSANT System

The Converse Data Return (conv_data) action can only be implemented on the tip/ring, LST1, and LSE1 channels. Therefore, the application to be used must be assigned to the appropriate tip/ring, LST1, and LSE1 channels. See the “Installing or Replacing Circuit Cards” chapter in the maintenance book for your platform for information about installing the necessary circuit cards. See Chapter 3, “Voice System Administration,” of *CONVERSANT System Version 8.0 Administration*, 585-313-510, for procedures on assigning service to channels.

The conv_data external action is a part of the base software. The CONVERSANT system’s base software must be installed prior to implementation of the CVS. See the “Installing the Base System Software” chapter of the maintenance book for your platform for the procedure.

Provisioning within the PBX

The Converse Data Return (conv_data) action can only be used when the CONVERSANT system is used with a DEFINITY switch that contains DS1 cards (version 767D or later in switch V5 or later). You must verify the G3V2 switch load prior to implementing the CVS. Failures occur in feature operation unless the G3V2 switch is running load 04.2.0.096 or greater.

CVS Administration

The following information describes the necessary administration of the CVS on the CONVERSANT system and the switch.

Administering within the CONVERSANT System

The conv_data return action executes a switch-hook-flash, and then transmits the digits that are contained in the Feature Access Code (FAC) and Data Return fields for conv_data. Set the duration of this flash at 600 milliseconds in the **Switch Hook Flash Duration** field on the CONVERSANT system. If you are using tip/ring lines, set this value in the Analog Interfaces screen. If you are using LST1 lines, set this value in the Digital Protocol: Line Side T1 - DEFINITY screen. See Chapter 5, “Switch Interface Administration,” of *CONVERSANT System Version 8.0 Administration*, 585-313-510, for the procedures.

Set the Dial Tone Delay in the Digital Protocol: Line Side T1 - DEFINITY screen. Try values between 200 and 1000 milliseconds. Even a value of 1000 miliseconds might not be sufficient for dial tone delay. Dial tone may not occur if your switch does not have enough dial tone detection registers. If you think that lack of dial tone may be a problem, extend this value. Unfortunately, this can cause additional delays in the data return phase and the customer might hear dead air on the line.

Administering within the Switch

If the Converse Data Return action step is implemented on LST1 and LSE1 channels, you must set the Converse First Data Delay parameter on the Systems Parameters Features screen on the DEFINITY switch to 1 instead of to 0 (zero). (The default setting is 0.)

The Feature Access Code field in the conv_data action must match the corresponding FAC code setup on the switch. See the DEFINITY G3V2 Call Vectoring documentation for more information.

CVS Application Development Issues

To use the CVS, you must

- Set up parameters to facilitate data-passing from the switch within the framework of the application being developed
- Define the data-return parameters to enable the CONVERSANT system to send the collected data back to the switch

Script Builder

Use the Prompt and Collect action step and the conv_data external action within an application to implement CVS. See Chapter 7, “Defining the Transaction,” of *CONVERSANT System Version 8.0 Application Development with Script Builder*, 585-313-217, for the procedures to use these actions.

To isolate and resolve problems during the CVS execution, application developers should use the call data event capabilities of Script Builder to log information about return code status. See Chapter 6, “Defining Parameters,” of *CONVERSANT System Version 8.0 Application Development with Script Builder*, 585-313-217, for the procedures to use call data events. This ensures that the Call Data Detail report reflects the outcome of the call using the DEFINITY feature. For example, the return code for conv_data can be stored in a variable and that variable can be one of the logged events in the Call Data Events screen.

Script Language

The **chantype** script instruction allows scripts to determine the type of channel on which they are running. See Chapter 3, “TSM Script Instructions,” and Appendix B, “Summary of TAS Script Instructions,” in *CONVERSANT System Version 8.0 Application Development with Advanced Methods*, 585-313-216, for more information.

Response Application Programming Interface

It is possible to write an Response Application Programming Interface (IRAPI) application that is installed as a start-up service to collect calling party and/or called party information. This application sets the IRD_ANI and/or IRD_DNIS information elements before “exec’ing” the desired application via the number services tables. The **irDial()** and **irGetInput()** functions can be used to exchange data with the switch. For additional information about these capabilities, see *CONVERSANT System Version 8.0 Application Development with Advanced Methods*, 585-313-216.

CVS Versus ASAI

For a discussion of the relative benefits of CVS and ASAI, see ASAI Versus the Converse Vector Step on page 44 in Chapter 3, Adjunct/Switch Application Interface.

CVS Examples

The following are typical ways in which CONVERSANT system applications can use the CVS.

- Port sharing

  By specifying the vector directory number (VDN) in the `data1` field for the CVS on the DEFINITY switch, information that is equivalent to dialed number identification service (DNIS) is available to the CONVERSANT system by way of the first Prompt and Collect action. Based on the VDN, the CONVERSANT system can execute an appropriate script using the Script Builder Execute action. This capability is similar to DNIS on T1 E&M channels. Prior to the CVS, tip/ring channels only had this capability through ASAI.

- Automatic number identification (ANI) – Called Party Number/Billed Number (CPN/BN)

  By specifying ANI in the `data2` field and VDN in the `data1` field for the CVS on the DEFINITY switch, the CPN/BN is available to the executed script by way of the second Prompt and Collect action. This information can be useful in a dealer or locator application.

- System announcement selection

  Hard-coded administered digit strings in the `data1` and/or `data2` fields can be used to instruct the CONVERSANT system to play selected announcements.

- Indication of anticipated delay

  If your CONVERSANT system is used with a DEFINITY G3V4 switch or later, a caller's expected wait time is passed using the keyword *EWT*. If your switch is G3V2 or G3V3, a caller's position in queue is passed using the keyword *qpos*. The CONVERSANT system can play an announcement Informing the caller of an anticipated wait based on either data item.

- ANI/routing

  Based on the CPN/BN, the CONVERSANT system can perform a database operation to determine further routing of the call. For example, in a credit card application, the CPN/BN can map to a premier account holder or a regular account holder. This information can be passed back using the data return string so that the DEFINITY switch can give priority treatment as required. The account number can be directed to appear on the agent's display.

- Enhanced call management system (CMS) call records

  Digit strings passed back to the DEFINITY can be stored in CMS call records to provide further detail as to call dispositions, for example, the number of premier versus regular account calls that are processed by the CONVERSANT system.

6 Call Classification Analysis

Overview

This chapter describes the use of call classification analysis (CCA) and the benefits it provides in analog and digital communications. It also details requirements for implementing this feature and suggested values for telephony parameters when using this feature.

What is CCA?

Call Classification Analysis (CCA) allows application developers to classify the disposition of originated and transferred calls. There are two types of CCA:

- Intelligent — This type of call classification supports call transfers and call bridges. It uses the signaling and tone-detection capabilities of the network interface card that is being used. Intelligent CCA is intended only for use on outbound calls that terminate on the switch or PBX to which the CONVERSANT system is connected.
- Full — This type of call classification provides enhanced capabilities to intelligent call classification. These capabilities include better answer detection, busy and audible ring tone detection, modem tone detection, and so on. Full CCA is offered as an optional feature package. It should be used when outbound calls will terminate beyond the local switch or PBX.

Note: Full CCA is used only in the United States and Canada.

CCA Provisioning

Full CCA requires at least one speech and signal processor (SSP) circuit card to be installed and operational prior to loading the Full CCA software. A single SSP circuit card supports 42 simultaneous channels of CCA. The SSP circuit card must be dedicated to call classification (see CCA Administration on page 112) and connected to the TDM bus. See the “Installing or Replacing Circuit Cards” chapter in the maintenance book for your platform for information on installing the SSP circuit card.

Intelligent CCA on T1 or Primary Rate Interface (PRI) digital lines provides answer and disconnect supervision only. Unless an AYC21 circuit card provides your digital interface, intelligent CCA is not available on Line Side T1 (LST1) lines because there is no answer supervision or dial tone detection.

If you require detection of call progress tones with LST1, you must either install Full CCA or install an AYC21 circuit card. If you require detection of call progress tones with T1 (E&M) or PRI, you must install Full CCA.

Note: Unless an AYC21 circuit card provides your digital interface, LST1 cannot detect dial tone or stutter dial tone prior to dialing, whether or not it is used with the Full CCA feature.

Note: CCA performance may be slightly less if used with analog tip/ring lines instead of digital lines. Analog lines tend to be more noisy than digital lines and may lead to occasional false identification of tones.

Note: Full CCA is not recommended for use with E1 (CAS) or LSE1, those protocols typically used outside the United States and Canada.

CCA Administration

You must assign CCA functionality to the SSP circuit card for the CCA feature to operate properly. See Chapter 3, “Voice System Administration,” of *CONVERSANT System Version 8.0 Administration,* 585-313-510, for the procedure to change the state of the SSP circuit card.

CCA Application Development Issues

This section covers general development issues with CCA and specific issues dealing with the use of CCA and Script Builder, script language, and Response Application Programming Interface (IRAPI) development issues with CCA.

General Issues

An error is generated if a script attempts to use Full CCA and the maximum number of CCA instances are running. The maximum number of CCA instances allowed on the SSP circuit card is 24. No further attempts to use Full CCA are made after the error is logged. See the system message TSM003 in Chapter 4, "Alarm and Log Messages," in *CONVERSANT System Version 8.0 System Reference*, 585-313-215, for more information.

Script Builder

Intelligent CCA or Full CCA can be activated when a call is dialed out during a switch-hook-flash transfer, a call bridge (internal transfer), or a make call (call origination), as defined in Script Builder. The Script Builder actions Transfer Call, Call Bridge, and Make Call use both intelligent and Full CCA. See Chapter 7, “Defining the Transaction,” in *CONVERSANT System Version 8.0 Application Development with Script Builder*, 585-313-217, for additional information.

Note: You must use the Make Call, Transfer Call, and Call Bridge actions to populate the database that is used in generating the Call Classification Report. The Call Classification Report provides information for each extension or number dialed, the total number of calls, and the number of transfer attempts for a specified date. See Chapter 8, “Daily Administration,” of *CONVERSANT System Version 8.0 Administration*, 585-313-510, for information about the Call Classification Report.

Script Language

The following instructions invoke Full CAA through script language:

- **setcca**
- **tic**

This section gives a brief discussion of these two instructions. For detailed information, see Chapter 3, “TAS Script Instructions,” and Appendix B, “Summary of TAS Script Instructions,” of *CONVERSANT System Version 8.0 Application Development with Advanced Methods*, 585-313-216.

setcca

The **setcca** script instruction allows the application developer to set CCA parameters at the script level. These parameters specify the following:

- Whether to use intelligent or Full CCA
- The number of rings to wait for an answer
- Whether to use answer detection or speech-energy detection

tic

The **tic** instruction specifies additional call dispositions for Full CCA if Full CCA is turned on via the **setcca** instruction.

IRAPI

The **irSetParam(3irAPI)** function can be used to set the IRP_OUTCALL_CCALEVEL to IRD_FULL_CCA. This parameter enables Full CCA on a channel for subsequent **irCall(3irAPI)** and **irDial(3irAPI)** function calls.

CCA Example

The following example is an excerpt from a script that shows how an application developer might use the **setcca** and **tic** instructions in a Full CCA application.

```
setcca(im.1,im.10,im.-1)
nextcall:
dbase( .... )     /* get number to dial from DIP */
tic(‘O’, r.3)    /* call number in register 3 */
jmp(r.0 == im.’N’, noAns)     /* no answer after 10 rings */
jmp(r.0 == im.’B’, busy)
jmp(r.0 == im.’F’, retry)
jmp(r.0 == im.’A’, answer)
jmp(r.0 == im.’s’, SIT)
jmp(r.0 == im.-4, noResource)
noAns:
tic(‘h’)     /* put line on-hook to stop ringing */
busy:
dbase ( .... )    /* report result to controlling DIP */
goto (nextcall)
SIT:
jmp(r.1 == im.’R’, retry)
jmp(r.1 == im.’r’, retry)
jmp(r.1 == im.’K’, retry)
jmp(r.1 == im.’k’, retry)
dbase ( .... )     /* report result to controlling DIP */
answer:
talk("Hello, you may be the winner of a free trip to Hawaii")
dbase ( .... )     /* report result to controlling DIP */
goto (nextcall)
```

7 Data Network Communications

Overview

The following data network communication interfaces are available for use in conjunction with the CONVERSANT system Version 8.0 software:

- TN3270E
- TCP/IP
- SQL*NET
- Asynchronous

This chapter provides information on each of these packages, including the configuration and administration procedures.

Host Interface Software

Note: Interface Systems, Inc. (ISI) provides separate documentation on TN3270. See http://www.interfacesystems.com/. Select Cleo products and then Host Interfaces for Avaya.

The host interface is a combination of software, and optionally hardware, designed to allow the transmission of information over a network. This network usually includes remote host computers and/or databases. The host interface software package allows applications running on the CONVERSANT system to send and receive screens from applications running on the host mainframe.

Host Interface Architecture Overview

The Interface Systems, Inc. TN3270 host interface software provides access to IBM Mainframe applications by providing a 3270 Mod 2-5Terminal Emulator and programmable interface to look like an IBM 3270 terminal. The TN3270 software uses a TCP/IP connection to a TNSERVER system that physically connects to the IBM Mainframe.

The Interface Systems, Inc. SNA 3270 host interface software (and optionally Hardware) provides access to IBM Mainframe applications by providing a 3270 Terminal Emulator and programmable interface to look like an IBM 3270 terminal. Additionally, the SNA 3270 host interface emulates an IBM 3274-41C or 3174-01R cluster controller with up to 128 logical units (LUS), that is, 3278 Model 2-5 terminals, connected to it. The connection to the host is typically done by linking to a Front End Processor (FEP) using synchronous data link control (SDLC) protocol (plus special hardware), token ring data streams (plus special hardware), or SNA protocol over LLC2 (no special hardware).

Hardware Environment Architecture

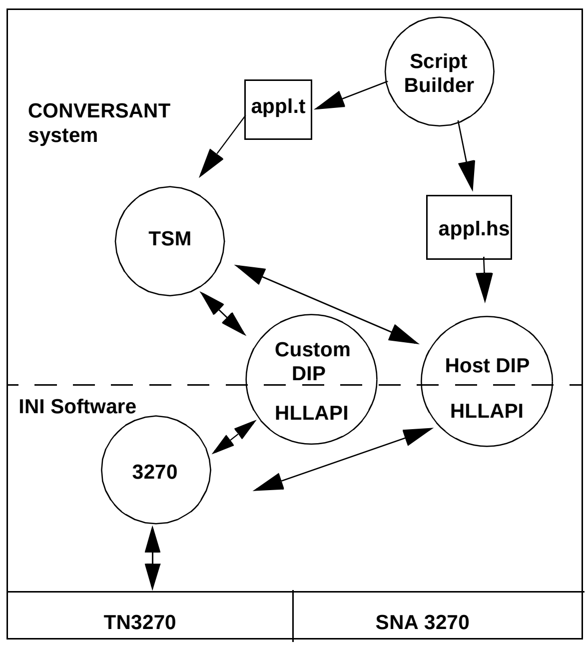
The hardware supported in CONVERSANT V8 platforms depends on the software link. The physical connection uses hardware specific to SDLC, token ring, or Ethernet supported for both the MAP/40P and the UCS 1000. See http://www.interfacesystems.com/. Select Cleo products and then Host Interfaces for Avaya.

Software Process Architecture

In CONVERSANT system Version 8.0 software, the Interface Systems, Inc. TN3270 or SNA 3270 product provides the host interface to the IBM Mainframe.

Figure 22 shows the current software process architecture for the host interface. Note that the dashed line separates the process ownership between the CONVERSANT system and Interface Systems, Inc.

Figure 22. Host Interface Protocol



Host Interface Features

The following are the basic features of the host interface software available with a CONVERSANT system and Interface Systems, Inc. software:

- Script Builder applications interface with host programs

  Script Builder can be used to create an application to interface with a complicated host computer application. The application developer logs in to the host computer and captures screen images. The developer then identifies the screens and fields on those screens that are needed during the transaction. An external function can also be created to allow Script Builder to interface with custom data interface processes (DIPs) that require data communications protocols other than TN3270, SNA 3270, or the High Level Language Application Programming Interface (HLLAPI),

- 3270 Terminal Emulation (for both TN3270 and SNA 3270)

  This capability allows a device or program to imitate another device or program. The 3270 terminal emulation software temporarily transforms itself into a look-alike of an IBM 3270 terminal. In addition to providing full 3270 functionality, the 3270 terminal emulator allows the transfer of files to and from UNIX.

- IND$FILE File Transfer

  The file transfer capability allows you to transfer text or binary files between a mainframe using the IBM host program IND$FILE and your CONVERSANT system.

  FTS can work with multiple IBM mainframe operating environments or processing subsystems. These host systems and their IBM IND$FILE program product numbers include:

  ~ Time Sharing Option (TSO), #5665-311

  ~ Conversational Monitor System (CMS), #5663-281

  ~ Customer Information Control System (CICS), #5789-DQH

  Once installed, file transfer can be initiated interactively through the Terminal Emulator (TE) program or directly from the UNIX command line either by entering the FTS commands or by running a shell script containing the commands.

Note: The host configuration for file transfer must use the write structured fields (WSF), also known as DFT mode, in the logmode table entry. This entry must be verified in both upgrades and new systems because it is a change from previous product releases.

- Enhanced File Transfer

Note: Not currently supported for CONVERSANT V8.

  Enhanced File Transfer uses the file transfer system to automatically transfer files between the CONVERSANT system and a synchronous host processor on a designated LU.

- HLLAPI

  HLLAPI is an application programming interface that allows users to write custom applications that can communicate with the host via an API. See the *Interface Systems, Inc. TN3270 or SNA 3270 HLLAPI Programmer's Guide* for more information.

Software Package Structure

See the Interface Systems, Inc. *TN3270* or *SNA 3270 Quick Start Guide* for specific software package structure information.

Administration Interfaces

The 3270 Synchronous Communications software can be administered from either the screen interface or the command line interface. This section details the command line interface.

Using Host Interface Commands

The following host interface commands are used in administering and maintaining the host interface environment and gathering network diagnostic information. Both Avaya and Interface Systems, Inc. have developed commands to support the host interface software on the CONVERSANT system.

Session Numbering Conventions

Many of the commands described in this section require you to specify the session on which the command is to be performed. CONVERSANT system commands require host session numbers. TN3270 and SNA 3270 commands require HLLAPI session IDs. The host session numbers range from 0 to 127. The HLLAPI session IDs range from 2 to 129 and are equal to the host session number plus 2. The host session numbers are assigned dynamically when the user configures the LUs and stops and restarts the voice system. The mapping from SNA_SERVER or connection name and LU number to host session number is provided on the Display Host Sessions screen as described in Chapter 3, “Voice System Administration,” of *CONVERSANT System Version 8.0 Administration*, 585-313-510. If only one connection is configured and consecutive LUs are configured starting at 2, the HLLAPI session IDs are equal to the LU numbers.

For example, if, on SNA_SERVER, tnsna1 is configured with LUs 2–33 and tnsna 2 is configured with LUs 2–33, host session 0 equals HLLAPI session 2 which also equals LU2 on connection tnsna 1. Also, host session 32 equals HLLAPI session 34 which also equals LU2 on tnsna 2.

CONVERSANT System Commands

For more information about these commands, see Appendix A, "Summary of Commands," in *CONVERSANT System Version 8.0 Administration*, 585-313-510.

- **sb_te [*session _range*** or ***session_number*]**

  This terminal emulation program allows a user to step through the host application, including the log-on, log-off, and recovery procedures of a Script Builder application. This session number or range is optional and can be from 0 to 127. If a session range is used, it can only include 10 sessions. If no session number is given, the command opens all available sessions that are installed in the system and automatically displays the first session (use **CTRL V** to display multiple sessions). If a session is not specified, the system assumes the value "all." The following are examples of valid **sb_te** commands:

  sb_te

  sb_te 5

  sb_te 5-14

Note: Use the Display Host Sessions screen in the CONVERSANT system menu to provide the mapping of connection name and LU number to the session number.

Use the **sb_te** command to verify if there have been any changes to the host application. Sometimes changes can occur on the host end that are not passed down to the CONVERSANT system development end. These discrepancies result in error messages being logged on the CONVERSANT system and the session stays in recovery. The session number chosen must be released from the host interface process before you invoke **sb_te**. To do this for non-Script Builder applications, stop the DIP. To do this for Script Builder applications, use the **hfree** command.

Use the following procedure to start terminal emulation:

a Establish the host connection by using the **start_hi** command.
b Start the 3270 Terminal Emulation software directly by entering:

sb_te [*session_number* or ***session_range*]**

The Terminal Emulator (TE) displays the current screen of the LU. The 3270 status line appears at the bottom of the screen to inform you whether the host is active. See Appendix B, "Status Line Information," of the *TN3270 or SNA 3270 User's Guide* for information about the indicators shown in the 3270 status line and what those values mean.

Note: The status line of the screen displays the HLLAPI session ID. This value equals the host session number plus 2.

You can now send commands to the host.

sb_te executes the HLLAPI TE. (See the *TN3270 or SNA 3270 User's Guide* for more information.) The CONVERSANT System V8.0 host software provides a new look and feel to the TE. Some important keystrokes to remember are:

CONTROL V – Goes to the next session

CONTROL U – Displays the 3270 command menu

CONTROL X – Exits the terminal emulator

CONTROL Z – Escape to the UNIX prompt

ESC R – Resets the keyboard

ESC B – Captures a screen

- **hspy [*session_number*** or ***range*** or ***all*]**

  By specifying a session number (or all), the **hspy** command shows what screen currently is being presented on that session. Make a note of this information. It will help you to isolate what screens might be involved in the problem.

- **hlogin [*host application*** or ***session_number*** or ***range*** or ***all*]**

  The **hlogin** command invokes the log-in procedure that is defined in the application's host session maintenance section. This command is often used in the system's cron table to log in early the next morning. It is a clean, convenient way to log in to the host application. Note that the session must be in the logged-out state before you can use the **hlogin** command.

- **hlogout [*host application*** or ***session_number*** or ***range*** or ***all*]**

  The **hlogout** command invokes the log-out procedure that is defined in the application's host session maintenance section. This command is often used in the system's cron table to log off the host before it goes down at night. It is a clean, convenient way to log off the host application. Note that the session must be in the logged-in state before you can use the **hlogout** command.

- **hfree [*host_application*** or ***session_number*** or ***range*** or ***all*]**

  The **hfree** command releases sessions from their Script Builder application assignments. You must use this command to switch from the application to the TE on a given session. Note that the **hfree** command does not automatically log out the specified session.

- **hassign *[hostsvc] host_application* to [*session_number*** or ***range*** or ***all [FTSCRT]*]**

  The **hassign** command assigns applications to session numbers. It is necessary to use this command to switch from using the terminal emulator to having an application assigned to a given session. Note that the **hassign** command automatically attempts to log in the specified session. Use the optional FTSCRT argument to assign a session for file transfer.

- **hdelete [hostsvc] *host_application* from [*session_number*** or ***range*** or ***all*]**

  The **hdelete** command invokes the log-out procedure that is defined in the application's host session maintenance section, releases LUs from their Script Builder application assignments, and automatically removes the host application from the session.

- **hnewscript *host_application***

  The **hnewscript** command updates the system memory with the latest copy of the specified host application. This command is required to place an updated version of the host application into effect.

- **hdisplay *[host_application]***

  The **hdisplay** command displays the host applications that have been verified and installed on the system, as well as the current session assignments for each host application.

- **hstatus [*host_application*** or ***session_number*** or ***range*** or ***all*]**

  The **hstatus** command displays the current status of the host application assigned to the associated session numbers. The command is useful when isolating problems with host applications and checking the number of sessions involved on a call.

- **start_hi**

  The **start_hi** command starts the host interface software.

CAUTION:
Normally, the **start_hi** command is not run by the user. This command should not be run when the voice system is at run level 4, as this command is run automatically when the system initializes.

- **stop_hi**

  The **stop_hi** command stops the host interface software.

CAUTION:
Normally, the ***stop_hi*** command is not run by the user. This command should not be run when the voice system is at run level 4.

Interface Systems Commands

For more information about the following commands, see the Interface Systems documentation that accompanied your software. Specific references to the Interface Systems documentation for each command are provided.

- **comsend** — Use this command to send a file to the host. When using this command, you must be logged in as root and identify the HLLAPI session ID on which the transfer will be performed. See the chapter on “Transferring Files,” of the *TN3270 or SNA 3270 User’s Guide* for more information. The HLLAPI session ID is equal to the host session number plus 2. You must specify this session ID as a hexadecimal value. For example, host session 10 uses HLLAPI ID 0xC.

Note: The host configuration for file transfer must use the write structured fields (WSF), also known as DFT mode, in the logmode table entry. You must verify this entry in both upgrades and new systems because this is a change from previous product releases.

- **comreceive** — Use this command to receive a file from the host. When using this command, you must be logged in as root and identify the HLLAPI session id on which the transfer will be performed. See the chapter on “Transferring Files,” of the *TN3270 or SNA 3270 User’s Guide* for more information. The HLLAPI session id is equal to the host session number plus 2. You must specify this session ID in as a hexadecimal value. For example, host session 10 uses HLLAPI id 0xC.

Note: The host configuration for file transfer must use the write structured fields (WSF), also known as DFT mode, in the logmode table entry. You must verify this entry in both upgrades and new systems because this is a change from previous product releases.

Administering File Transfer

You can perform file transfer either interactively through the screen interface or directly via UNIX commands. To perform file transfer interactively via the CONVERSANT system screens, use the File Transfer option that is provided by way of the Terminal Emulator selection in the Command Menu. For information on performing file transfer using either method, see the chapter on “Transferring Files,” of the *TN3270 or SNA 3270 User’s Guide*.

Interactive File Transfer

See the chapter on “Controlling 3270 Emulation,” of the *TN3270 or SNA 3270 User’s Guide* for information on:

- Accessing the main screen and navigating through its menus
- Controlling display sessions
- Controlling printer sessions
- Viewing host response times
- Sending NetView alert messages
- Exiting and resuming 3270 emulation

Direct File Transfer

To perform file transfers directly, use the **comsend** and **comreceive** programs in the directory **/usr/bin**. These programs transfer files using a screen buffer that interacts with the host IND$FILE file transfer program.

Note: Log on to the host session and access the system-ready prompt before executing the **comsend** and **comreceive** commands. Be certain that the logmode is set appropriately for your host connection. Use the **sb_te** command to establish the host session before using the file transfer program.

comsend

Use the **comsend** program to upload a file, that is, to transfer a file from the CONVERSANT system to the host mainframe. Following is an example of the **comsend** program:

comsend -h 0x*N unix_file host_filename options*

- *-h* is an argument indicating the HLLAPI ID number used to send files. *N* is a value for this argument. The value for *N* ranges from 2 through 129 (0x81). You must specify these values as hexadecimal.
- *unix_file* is the name of the CONVERSANT system file to be transferred. Note that the naming convention of the file follows UNIX standards. The file must be named with a full path. No directory is required if the file is in the current working directory. See Table 18 on page 124 for suggestions on how to specify filenames when performing file transfers.
- *host_filename* is the name of the target host mainframe file.
- You can enter several *options* to control the file transfer. These options are described in the chapter on “Transferring Files,” of the *TN3270 or SNA 3270 User’s Guide*. Note that some options are not available with all systems.

Note: Mainframes vary in their requirements for the options list. Some require that the option list be enclosed in parentheses, some require only the left parentheses, and others do not permit the use of parentheses. You should therefore verify the requirements of the mainframe you are using before using any of these options. All meta characters, for example, asterisk (*) comma (,), parentheses (), and so on, must be preceded by a backslash (\) character in the **comsend** command line. Other characters might work, but the backslash is recommended in all cases.

comreceive

Use the **comreceive** program to download, that is, to transfer, a file from the host mainframe to the CONVERSANT system. Following is an example of the **comreceive** program:

comreceive -h 0x*N unix_file host_filename options*

- *-h* is an argument indicating the HLLAPI ID number used to receive files. *N* is a value for this argument. The value for *N* ranges from 2 through 129. You must specify this value as hexadecimal.

- *unix_file* is the name of the target CONVERSANT system file on download. Note that the naming convention of the file follows UNIX system standards. The file must be named with a full path. No directory is required if the file is in the current working directory. See Table 18 on page 124 for suggestions on how to specify filenames when performing file transfers.
- *host_filename* is the name of the host mainframe file to be transferred.
- You can enter several *options* to control the file transfer. File transfer options are provided in the chapter on "Transferring Files," of the *TN3270 or SNA 3270 User's Guide*. Note that some options are not available with all systems.

Note: Mainframes vary in their requirements for the options list. Some require that the option list be enclosed in parentheses, some require only the left parentheses, and others do not permit the use of parentheses. You should therefore verify the requirements of the mainframe you are using before using any of these options. All meta characters, for example, asterisk (*) comma (,), parentheses (), and so on, must be preceded by a backslash (\) character in the **comreceive** command line. Other characters might work, but the backslash is recommended in all cases.

When an ASCII file is received from the host, it may have been sent with a ^Z (**CONTROL Z**) at the end of the file. When you try to "vi" the file, a message may complain about an unrecognized character. You should attempt to remove the character from the file. This is typically a problem with TSO and VM systems.

When a binary file is received from the host, zeros are added to the end of the block to make it a multiple of 80. For example, if a file of 4 bytes is sent to the host, it may contain 76 more bytes when it is returned (4 + 76 = 80).

Table 18. Filename Guidelines for File Transfer

If Filename Contains	UNIX			Host 3270		
	Syntax	Examples		Syntax	Examples	
		Original	Converted		Original	Converted
| & ; < > () ' \ ' * ? [] # ~ [1]	Precede each special character with a backslash (\)	ix'yy'a\bc	x\'yy'\a\\bc	Precede each special character with a backslash (\)	#AB~C*DE?cde#f*h	\#AB~C*DE\?cde#f*h
dollar sign ($)	Precede dollar sign ($) with a backslash (\)	AB$tmp	AB\$tmp	Precede dollar sign ($) with a backslash (\) [2]	XXyy$zz	XXyy\$zz
at sign (@)	Precede at sign(@) with a backslash (\)	AB@tmp	AB\@tmp	Precede at sign (@) with a backslash (\)[3]	XXyy@zz	XXyy\@zz

1 of 2

Table 18. Filename Guidelines for File Transfer

If Filename Contains	UNIX			Host 3270		
	Syntax	Examples		Syntax	Examples	
		Original	Converted		Original	Converted
period (.)	No special syntax	s.xx.c	s.xx.c	Enclose filename first with a backslash (\) followed by an apostrophe (‘)[4]	s.xx.c	\’.xx.c\’
Any character not shown above[5]	No special syntax	abcd	abcd	No special syntax	a123bcd	a123bcd
						2 of 2

[1] Precede # and ~ with a backslash only if they begin the filename.

[2] Precede $ with a backslash only when the file transfer is done directly with the comsend or comreceive commands. Do not precede $ when the file transfer is done through the 3270 terminal emulator.

[3] Precede @ with a backslash only when the file transfer is done directly with the **hsend**, **comsend,** or **comreceive** commands. Do not precede @ when the file transfer is done through the 3270 terminal emulator.

[4] Enclose . with a backslash and apostrophe only if transferring files to or from a tso system and the dots in the filename are a fully qualified filename (containing the user id).

[5] You may not use an underscore when specifying a filename.

Host DIP Parameter File

The host DIP parameter file **/vs/etc/default/agdip3270** allows access to certain parameters that may be useful when designing your host application.

SESSIONS_TO_START Parameter

The **SESSIONS_TO_START** parameter allows you to specify the number of sessions to which you want to receive and send screens concurrently. Setting this parameter to 5, for example, means that five sessions at most are allowed to start logging in, logging out, or recovering at one time. The rest of the sessions wait to start until one or more of the five sessions complete executing their log-in, log-out, or recover sequences. The default is to allow all 32 sessions to access the host concurrently.

In most cases, the default works well. However, if all 32 sessions are logging in, an individual session takes longer to log in than it would if it was the only one accessing the host. This is because an individual session has to compete for the host link resource with 31 other sessions.

On the other hand, setting **SESSIONS_TO_START**=1 allows only one session to log in at a time while the rest wait their turn. This speeds up the logging in for one session, but overall it takes longer to log in all sessions than if multiple session were logging in at a time.

Selecting a suitable value for **SESSIONS_TO_START** depends on the host environment and the applications and involves some trial and error. However, in most cases the default of 32 is acceptable.

To set the **SESSIONS_TO_START** parameter:

1. Stop the voice system. See “Common System Procedures” in *CONVERSANT System Version 8.0 System Reference*, 585-313-215, for the procedure.
2. Enter **vi /vs/etc/default/agdip3270**
3. Set the **SESSIONS_TO_START** parameter to the maximum number of sessions you want to be receiving and sending screens concurrently. For example, to have only one session interacting with the host, set **SESSIONS_TO_START**=1.
4. Exit the file.
5. Start the voice system. See “Common System Procedures” in *CONVERSANT System Version 8.0 System Reference*, 585-313-215, for the procedure.

LOGOFF_TIMEOUT Parameter

The **LOGOFF_TIMEOUT** parameter specifies the maximum amount of time the **stop_vs** command waits for any active session to be logged out before the voice system is stopped. The default value for **LOGOFF_TIMEOUT** is 60 seconds. You should increase this value only if **stop_vs** does not allow enough time for all LUs to be logged off. This may be necessary if your system has many LUs or the LUs have lengthy logout sequences.

MAX_NUMBER_OF_LUs Parameter

The **MAX_NUMBER_OF_LUs** parameter specifies the maximum number of LUs that can be configured for a system. The default value is 128 LUs.

AUTORESET_LUs Parameter

The **AUTORESET_LUs** parameter specifies that the hostdip automatically sends a reset key if the LU is in recovery and input is inhibited. It also sends the system reset key if the LU is in recovery and the screen is the system services control point (SSCP) or UNOWNED. The default value for **AUTORESET_LUs** is No. This parameter should only be set to Yes if the LUs get stuck in recovery for one of the reasons listed previously in this description.

RETCOUNT_TODO_POWEROFF Parameter

The **RETCOUNT_TODO_POWEROFF** parameter allows you to specify the retry count interval during the recovery process to simulate a terminal Power Off/Power On sequence. Most of the time the Power Off/Power ON sequence will result in a Host BANNER screen being sent from the host, so that the “login” process can be tried. For example setting, **RETCOUNT_TODO_POWEROFF=2**, will result in a Power Off/Power ON sequence being generated at every other recovery retry attempt. Using a Power Off/Power On sequence is useful for trying to get an individual LU to recover without having to stop and start the host connection and thus causing all the LUs to have to recover.

STAGGER_ BETWEEN_ RETRIES Parameter

The **STAGGER_BETWEEN_RETRIES** parameter specifies the number of seconds to wait between RECOVERING LUs to be RESTARTED. It is rare that a wait time is used; however, some hosts do require it.

SYSREQ_IS_ POWEROFF Parameter

The **SYSREQ_IS_POWEROFF** parameter is set to false by default. If set to true, then when a Voice@Work or Script Builder script attempts to do a SYSREQ keystroke, a Power Off/Power On sequence will be done instead.

Retry Strategy

Sessions that repeatedly fail to log in are subject to a retry delay before trying to recover again. The retry delay is incremented by 20 seconds for each consecutive failure. For example, six consecutive failed attempts results in 120 seconds of delay before the session is allowed to start its seventh attempt to log in. The session will wait no longer than 600 seconds to attempt to log in again.

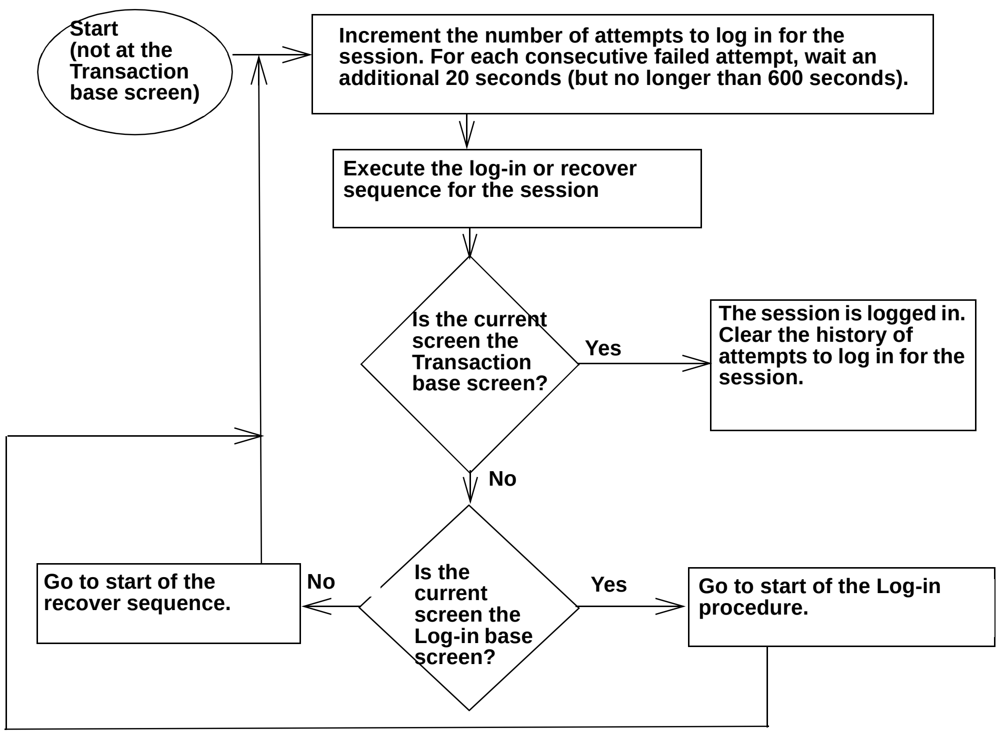
A session is *not* delayed the next time it tries to log in if one of the following occurs:

- The session is freed via **hfree**. This clears all past failed attempts made to log the session in.
- The **hlogout**, **hassign**, **hnewscript**, or **hdelete** commands are executed on the session. These commands are queued if the session is in the middle of executing its log-in or recover sequence. Once the log-in or recover sequences completes, the commands are executed.
- The session recovers and becomes logged in.

Figure 23 shows how a session tries to log in. After a session is assigned a Script Builder application, it begins to log in. After it completes the log-in sequence, the session is in one of the following states:

- The session is **logged in** if the current screen is the transaction base screen. In this state, the session is ready to get data when a call is made to a Script Builder application.
- The session is **logging in** if the current screen is the log-in base screen. In this state, the session waits an additional 20 seconds before attempting to reach the transaction base screen.
- The session is **recovering**. In this state, the session waits an additional 20 seconds before attempting to reach the transaction base screen.

Figure 23. How a Session Tries to Log in



Application Development Issues

The following are current application development issues concerning the host interface software.

Intermediate Screens

It has always been important for host applications to deal with intermediate screens. An intermediate screen occurs when the host responds with a screen and unlocks the keyboard (the sign that the voice system can send another screen to the host), but in fact the host is sending another screen. This behavior occurs most frequently during the log-in process.

Because some networks are faster than other networks, it is possible that a host application will experience more intermediate screens over a faster network. If an application is moved from a slower to a faster network environment, and the log-in sequence does not work as it used to, it is likely that the application is receiving these intermediate screens. If you experience this problem:

- Add recognition criteria to the screen definition to differentiate the intermediate screen from the final screen.
- Add an additional Get Host Screen action between the Send Host Screen Action and the real Get Host Screen action. In the new Get Host Screen action, wait for a screen that will not be sent. This forces a pause in the sequence. Then, the next Get Host Screen executes after the host has had a chance to send all screens.

TCP/IP Communications

Transmission Control Protocol/Internet Protocol (TCP/IP) is a process-to-process protocol. The IP component dispatches information around the network, and the TCP component assures that information's accuracy. TCP/IP within the CONVERSANT system provides high-speed data transmission over an Ethernet network.

The UCS 1000 provides dual, integrated LAN connections on the CPU complex. The MAP/40P supports 1 or 2 PCI LAN circuit cards.

There are three areas that you must address when using TCP/IP protocol with the CONVERSANT system.

- Current network topology — See Network Architecture on page 129.
- Application structure — See Application Development Issues on page 129.
- Software installation — See "Installing Base System Software," in the maintenance book for your platform.

See http://www.sco.com/documentation for more information on TCP/IP protocol.

Network Architecture

UnixWare 7 includes an implementation of the TCP/IP protocol. The package has been internetworked successfully by Avaya and others with a wide variety of TCP/IP networks. Given this standard and compliant implementation, there is no reason that a CONVERSANT system running this software cannot be connected successfully to a standard, compliant TCP/IP network.

Figure 24 shows the layering of TCP/IP over Ethernet and token ring in the context of the first four layers of the OSI Reference Model. This figure shows that the styles of networking differ at the physical and link layer only (Ethernet versus token ring). The network layer and above are the same, regardless of the physical and link layer.

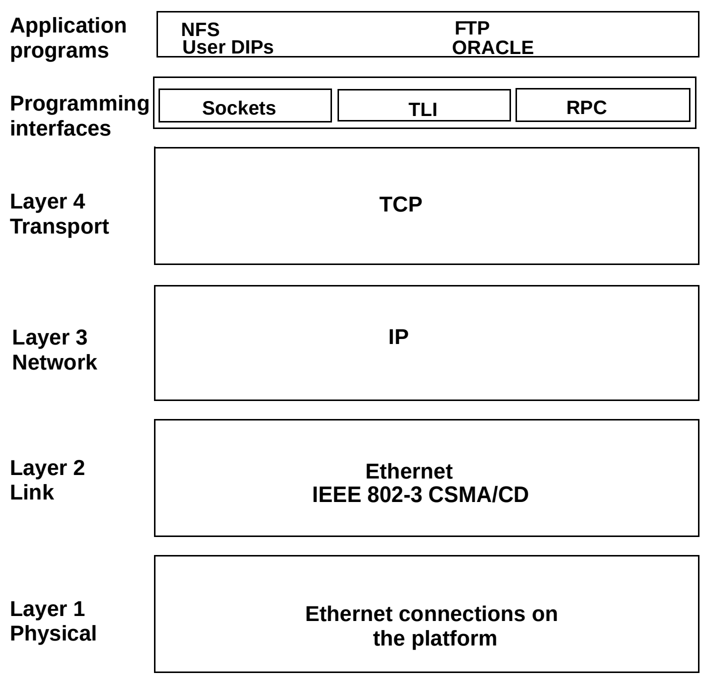
Some standard networking utilities are available with UnixWare. These utilities are used to network the CONVERSANT system with other machines without developing a custom application interface. These utilities include:

- **rcp** — Allows a user to copy files to and from a remote machine.
- **rlogin** — Allows a user to log in to a remote machine from a local machine.
- **ftp** — Transfers files to and from a remote network.
- **telnet** — Enables terminal and terminal-oriented processes to communicate on a TCP/IP network.

See the UnixWare network administration book for more information about standard networking utilities.

Sockets, TLI, and RPC are alternative and equivalent application programming interfaces to the network. Sockets was introduced as part of the UNIX systems 4.2BSD. Almost every implementation of TCP/IP for UNIX includes a sockets interface. TLI was released with AT&T UNIX R3. It offers a streams-based interface to the transport layer. As a streams interface, it offers a measure of portability from one protocol suite to another. RPC is a remote procedure call interface. This implementation of TCP/IP offers a Sockets, a TLI, and an RPC interface.

Figure 24. Network Layering



Application Development Issues

Typically, a CONVERSANT system is added to a network that is already in place. Adding a CONVERSANT system to your network allows you to use information from the network in a custom application. You must first determine if the information you want is available through the standard UnixWare utilities (for example, **rcp**, **rlogin**, **ftp**) or whether a custom process is necessary. See the UnixWare network administration book for more information about the standard network utilities.

If it is necessary to write a custom program, you may also write a data interface process (DIP) to access the program. See *CONVERSANT System Version 8.0 Application Development with Advanced Methods,* 585-313-216, for more information on DIPs. When writing the DIP, you must use the Sockets, TLI, or RPC application programming interface (see *NFS/RPC/NIS Administration*). Within Script Builder you must create an external action to call the **dbase** script instruction to execute the DIP. See Chapter 11, "Using Advanced Features," of *CONVERSANT System Version 8.0 Application Development with Script Builder*, 585-313-217, for more information.

It is also possible to use sockets, TLI, or RPC with a Response Application Programming Interface (IRAPI) application. Care must be used to determine who the process should block. See Chapter 5, "IRAPI," of *CONVERSANT System Version 8.0 Application Development with Advanced Methods*, 585-313-216, for more information.

Provisioning TCP/IP

The following sections detail the network addressing and hardware and software requirements for the TCP/IP protocol.

Network Addressing

TCP/IP allows each machine on the network to be "addressed" so that it can be distinguished from other machines. Every host on the network must have a unique network address. The addresses consist of four decimal integers, each of which is separated by a dot (.). Three different classes of addresses are possible with the TCP/IP protocol. The default network uses a class A address. However, if you want to assume responsibility for maintaining the network database files, other network architectures are possible.

See TCP/IP administration at http://www.sco.com/documentation for more information on setting up the network.

Hardware Requirements

UCS 1000

The UCS 1000 supports dual, integrated LAN connections:

- One integrated on the the SBC card (rear I/O)
- Second on front I/O

Both LAN connections support either 10 Base T or 100 Base T.

MAP/40P

To support the TCP/IP protocol, the MAP/40P requires either an Ethernet or token/ring circuit card, depending on the physical and link layer.

The SMC Ethernet circuit card supports the following physical interfaces to the network:

- External transreceiver
- 10BASE2 (ThinNet)
- 10BASET (Twisted Pair)
- 100BASE-TX with category 5 cable

The IBM token ring circuit card supports the following physical interfaces to the network:

- IBM token ring network PC adapter cable
- Category 3, 4, or 5 cable
- See the "Installing and Replacing Circuit Cards" in the maintenance book for your platform for information on installing the circuit cards and configuring TCP/IP.

Software Requirements

UnixWare 7.1.1 must be installed on the CONVERSANT system to use TCP/IP protocol. The driver (Ethernet or Token Ring for MAP/40P) is included on this tape and should be configured properly.

SQL*NET Communications

SQL*NET or NET8 is the ORACLE communications component that allows the CONVERSANT system to share information that is stored in different remote ORACLE databases. With SQL*NET or NET8, you can run an ORACLE tool or another application on the CONVERSANT system and be able to find, manipulate, and store data in an ORACLE database that is located on another machine.

For more information on ORACLE SQL*NET or NET8 communications, see the ORACLE SQL*NET or NET8 TCP/IP documentation on the *ORACLE8i Online Generic Documentation*.

Asynchronous Communication

Asynchronous communication is a method of data transmission that allows characters to be sent at irregular intervals by preceding each character with a start bit and following it with a stop bit.

The CONVERSANT system platform supports two standard asynchronous connections and one standard parallel printer connection on the system by way of an EIA-232 serial port. One of the standard asynchronous connections is reserved for the remote maintenance circuit card (COM1 in the UCS 1000 and COM2 in the MAP/40P). This circuit card provides a standard modular connection for access to the built-in modem. This arrangement allows access to the CONVERSANT system through a remote terminal. This makes it possible to monitor system output and alarms, manipulate system resources, and perform software-related tasks without being physically near the CONVERSANT system platform.

Data transmission is limited to 9600 bps (maximum) for asynchronous communication established with any device.

See “Ports” and “Printers” in Chapter 7, “Peripheral Administration,” of *CONVERSANT System Version 8.0 Administration*, 585-313-510, for information on setting up the ports and printers.

Standard Asynchronous Connections

On the UCS 1000, COM1 is located on the front and rear I/0 and COM2 is located on the rear I/O. COM1 is reserved for the remote maintenance circuit card.

On the MAP/40P, the standard asynchronous ports are located on the back of platform. COM2 is reserved for the remote maintenance circuit card.

See Chapter 2, “Unpacking and Installing the System,” in *CONVERSANT System Version 8.0 New System Installation*, 585-313-149, for more information.

Note that the distance between transmission devices (for example, the CONVERSANT system and a terminal) should not exceed 15 meters (50 feet) according to the EIA-232 standard recommendation. Devices can be separated by longer distances, however, depending on how much electrical interference exists in the area. Use an asynchronous data unit (ADU) for distances from 15 to 1525 meters (50 to 5000 feet). See the appropriate ADU documentation for maximum limits.

8-Port Asynchronous Circuit Card Connections

The CONVERSANT system supports connections to one or more asynchronous host computers or additional modems via an 8-port asynchronous interface. These eight additional serial ports are provided by an 8-port asynchronous circuit card. See Chapter 3, “Making Cable Connections,” of *CONVERSANT System Version 8.0 New System Installation*, 585-313-149, for more information.

A Transmission Level Adjustment

Overview

A Transmission Level Plan (TLP) for a piece of telecommunications equipment is a set of specifications dictating the incoming and outgoing speech volume levels that pass through the equipment and the hardware and software tools for implementing those specifications. The specifications take into account the level plans of the various telephone network interfaces to which the equipment will connect. The goal of the plan is to ensure that all speech heard by a caller be at a level that is appropriate for listening without causing oscillations or distortions in the network.

Note: The UCS 1000 does not support analog connections.

Transmission Level Plan

Network-Interface Hardware

The CONVERSANT system connects to two types of telephone network facilities, analog (tip/ring) and digital (T1/E1).

Note: The UCS1000 does not support an analog interface.

The CONVERSANT system's default TLP is partially based on the following facts concerning the system's network interface hardware:

- The system's T1/E1 interface circuit cards have a gain of 0 dB built into the hardware interface.
- The system's tip/ring interface circuit cards have a nominal gain of -0 dB built into the hardware interface (when a perfect impedance match exists between the interface and the line to which it is connected).

Typical Network TLP Characteristics

The tip/ring and T1/E1 network facilities have typical TLP characteristics associated with them. The system default TLP is partially based on the following typical network TLP characteristics:

- The system default TLP assumes a nominal gain of 0 dB in each digital trunk that is connected to any T1/E1 card in the system.
- The system default TLP assumes a nominal gain of -3 dB in each analog loop that is connected to any tip/ring card in the system.

Incoming and Outgoing Speech Volume Nonbridging Modes

When a voice signal enters a CONVERSANT system in a nonbridged connection, it is usually going to be coded and stored in the speech filesystem of the machine. Before it is coded, its incoming volume can be adjusted by the IVOL parameter.

By default, all coding modes are subjected to an automatic gain control (AGC) after the IVOL is applied. The AGC is used to maintain a proper recording level. AGC attenuates signals that would otherwise be too loud and amplifies signals that would otherwise be too quiet. For this reason, small adjustments of IVOL have little impact when AGC is active. It may, however, be necessary to increase IVOL if the input is so low that the AGC takes it to be silence. (Such input the AGC treats as background noise and, for the listener’s comfort, does not pass it. Consequently, input that is too low may be cut off and short phrases may be completely missing.)

When a voice signal that is stored in the speech file system is played back from a CONVERSANT system to a caller, its outgoing volume can be adjusted by the OVOL parameter.

The CONVERSANT Digital Interfaces screen allows the user to adjust both the incoming and outgoing speech volume for analog (tip/ring) and digital (T1/E1) network interfaces. The analog IVOL and OVOL parameters apply to all analog circuit cards in the system. The digital IVOL and OVOL parameters apply to T1/E1 circuit cards on a per-card basis.

IVOL and OVOL should be thought of as volume multipliers (that is, +/- gain) of the incoming or outgoing signal. A value of 1000 for IVOL or OVOL is equivalent to multiplying the incoming or outgoing signal volume by 1, that is, *unity gain*. Each multiplication of the current IVOL or OVOL setting by a factor of 0.707 results in a signal volume gain of -3 dB from the current volume (s volume of 3 dB lower); each multiplication of the current IVOL or OVOL setting by a factor of 1.414 results in a signal volume gain +3 dB from the current volume (s volume of 3 dB higher).

Note: IVOL and OVOL affect only signals being coded or played back by the CONVERSANT system. They do not affect end-to-end conversations in call bridge mode, DTMF or CPT tone detection, or speech recognition.

Table 19 shows the IVOL and OVOL settings required to implement the default TLP along with the actual gain in decibels (shown in parenthesis) that each setting represents.

Table 19. Default System IVOL and OVOL Settings

Network Facilities	IVOL	OVOL	Text-to-Speech (TTS) OVOL[1]
Analog	4000(+12)	1000(0)	4000
Digital	1414(+3)	707(-3)	1000[2]

[1] The TTS OVOL is an option only when the TTS package is installed.
[2] The TTS OVOL default value may be too low in some cases. You might want to use a higher value. However, if a value is too high, it can cause distortion of the outgoing text.

Voice Coding and Play

As described above, most switches build in some loss in a typical station-set-to-station-set connection. With the system in a nonbridging mode, station-set-to-station-set connection actually involves a signal being affected by IVOL while it is coded and stored on the disk, and then affected by OVOL when it is played back. To be in accordance with the TLP, the level the caller hears during playback should be somewhat lower than the level that was spoken when the signal was coded. See Reasons for Deviating from the Default IVOL and OVOL Settings below for considerations used to determine proper input and output volume.

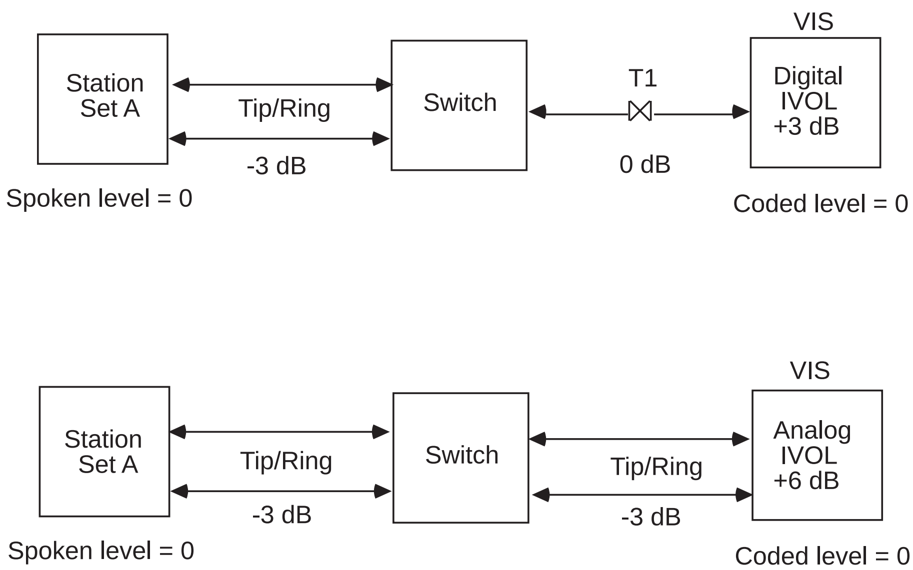
Voice Coding

Figure 25 shows an example of how the IVOL parameters control the level at which a voice signal is coded and stored in the system speech filesystem. The levels in Figure 25 show the interaction between a switch and the CONVERSANT system.

Note: The actual default IVOL is +12 dB rather than the +6 shown in Figure 25. The +12 dB level reduces the chance of low input volume levels being recorded as silence. Automatic gain control (AGC) makes it unlikely that the higher input volume will cause clipping or other distortion. See Reasons for Deviating from the Default IVOL and OVOL Settings on page 138.

The top part of Figure 25 shows a T1/E1 interface connected to the system. The bottom part shows a tip/ring interface connected to the system. As you follow the signal from left to right, if the initial spoken level is 0 and all typical network TLP characteristics listed above are true, the coded level that is stored in the speech filesystem will always be zero (0), regardless of which type of network interface is connected to the CONVERSANT system.

Figure 25. Effect of IVOL Parameters on Voice Coding

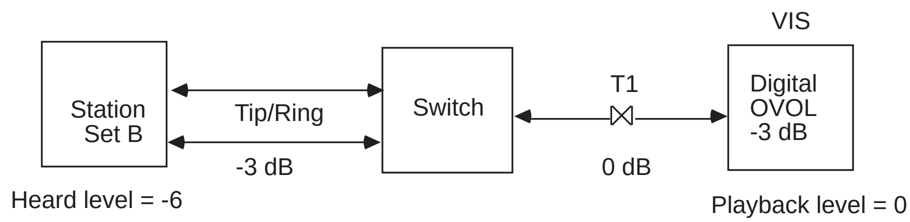


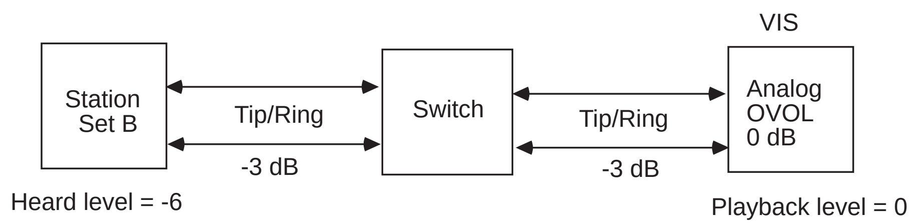
Voice Play

Figure 26 shows how the default OVOL parameters control the level at which a previously coded voice signal that is stored in the speech filesystem is played back.

The top part of Figure 26 shows a T1 interface connected to the system. The bottom part shows a tip/ring interface connected to the system. As you follow the signal from right to left, if the signal was coded in the manner depicted in Figure 25 on page 137, the initial playback level is 0. If all typical network TLP characteristics listed above are true, the level heard at the station set is always -6, regardless of which type of network interface is connected to the system. Since the initial spoken level shown in Figure 25 on page 137 was 0, the heard level of -6 is in accordance with the CONVERSANT TLP.

Figure 26. Effect of OVOL Parameters on Voice Play





Reasons for Deviating from the Default IVOL and OVOL Settings

For most applications, the default TLP provides callers with appropriate speech volume levels for prompts that were coded as shown in Figure 25 on page 137.

In many cases, however, speech prompts are coded in a studio at higher volumes than they would have been coded from a system network interface. In these situations, it may be desirable to decrease the applicable OVOL parameter (analog or digital, depending on whether playback is from tip/ring or T1) to decrease the volume the caller actually hears. Note that if the system is used to code speech that will be played back with the prerecorded speech, you should increase IVOL by the same amount that you decrease OVOL to ensure that speech is coded at the same level.

Also, some network lines and trunks do not abide by the typical network characteristics listed above. For example, some T1 trunks actually have insertion loss in the network. This loss can be compensated for by increasing the corresponding IVOL and OVOL parameters by an amount that is equal to the additional insertion loss. For example, if the digital trunks connected to a system had insertion loss of -3 dB instead of 0 associated with them as the default CONVERSANT system TLP assumes, the default digital IVOL and OVOL parameters could be changed to 2000 and 1000, respectively. This would have the effect of adding a gain of +3 dB to the incoming signal before coding, and adding a gain of +3 dB to the outgoing signal before playback (see Table 19 on page 136 and the accompanying explanation). Making these changes results in meeting the TLP goal of -6 dB gain from end to end.

If the IVOL is set too low, phrases may be cut short or may be missing. In such cases the input may be so low that the AGC takes it to be silence. (Such input the AGC treats as background noise and, for the listener’s comfort, does not pass it. Consequently, input that is too low may be cut short and some phrases may be completely missing.) Try turning up the IVOL to solve the problem.

If IVOL is set too high, the recorded phrases may be recorded louder than prerecorded speech or speech that is heard while connected to a bridge to another person. The AGC generally prevents this from being a problem, but if recorded speech appears to be to loud, try using a lower IVOL setting.

Finally, subjectivity plays a large role in the effectiveness of a TLP. What sounds appropriate to one person may sound inappropriate to another. The default IVOL and OVOL parameters have been carefully selected to provide appropriate volume levels in the majority of applications. It is strongly recommended that you do not change them based on subjective evaluation. However, the flexibility is provided to tune them to whatever suits the needs of the application at hand.

Transmission Level Plan and Call Bridging

When two incoming calls are bridged together by the system, the callers on either end (station set A and station set B) can talk with each other through the system. In such a situation, the previously discussed IVOL and OVOL parameters do not apply. Instead, software on the CONVERSANT system (specifically the TSM process) has built-in rules for directing the CONVERSANT system network interface cards to insert up to +6 dB gain in either direction of a call bridge connection.

As mentioned earlier, the CONVERSANT system TLP dictates that there be a gain of -6 dB from station-set-to-station-set. Assuming the typical network TLP characteristics for the network facilities (as discussed in Typical Network TLP Characteristics on page 135), Figure 27 through Figure 30 on page 140 show the amount of gain (in decibels) that is automatically inserted in each direction for each of the four possible call bridging scenarios.

- Figure 27 on page 140 shows analog-to-analog (tip/ring-to-tip/ring) call bridging.
- Figure 28 on page 140 shows digital-to-digital (T1-to-T1) call bridging.
- Figure 29 on page 140 shows analog-to-digital (tip/ring-to-T1) call bridging.
- Figure 30 on page 140 shows digital-to-analog (T1-to-tip/ring) call bridging.

Figure 27. Analog-to-Analog Call Bridging

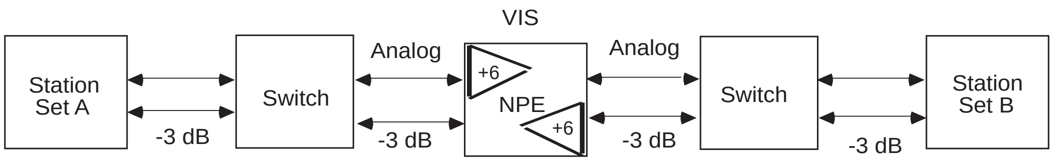


Figure 28. Digital-to-Digital Call Bridging

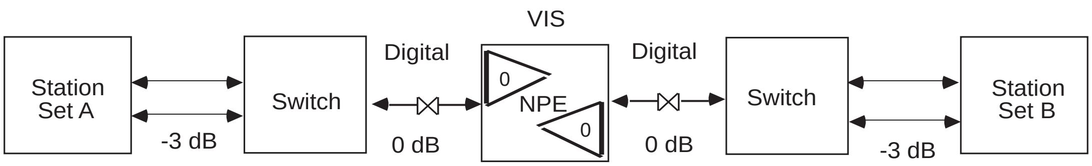


Figure 29. Analog-to-Digital Call Bridging

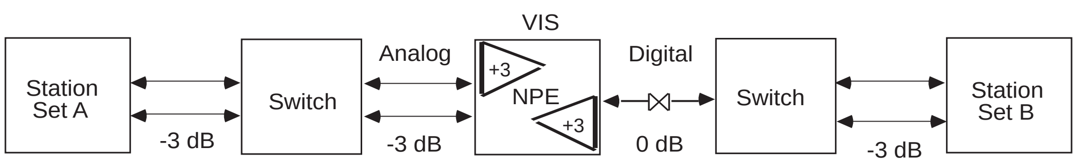
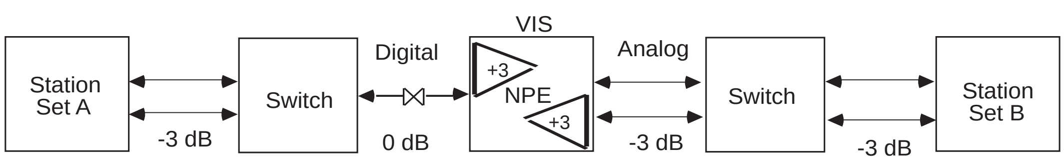


Figure 30. Digital-to-Analog Call Bridging



Possible Exceptions to the CONVERSANT System TLP

When a CONVERSANT system is used as a network adjunct within the network, some changes to the default TLP settings are recommended to ensure optimal speech volume and clarity. Similar conditions may apply to commercial customers providing voice-response services that are primarily accessed via the long distance network.

Note: Customers should check with their switch and/or network services provider before deviating from the CONVERSANT system TLP.

In addition to the 6-dB end-to-end loss described above, the FCC requires that the local exchange carrier (LEC) insert a 6-dB loss as signals leave the long distance network. AT&T TrueVoice feature adds up to a gain of 4 dB as low volume level signals leave the network. This partially compensates for the loss of the 6 dB that the LEC is required to insert.

Within the AT&T network, network recordings and announcements (and operator speech) should be presented at a volume level of -21 dBm0 at the AT&T Point of Presence. If recordings and announcements are recorded at a volume level that is too high, the calling party is likely to hear distortion. This distortion is due to the clipping that occurs when high volume levels exceed the capability of the network to represent the signal. Clipping can occur at -13 dBm0. Excessive volume levels on prerecorded speech is one of the most frequent causes of hearing distortion.

Within the AT&T network, all trunks and bridges should insert zero gain so that the volume level remains as -21 dBm0 throughout the AT&T network.

When a CONVERSANT system is being used as a network adjunct and digital trunks are used, it is recommended that IVOL and OVOL settings be set to the nondefault value of 1000 (for zero gain) and that prerecorded speech be recorded at -21 dBm0. By using zero gain, the CONVERSANT system being used as the network adjunct may avoid introducing another digital signal transformation that contributes to the distortion heard by users of the network.

When the quality of speech is more important than minimizing space usage (as for most prerecorded announcements and prompts), encode the speech using 64 Kbps PCM rather than 32 Kbps ADPCM.

When the highest quality speech is required, ISDN PRI may provide slightly better sound quality than T1 E&M robbed-bit signaling (see Chapter 2, Digital Telephony Interfaces), where the least significant bits rather than voice data are used for signaling. However, the difference in sound quality is not the only advantage to using ISDN PRI.

The AT&T Truevoice processing inserts a gain of up to 14 dB at low frequencies (around 180 Hz). This is designed to compensate for the normal losses in the analog loop and in telephone handsets. This helps to make low- frequency voices sound richer and more like the person is nearby. A 25-Hz (inaudible) tone is used to prevent doubling of the AT&T Truevoice effect in bridges and speech that is recorded via the long distance network. This tone is lost in an analog bridge or recording that is made over an analog interface. Fortunately, most of the inserted gain is also lost, so there is not a full doubling effect of AT&T Truevoice. When a digital bridge is used or a digital interface is used to make a recording, the 25-Hz tone is preserved along the enhanced signal and the AT&T Truevoice effect is not applied twice. Unfortunately, it may take about a second for the 25-Hz tone to be recognized and for the redundant Truevoice processing to be disabled.

To prevent problems with excessive volume levels from enhanced AT&T Truevoice processing, it is recommended that recordings and announcements be recorded in a studio lab (rather than via the network) and that the low frequencies not be enhanced by the studio.

Tip/Ring Switch Integration Issues

Switch integration for tip/ring circuit cards is done using the Analog Interfaces screen. This screen is described in Chapter 5, “Switch Interface Administration,” of *CONVERSANT System Version 8.0 Administration*, 585-313-508. The tip/ring interface is administered on a system-wide basis; that is, the tip/ring parameters apply to all tip/ring circuit cards. To administer the tip/ring interface, you can specify several parameters or accept the default values.

Note: The IVOL, OVOL, and TDM output gains are system-wide parameters for analog interfaces and can be changed on a per-card basis for digital interfaces. These parameters can be modified via the Switch Interfaces screens as described in Chapter 5, “Switch Interface Administration,” of *CONVERSANT System Version 8.0 Administration*, 585-313-508. Gains can also be overridden on a per-channel basis by an IRAPI application. However, even with IRAPI, the IVOL cannot be overridden for speech that is recorded on a tip/ring channel. See *CONVERSANT System Version 8.0 Application Development with Advanced Methods*, 585-313-216, for the IRP_PLAYGAIN and IRP_RECORD_GAIN parameters under IrPARAMETERS(4IRAPI).

All tip/ring lines originating from a Merlin Legend switch connected to the system must be set up in a Merlin Legend calling group as type “Generic VMI.”

Calculating Volume Settings

This section offers a method for calculating the volume settings just described. The same method applies for calculation of IVOL and for OVOL. The method applies to speech and signal processor (SSP) circuit cards.

Calculation of volume settings for the CONVERSANT system is very similar to calculation of relative voltage levels. So that volume settings take the form of integers, however, the equation is calculated relative to the (arbitrary) constant of 1000 rather than to a second voltage:

$$\text{dB} = 20\log\frac{\text{Vol}}{1000}$$

To calculate a setting where the volume level to be set is known and is expressed in decibels, the required setting becomes relative to the inverted log (or antilog) of 10:

$$\text{Vol} = 1000\times 10^{(\text{dB} \div 20)}$$

Using this formula, the formula for the setting required to get an OVOL level of -3dB would look like:

$$\text{OVOL} = 1000\times 10^{(-3 \div 20)}$$

which becomes: $\text{OVOL} = 1000\times 10^{-0.15}$ or: $\text{OVOL} = (1000\times 0.707) = 707$.

The setting would be 707.

Table 20 sets out the results of this calculation in 3-dB increments from -21 dB to 21 dB.

Table 20. Loss and Gain Settings

dB Loss	Setting	dB Gain	Setting
0	1000	0	1000
-3	707	3	1412
-6	501	6	1995
-9	354	9	2818
-12	251	12	3981
-15	177	15	5623
-18	125	18	7943
-21	89	21	11220

Glossary

Numerics

23B+D

23 bearer (communication) and 1 data (signaling) channel on a T1 PRI circuit card.

30B+D

30 bearer (communication) and 1 data (signaling) channel (plus framing channel 0) on an E1 PRI circuit card.

3270 interface

A link between one or more CONVERSANT machines and a host mainframe. In CONVERSANT system documentation, the 3270 interface specifically means the link between one or more system machines and an IBM host mainframe.

47B+D

47 bearer (communication) and 1 data (signaling) channel on two T1 PRI circuit cards.

4ESS®

A large Lucent central office switch used to route calls through the telephone network.

A

AC

alternating current

ACD

automatic call distributor

AD

application dispatch

AD-API

application dispatch application programming interface

adaptive differential pulse code modulation

A means of encoding analog voice signals into digital signals by adaptively predicting future encoded voice signals. This adaptive modulation method reduces the number of bits required to encode voice. See also “pulse code modulation.”

adjunct products

Products (for example, the Adjunct/Switch Application Interface) that the system administers via cut-through access to the inherent management capabilities of the product itself. This is in opposition to the ability of the system to administer the switch directly.

Adjunct/Switch Application Interface

An optional feature package that provides an Integrated Services Digital Network-based interface between Avaya PBXs and adjunct processors.

ADPCM

adaptive differential pulse code modulation

ADU

asynchronous data unit

advanced speech recognition

A speech recognition ability that allows the system to understand WholeWord, FlexWord, and Natural Language Speech Recognition inputs from callers.

affiliate

A business organization that Avaya controls or with which Avaya is in partnership.

AGL

application generation language

ALERT

System alerter process

alerter

A system process that responds to patterns of events logged by the “logdaemon” process.

American Standard Code for Information Interchange

A standard code for data representation that represents alphanumeric characters as binary numbers. The code includes 128 uppercase and lowercase letters, numerals, and special characters. Each alphanumeric and special character has an ASCII code (binary) equivalent that is 1 byte long.

analog

An analog signal, such as voice or music, that varies in a continuous manner. An analog signal may be contrasted with a digital signal, which represents only discrete states.

ANI

automatic number identification

announcement

A message the system plays to the caller to provide information. The caller is not asked to give a response. Compare to prompt.

API

Application programming interface

application

The automated transaction (interactions) among the caller, the voice response system, and any databases or host computers required for your business. See also application script.

application administration

The component of the system that provides access to the available applications and helps you manage and administer them.

application installation

A two-step process in which the CONVERSANT system invokes the TSM script assembler for the specific application name and moves files to the appropriate directories.

application script

The computer program that controls the application (the transaction between the caller and the system). The CONVERSANT system provides several methods for creating application scripts, including Voice@Work, Script Builder, Transaction Assembler Script (TAS) language, and the Intuity Response Application Programming Interface (IRAPI).

application simulation

A process in which the system simulates the behavior of an application as it is expected to behave on the CONVERSANT system. It is useful as a debugging tool.

application verification

A process in which the system verifies that all the components needed by an application are complete.

ASCII

American Standard Code for Information Interchange

ASI

analog switch integration

ASR

advanced speech recognition

asynchronous communication

A method of data transmission in which bits or characters are sent at irregular intervals and spaced by start and stop bits rather than by time. Compare to synchronous communication.

asynchronous data unit

An electronic communications device that allows computer systems to communicate over asynchronous lines more than 50 feet (15 meters) in length.

asynchronous event

An event detected by the system that disrupts the normal flow of an application that is running. At present, the CONVERSANT system recognizes only one type of asynchronous event—a hang up.

automatic call distributor

That part of a telephone system that recognizes and answers incoming calls and completes these calls based on a set of instructions contained in a database. The ACD can send the call to an operator or group of operators as soon as the operator has completed a previous call or after the system has played a message to the caller.

automatic number identification

A method of identifying the calling party by automatically receiving a string of digits that identifies the calling station of a particular customer.

B

back up

The preservation of the information in a file in a different location so that the data is not lost in the event of hardware or system failure.

backing up an application

Using a utility that makes an archive copy of a completed application or an interim copy of an application in progress. The backup copy can be restored to the system if the online version is damaged, or if you make revisions and want to go back to the previous version.

barge-in

A capability provided by WholeWord speech recognition, Dial Pulse Recognition (DPR), and Natural Language Speech Recognition (NLSR) that allows callers to speak or enter their responses during the prompt and have those responses recognized (similar to the Speak with Interrupt capability). See also echo cancellation.

batch file

A file containing one or more lines, each of which is a command executable by the UNIX shell.

BB

bulletin board

binary synchronous communications

A character-oriented synchronous link protocol.

blind transfer protocol

A protocol in which a call is completed as soon as the extension is dialed, without having to wait to see if the telephone is busy or if the caller answered.

bps

bits per second

BRDG

call bridging process

bridging

The process of connecting one telephone network connection to another over the system TDM bus. Bridging decreases the processing load on the system since an active bridge does not require speech processing, database access, host activity, and so on, for the transaction.

BSC

binary synchronous communications

bundle

In the context of the Enhanced File Transfer package, this term is used to denote a single file, a group of files (package), or a combination of both.

byte

A unit of storage in the computer. On many systems, a byte is 8 bits (binary digits), which is the equivalent of one character of text.

C

call classification analysis

A process that enables application designers to use information available within the system to classify the disposition of originated and transferred calls. Intelligent CCA is provided with the system. Full CCA is an optional feature package.

call data event

A parameter that specifies a list of variables that are appended to a call data record at the end of each call.

call data handler process

A software process that accumulates generic call statistics and application events.

called party number

The number dialed by the person making a telephone call. Telephone switching equipment can use this number to selectively route an incoming call to a particular department or agent.

caller

The party who calls for a service, gets connected to the system, and interacts with it. Because the system can also make outbound calls for service, the caller can also be the person who responds to those outbound calls.

call flow

See transaction.

call progress tones

Standard telephony sounds that indicate the status of the call. These sounds include busy, fast busy, ringback, reorder, etc.

card cage

An area within a hardware platform that contains and secures all of the standard and optional circuit cards used in the system.

cartridge tape drive

A high-capacity data storage and retrieval device that can be used to transfer large amounts of information onto high-density magnetic cartridge tape based on a predetermined format. This tape can be removed from the system and stored as a backup or used on another system.

CAS

channel associated signalling

caution

An admonishment or advisory statement used in the system documentation to alert the user to the possibility of a service interruption or a loss of data.

CCA

call classification analysis

CDH

call data handler process

CELP

code excited linear prediction

central office

A location in which large telecommunication devices such as telephone switches and network access facilities are maintained. These locations follow strict installation and operation requirements.

central processing unit

See processor.

CGEN

Voice system general message class

channel

See port.

channel associated signaling

A type of signaling that can be used on E1 circuit cards. It occurs on channel 16.

CICS

Customer Information Control System

circuit card upgrade

A new circuit card that replaces an existing card in the platform. Usually the replacement is an updated version of the original circuit card to replace technology made obsolete by industry trends or a new system release.

cluster controller

A bisynchronous interface that provides a means of handling remote communication processing.

CMS

Call Management System

CO

central office

code excited linear prediction

A means of encoding analog voice signals into digital signals that provides excellent quality with use of minimum disk space.

command

An instruction or request the user issues to the system software to make the system perform a particular function. An entire command consists of the command name and options.

configuration

The arrangement of the software and hardware of a computer system or network. The system configuration includes either a standard or custom processor, peripheral equipment (for example, printers and modems), and software applications. Configuration also refers to the way in which the switch network is set up; that is, the types of products that are in the network and how those products communicate.

configuration management

The component of the system that allows you to manage the current configuration of voice channels, host sessions, and database connections, assign scripts to run on specific voice channels or host sessions, assign functionality to SSP and E1/T1 circuit cards, and perform various maintenance functions.

connect and disconnect (C and D) tones

DTMF tones that inform the system when the attendant has been connected (C) and when the caller has been disconnected (D).

connected digits

A sequence of digits that the system can process as a group, rather than requiring the caller to enter the digits one at a time.

Converse Data Return (conv_data)

A Voice@Work external function or a Script Builder external action that supports the DEFINITY® call vectoring (routing) feature by enabling the switch to retain control of vector processing in the system environment. It supports the DEFINITY “converse” vector command to establish a two-way routing mechanism between the switch and the system to facilitate data passing and return.

controller circuit card

A circuit card used on a computer system that controls its basic functionality and makes the system operational. These circuit cards are used to control magnetic peripherals, video monitors, and basic system communications.

copying an application

A utility in which information from a source application is directed into the destination application.

coresidency

The ability of two products or services to operate and interact with each other on a single hardware platform.

CPE

customer-provided equipment or customer premise equipment

CPN

called party number

CPT

call progress tones

CPU

central processing unit

crash

An interactive utility for examining the operating system core and for determining if system parameters are being exceeded.

CSU

channel service unit

custom grammar

See custom vocabulary.

custom speech

Unique words or phrases to be used in system voice prompts that Avaya records on a per-customer basis.

custom vocabulary

A specialized package of unique words or phrases created on a per-customer basis and used by WholeWord or FlexWord speech recognition.

Customer Information Control System

Part of the operating system that manages resources for running applications (for example, IND$FILE). Note that TSO and CMS provide analogous functionality in other host environments.

CVS

converse vector step

D

danger

An admonishment or advisory statement used in the system documentation to alert the user to the possibility of personal injury or death.

data interface process

A software process that communicates with interactive voice response (IVR) applications.

database

A structured set of files, records, or tables.

database field

A field used to extract values from a local database and form the structure upon which a database is built.

database record

The information in a database for a person, product, event, and so on. The database record is made up of individual fields for each information item.

database table

A structure, made up of columns and rows, that holds information in a database. Database tables provide a means of storing information that changes too often to “hard-code,” or store permanently, in the transaction outline.

dB

decibel

DB

database

DBC

database checking process

DBMS

database management system

DC

direct current

DCE

data communications equipment

DCP

digital communications protocol

debug

The process of locating and correcting errors in computer programs; also referred to as troubleshooting.

default

The way a computer performs a task in the absence of other instructions.

default owner

The owner of a channel when no process takes ownership of that channel. The default owner holds all idle, in-service channels. In terms of the IRAPI, this is typically the Application Dispatch process.

diagnose

The process of performing diagnostics on a bus or on circuit cards.

dial ahead

The ability to collect and process touchtone inputs in sequence, even when they are received before the prompts.

dial pulse recognition

A method of recognizing caller pulse inputs from a rotary telephone.

dialed number identification service

A service that allows incoming calls to contain information about the telephone number for which it is destined.

dial through

A capability provided by touchtone and dial pulse recognition that allows callers to enter their responses during the prompt and have those responses recognized (similar to the Speak with Interrupt capability). See also barge-in and echo cancellation.

DIMM

dual in-line memory module

DIO

disk input and output process

DIP

data interface process

directory

A type of file used to group and organize other files or directories.

display errdata

A command that displays system errors sent to the logger.

DMA

direct memory address

DNIS

dialed number identification service

DPR

dial pulse recognition

DSP

digital signal processor

DTE

data terminal equipment

DTMF

dual tone multi-frequency

DTR

data terminal ready

dual 3270 links

A feature that provides an additional physical unit (PU) for a cost-effective means of connecting to two host computers. The customer can connect a system to two separate FEPs or to a single FEP shared by one or more host computers. Each link supports a maximum of 32 LUs.

dual tone multi-frequency

A touchtone sound that is an audio signal including two different frequencies. *DTMF feedback* is the process of the switch providing this information to the system. *DTMF muting* is the process of ignoring these tones (which might be simulated by human speech) when they are not needed for the application.

dump space

An area of the disk that is fixed in size and should equal the amount of RAM on the system. The operating system “dumps” an image of core memory when the system shuts down automatically. The dump can be fetched after rebooting to help in analyzing the cause of the shutdown.

E

E&M

Ear and Mouth

E1 / T1

Digital telephony interfaces, commonly called *trunks*. E1 is an international standard at 2.048 Mbps. T1 is a North American standard at 1.544 Mbps.

Ear and Mouth

A common T1 trunking protocol for connection between two switches.

EBCDIC

Extended Binary Coded Decimal Interexchange Code

echo cancellation

The process of making the channel quiet enough so that the system can hear and recognize WholeWord, dial pulse, and Natural Language inputs during the prompt. See also barge-in.

ECS

Enterprise Communications Server

editor system

A system that allows speech phrases to be displayed and edited by a user.

EFT

Enhanced File Transfer

EIA

Electronic Industries Association

EISA

Extended Industry Standard Architecture

EMI

electromagnetic interference

emulator

Software on one operating system that imitates or reproduces the behavior of input and output on a different operating system.

engine

The software used to perform speech recognition or text-to-speech functions. Usually used with reference to proxy software and systems. See also Proxy Text-to-Speech (PTTS) and Natural Language Speech Recognition (NLSR).

enhanced basic speech

Prerecorded speech available from Avaya in several languages. Sometimes called standard speech.

Enhanced File Transfer

A feature that allows the transferring of files automatically between the CONVERSANT system and a synchronous host processor on a designated logical unit.

Enhanced Serial Data Interface

A software-controlled and hardware-controlled method used to store data on magnetic peripherals.

Enterprise Communications Server

The telephony equipment that connects your business to the telephone network. Sometimes called a switch.

error message

A message on the screen indicating that something is wrong with the system, often with a suggestion of how to correct it.

ESD

electrostatic discharge

ESDI

Enhanced Serial Data Interface

ESS

electronic switching system

EST

Enhanced Software Technologies, Inc.

ET

error tracker

Ethernet

A name for a local area network that follows IEEE Standard 802.3. Supported implementations are 10Baset and 100Baset.

event

The notification given to an application when some condition occurs that is generally not encountered in normal operation.

EXTA

external alarms feature message class

external actions

Specific predefined (or customer-created) system tasks that Script Builder can call or *invoke* to interact with other products or services. When an external action is invoked, the systems displays a form that provides choices in each field for the application developer to select. Examples are Call_Bridge, Make_Call, SP_Allocate, SR_Prompt, and so on. In Voice@Work, external actions are called external functions.

external functions

Specific predefined (or customer-created) system tasks that Voice@Work can call or *invoke* to interact with other products or services. The function allows the application developer to enter the arguments for the function to act on. Examples are concat, getarg, length, substring, and so on. In Script Builder, external functions are called external actions.

F

FAX Actions

An optional feature package that allows the system to send fax messages.

FCC

Federal Communications Commission

FDD

floppy disk drive

feature

A function or capability of a product or an application within the system.

feature package

An optional package that may contain both hardware and software resources to provide additional functionality to a standard system.

feature_tst script package

A standard system software program that allows a user to perform self-tests of critical hardware and software functionality.

FEP

front end processor

field

See database field.

FIFO

first-in-first-out processing order

file

A collection of data treated as a basic unit of storage.

file transfer

An option that allows you to transfer files interactively or directly to and from UNIX using the file transfer system (FTS).

filename

Alphabetic characters used to identify a particular file.

FlexWord™ speech recognition

A type of speech recognition based on subword technology that recognizes phonemes or parts of words in a specific language. See also subword technology.

foos

facility out-of-service state

FTS

file transfer process message class

Full CCA

A feature package that augments the types of call dispositions that Intelligent CCA can provide.

function key

A key, labeled F1 through F8, on your keyboard to which the system software gives special properties for manipulating the user interface.

G

GEN

PRISM logger and alerter general message class

grammar

The inputs that a recognizer can match (identify) from a caller.

GUI

graphical user interface

H

hard disk drive

A high-capacity data storage and retrieval device that is located inside a computer platform. A hard disk drive stores data on nonremovable high-density magnetic media based on a predetermined format for retrieval by the system at a later date.

hardware

The physical components of a computer system. The central processing unit, disks, tape and diskette drives, and so on, are all hardware.

hardware upgrade

Replacement of one or more fundamental platform hardware components (for example, the CPU or hard disk drive), while the existing platform and other existing optional circuit cards remain.

HDD

hard disk drive

High Level Language Applications Programming Interface

An application programming interface that allows a user to write custom applications that can communicate with a host computer via an API.

HLLAPI

High Level Language Applications Programming Interface

HOST

host interface process message class

host computer

A computer linked to a network to provide a range of services, such as database access and computation. The host computer operates in a time-sharing manner with other computers linked to it via the network.

hwoos

hardware out-of-service state

Hz

Hertz

I

IBM

International Business Machines

iCk or ICK

The system integrity checking process.

ID

identification

IDE

integrated disk electronics

idle channel

A channel that either has no owner or is owned by its default owner and is onhook.

IE

information element

IEEE

Institute of Electrical and Electronic Engineers

IND$FILE

The standard SNA file transfer utility that runs as an application under CICS, TSO, and CMS. IND$FILE is independent of link-level protocols such as BISYNC and SDLC.

independent software vendor

A company that has an agreement with Avaya to develop software to work with the system to provide additional features required by customers.

indexed table

A table that, unlike a nonindexed table, can be searched via a field name that has been indexed.

industry standard architecture

A PC bus standard that allows processors and other circuit cards to communicate with each other.

INIT

voice system initialization message class

initialize

To start up the system for the first time.

inserv

in-service state

Integrated Services Digital Network

A network that provides end-to-end digital connectivity to support a wide range of voice and data services.

intelligent CCA

Monitoring the line after dialing is complete to determine whether a busy, reorder (fast busy), or other failure has been encountered. Intelligent CCA also recognizes when the extension is answered or if the extension is not answered after a specified number of rings. The monitoring capabilities are dependent on the network interface circuit card and protocol used

interface

The access point of a system. The interface is designed to provide you with easy access to the software capabilities of the system.

interrupt

The termination of voice and/or telephony functions when some condition occurs.

Intuity Response Application Programming Interface

A library of commands that provide a standard development interface for voice-telephony applications.

IOB

I/O companion card to the SBC. This is part of the CPU Complex.

IPC

interprocess communication

IRAPI

Intuity Response Application Programming Interface

IRQ

interrupt request

ISA

industry standard architecture

ISDN

Integrated Services Digital Network

ISV

independent software vendor

ITAC

International Technical Assistance Center

K

Kbps

kilobytes per second

KB

kilobyte

keyboard mapping

In emulation mode, this feature enables the keyboard to send 3270 keyboard codes to the host according to a configuration table set up during installation.

keyword spotting

A capability provided by WholeWord speech recognition, FlexWord speech recognition, and Natural Language speech recognition that allows the system to recognize a single word in the middle of an entire phrase spoken by a caller in response to a prompt.

L

LAN

local area network

LDB

local database

LED

light-emitting diode

library states

The state information about channel activities maintained by the IRAPI.

LIFO

last-in-first-out processing order

line side E1

A digital method of interfacing a system to a PBX or switch using E1-related hardware and software.

line side T1

A digital method of interfacing a system to a PBX or switch using T1-related hardware and software.

listfile

An ASCII catalog that lists the contents of one or more talkfiles. Each application script is typically associated with a separate listfile. The listfile maps speech phrase strings used by application scripts into speech phrase numbers.

local area network

A data communications network in a limited geographical area. The LAN provides communications between computers and peripherals.

local database

A database residing on the system.

LOG

System logger process message class

logical unit

A type of SNA Network Addressable Unit.

logdaemon

A UNIX system information and error logging process.

logger

See logdaemon.

logging on/off

Entering or exiting the system software.

LSE1

line side E1

LST1

line side T1

LU

logical unit

M

magnetic peripherals

Data storage devices that use magnetic media to store information. Such devices include hard disk drives, diskette drives, and cartridge tape drives.

main screen

The system screen from which you are able to enter either the System Administration or Voice System Administration menu.

maintenance process

A software process that runs temporary diagnostics and maintains the state of circuit cards and channels.

manoos

manually out-of-service state

masked event

An event that an application can ignore (that is, the application can request not to be informed of the event).

master

A circuit card that provides clock information to the TDM bus.

Mbps

megabits per second

MB

megabyte

megabyte

A unit of memory equal to 1,048,576 bytes (1024 x 1024). It is often rounded to one million.

menu

Options presented to a user on a computer screen or with voice prompts.

MF

multifrequency

MHz

megahertz

mirroring

A method of data backup that allows all of the data transactions to the primary hard disk drive to be copied and maintained on a second identical drive in near real time. If the primary disk drive fails or becomes disabled, all of the data stored on it (up to 1.2 billion bytes of information) is accessible on the second mirrored disk drive.

ms

millisecond

msec

millisecond

MS-DOS

A personal computer disk operating system developed by the Microsoft Corporation.

MTC

maintenance process

multifrequency

Dual tone digit signaling (similar to DTMF), used for trunk addressing between network switches or by network operators.

multithreaded application

A single process or application that controls several channels. Each thread of the application is managed explicitly. Typically this means state information for each thread is maintained and the state of the application on each channel is tracked.

N

Natural Language Speech Recognition (NLSR)

An advanced type of speech recognition. Like WholeWord and Flexword speech recognition, NLSR can recognize particular words and phrases, but it can also interpret and assign meaning to those words and phrases. NLSR can also recognize natural numbers and currency amounts. Because of the greater vocabulary and grammar requirements associated with NLSR, it works best with an external speech recognition or "proxy" server.

NCP

Network Control Program

NEBS

Network Equipment Building Standards

NEMA

National Electrical Manufacturers Association

netoos

network out-of-service state

NetView

An optional feature package that transmits high-priority (major or critical) messages to the host as operator-generated alerts (OGAs) over the 3270 host link. The NetView Alarm feature package does not require a dedicated LU.

NFAS

non-facility associated signaling

NFS

network file sharing

NM-API

Network Management - Application Programming Interface

NMVT

network management vector transport

nonex

nonexistent state

nonindexed table

A table that can be searched only in a sequential manner and not via a field name.

nonmasked event

An event that must be sent to the application. Generally, an event is nonmaskable if the application is likely to encounter state transition errors by trying to ignore it.

NRZ

non return to zero

NRZI

non return to zero inverted

null value

An entry containing no value. A field containing a null value is normally displayed as blank and is different from a field containing a value of zero.

O

OEM

original equipment manufacturer

OGA

operator-generated alert

online help

Messages or information that appear on the user’s screen when a function key (F1 through F8) is pressed or a "Help" menu item or icon is clicked.

operator-generated alert

A system-monitoring message that is transmitted from the CONVERSANT system or other computer system to an IBM host computer and is classified as critical or major.

option

An argument used in a command line to modify program output by modifying the execution of a command. When you do not specify any options, the command executes according to its default options.

ORACLE

A company that produces relational database management software. It is also used as a generic term that identifies a database residing on a local or remote system that is created and maintained using an ORACLE RDBMS product.

P

P&C

Prompt and Collect Voice@Work node or Script Builder action step

PBX

private branch exchange

PC

personal computer

PCB

printed circuit board

PCI

peripheral component interconnect

PCI Mezzanine Card

A PCI module, usch as a LAN or RAID controller, that connects to the CPU Complex IOB companion card.

PCM

pulse code modulation

PEC

price element code

peripheral (device)

Equipment such as printers or terminals that is in addition to the basic processor.

peripheral component interconnect

A newer, higher speed PC bus that is gradually displacing ISA for many components.

permanent process

A process that starts and initializes itself before it is needed by a caller.

phoneme

A single basic sound of a particular spoken language. For example, the English language contains 40 phonemes that represent all basic sounds used with the language. The English word “one” can be represented with three phonemes, “w” - “uh” - “n.” Phonemes vary between languages because of guttural and nasal inflections and syllable constructs.

phrase

A set of one or more words used within an application. Examples include "Thank you for calling XZY Business," "One," and "At the tone, press—."

phrase filtering (screening)

The rejection of unrecognized speech. The WholeWord, FlexWord, and Natural Language speech recognition packages can be programmed to reprompt the caller if the system does not recognize a spoken response.

phrase number

An identification number associated with a particular phrase in a speech pool.

phrase tag

A string of up to 50 characters that identifies the contents of a speech phrase used by an application script.

platform migration

See platform upgrade.

platform upgrade

The process of replacing the existing platform with a new platform.

pluggable

A term usually used with speech technologies, in particular standard speech, to indicate that a basic algorithmic technique has been implemented to accept one or more sets of parameters that tailors the algorithm to perform in one or more languages.

PMC

PCI Mezzanine Card

poll

A message sent from a central controller to an individual station on a multipoint network inviting that station to send if it has any traffic.

polling

A network arrangement whereby a central computer asks each remote location whether it wants to send information. This arrangement enables each user or remote data terminal to transmit and receive information on shared facilities.

port

A connection or link between two devices that allows information to travel to a desired location. See telephone network connection.

PRI

Primary Rate Interface

Primary Rate Interface

An ISDN term for connections over E1 or T1 facilities that are usually treated as trunks.

private branch exchange

A private switching system, either manual or automatic, usually serving an organization, such as a business or government agency, and usually located on the customer's premises.

processor

In system documentation, the computer on which UnixWare and the system software runs. In general, the part of the computer system that processes the data. Also known as the central processing unit.

prompt

A message played to a caller that gives the caller a choice of selections in a menu and asks for a response. Compare to announcement.

prompt and collect (P and C)

A message played to a caller that gives the caller a choice of selections in a menu and asks for a response. The response is collected and the script progresses based on the caller's response.

proxy server

A server external to the CONVERSANT system used in a client/server configuration to perform processor-intensive functions, such as Natural Language Speech Recognition or text-to-speech beyond the capabilities of the CONVERSANT system. See also Natural Language Speech Recognition (NLSR) and Proxy Text-to-Speech (PTTS).

Proxy Text-to-Speech (PTTS)

The capability to do text-to-speech processing using one or more auxiliary computers that are connected to the CONVERSANT in a client/server configuration. PTTS is an alternative to the standard Text-toSpeech feature for use in applications where the demand is very high or where a language is needed that is not supported on the SSP circuit card. See also Text-to-Speech.

pseudo driver

A driver that does not control any hardware.

PSTN

public switch telephone network

pulse code modulation

A digital modulation method of encoding voice signals into digital signals. See also adaptive differential pulse code modulation.

R

RAID

redundant array of independent disks

RAID array

An assembly of disk drives configured to provide some level of RAID functionality.

RAM

random access memory

RDMBS

ORACLE relational database management system

RECOG

speech recognition feature message class

recognition type

The type of input the recognizer can understand. Available types include touchtone, dial pulse, and Advanced Speech Recognition (ASR), which includes WholeWord, FlexWord, and Natural Language speech recognition.

recognizer

The part of the system that compares caller input to a grammar to correctly match (identify) the caller input.

record

See database record.

recovery

The process of using copies of the system software to reconstruct files that have been lost or damaged. See also restore.

remote database

Information stored on a system other than your current system that can be accessed by the CONVERSANT system.

remote maintenance circuit card

A CONVERSANT system circuit card, available with a built-in modem, that allows remote personnel (for example, field support) to access all CONVERSANT system machines. This card is standard equipment on all new purchases.

REN

ringer equivalence number

reports administration

The component of the system that provides access to system reports, including call classification, call data detail, call data summary, message log, and traffic reports.

restore

The process of recovering lost or damaged files by retrieving them from available backup tapes or from another disk device. See also recovery.

restore application

A utility that replaces a damaged application or restores an older version of an application.

reuse

The concept of using a component from a source system in a target system after a software upgrade or platform migration.

RFS

remote file sharing

RM

resource manager

RMB

remote maintenance circuit card

roll back

To cancel changes to a database since the point at which changes were last committed.

rollback segment

A portion of the database that records actions that should be undone under certain circumstances. Rollback segments are used to provide transaction rollback, read consistency, and recovery.

RTS

request to send

S

SBC

(1) sub-band coding; (2) a single-board computing circuit card that is part of the CPU Complex

SCA

single connector architecture

screen pop

A method of delivering a screen of information to a telephone operator at the same time a telephone call is delivered. This is accomplished by a complex chain of tasks that include identifying the calling party number, using that information to access a local or remote ORACLE database, and pulling a “form” full of information from the database using an ORACLE database utility package.

script

The set of instructions for the CONVERSANT system to follow during a transaction.

Script Builder

An optional software package that provides a menu-oriented interface designed to assist in the development of custom voice response applications on the CONVERSANT system (see also Voice@Work).

SCSI

small computer system interface

SDLC

synchronous data link control

SDN

software defined network

shared database table

A database table that is used in more than one application.

shared speech

Speech that is a part of more than one application.

shared speech pools

A parameter that allows the user of a voice application to share speech components with other applications.

SID

station identification

signal processor circuit card

A speech processing circuit card that is an older, lower-capacity version of the speech and signal processor (SSP) circuit card.

single-threaded application

An application that runs on a single voice channel.

slave

A circuit card that depends on the TDM bus for clock information.

SLIP

serial line interface protocol

small computer system interface

A disk drive control technology in which a single SCSI adapter circuit card plugged into a PC slot is capable of controlling as many as seven different hard disks, optical disks, tape drives, and so on.

SNA

systems network architecture

SNMP

simple network management protocol

software

The set or sets of programs that instruct the computer hardware to perform a task or series of tasks, for example, UnixWare software and the system software.

software upgrade

The installation of a new version of software in which the existing platform and circuit cards are retained.

source system

The system from which you are upgrading (that is, your system as it exists *before* you upgrade).

speech and signal processor circuit card

A high-performance signal processing circuit card capable of simultaneous support for various speech technologies.

speech energy

The amount of energy in an audio signal. Literally translated, it is the output level of the sound in every phonetic utterance.

speech envelope

The linear representation of voltage on a line. It reflects the sound wave amplitude at different intervals of time. This envelope can be plotted on a graph to represent the oscillation of an audio signal between the positive and negative extremes.

speech file

A file containing an encoded speech phrase.

speech filesystem

A collection of several talkfiles. The filesystem is organized into 16-KB blocks for efficient management and retrieval of talkfiles.

speech modeling

The process of creating WholeWord speech recognition algorithms by collecting thousands of different speech samples of a single word and comparing them all to obtain a statistical average of the word. This average is then used by a WholeWord speech recognition program to recognize a single spoken word.

speech space

An area that contains all digitized speech used for playback in the applications loaded on the system.

speech phrase

A continuous speech segment encoded into a digital string.

speech recognition

The ability of the system to understand input from callers.

speech recognition engine

See engine.

SPIP

signal processor interface process

SPPLIB

speech processing library

SQL

structured query language

SR

speech recognition

SSP

speech and signal processor circuit card

standard speech

The speech package available in several languages containing simple words and phrases produced by Avaya for use with the system. This package includes digits, numbers, days of the week, and months, each spoken with initial, medial, and falling inflection. The speech is in digitized files stored on the hard disk to be used in voice prompts and messages to the caller. This feature is also called enhanced basic speech.

standard vocabulary

A standard package of simple word speech models provided by Avaya and used for WholeWord speech recognition. These phrases include the digits “zero” through “nine,” “yes,” “no,” and “oh,” or the equivalent words in a specific language.

string

A contiguous sequence of characters treated as a unit. Strings are normally bounded by white spaces, tabs, or a character designated as a separator. A string value is a specified group of characters symbolized by a variable.

structured query language

A standard data programming language used with data storage and data query applications.

subword technology

A method of speech recognition used in FlexWord recognition that recognizes phonemes or parts of words. Compare to WholeWord speech recognition.

switch

A software and hardware device that controls and directs voice and data traffic. A customer-based switch is known as a private branch exchange.

switch hook

The device at the top of most telephones that is depressed when the handset is resting in the cradle (in other words, is *on hook*). The device is raised when the handset is picked up (in other words, when the telephone is *off hook*).

switch hook flash

A signaling technique in which the signal is originated by momentarily depressing the switch hook.

switch interface administration

The component of the system that enables you to define the interaction between the system and switches by allowing you to establish and modify switch interface parameters and protocol options.

switch network

Two or more interconnected telephone switching systems.

synchronous communication

A method of data transmission in which bits or characters are sent at regular time intervals, rather than being spaced by start and stop bits. Compare to asynchronous communication.

SYS

UNIX system calls message class

sysgen

system generation

System 75

An advanced digital switch supporting up to 800 lines that provides voice and data communications for its users.

System 85

An advanced digital switch supporting up to 3000 lines that provides voice and data communications for its users.

system administrator

The person assigned the responsibility of monitoring all system software processing, performing daily system operations and preventive maintenance, and troubleshooting errors as required.

system architecture

The manner in which the system software is structured.

system message

An event or alarm generated by either the system or an end-user process.

system monitor

A component of the system that tests to verify that each incoming telephone line and its associated circuit card is functional. Through the “System Monitor” component, you are able to see displays of the Voice Channel and Host Session Monitors.

T

T1

A digital transmission link with a capacity of 1.544 Mbps.

table

See database table.

tag image file format

A format for storing and exchanging digital image data associated with fax modem data transfers and other applications. These files can be identified by the .tif extension.

talkfile

An ASCII file that contains the speech phrase tags and phrase tag numbers for all the phrases of a specific application. The speech phrases are organized and stored in groups. Each talkfile can contain up to 65,535 phrases, and the speech filesystem can contain multiple talkfiles.

talkoff

The process of a caller interrupting a prompt, so the prompt message stops playing.

target system

The system to which you are upgrading (that is, your system as you expect it to exist *after* you upgrade).

TAS

transaction assembler script

TCC

Technology Control Center

TCP/IP

transmission control protocol/internet protocol

TDM

time division multiplexing

TE

terminal emulator

telephone network connection

The point at which a telephone network connection terminates on a system. Supported telephone connections are T1 and E1.

terminal emulator

Software that allows a PC or UNIX process to look like a specific type of terminal. In particular, it allows the system to temporarily transform itself into a “look alike” of an IBM 3270 terminal. In addition to providing full 3270 functionality, the terminal emulator enables you to transfer files to and from UNIX.

Text-to-Speech

An optional feature that allows an application to play US English speech directly from ASCII text by converting that text to synthesized speech. The text can be used for prompts or for text retrieved from a database or host, and can be spoken in an application with prerecorded speech.

ThickNet

A 10-mm (10BASE5) coaxial cable used to provide interLAN communications.

ThinNet

A 5-mm (10BASE2) coaxial cable used to provide interLAN communications.

TIFF

tag image file format

time-division multiplex

A method of serving a number of simultaneous channels over a common transmission path by assigning the transmission path sequentially to the channels, with each assignment being for a discrete time interval.

token ring

A ring type of local area network that allows any station in the network to communicate with any other station.

trace

A command that can be used to monitor the execution of a script.

traffic

The flow of information or messages through a communications network for voice, data, or audio services.

transaction

The interactions (exchanges) between the caller and the voice response system. A transaction can involve one or more telephone network connections and voice responses from the system. It can also involve one or more of the system optional features, such as speech recognition.

transaction assembler script

The computer program code that controls the application operating on the voice response system. The code can be produced from Voice@Work, Script Builder, or by writing directly in TAS code.

transaction state machine process

A multi-channel IRAPI application that runs applications controlled by TAS script code.

transient process

A process that is created dynamically only when needed.

troubleshooting

The process of locating and correcting errors in computer programs. This process is also referred to as debugging.

TSO

(1) Technical Services Organization; (2) time share operation

TSM

transaction state machine process

TTS

Text-to-Speech

TWIP

T1 interface process

U

UK

United Kingdom

US

United States of America

UNIX operating system

A multiuser, multitasking computer operating system originally developed by Lucent Technologies.

UNIX shell

The command language that provides a user interface to the UNIX operating system.

upgrade scenario

The particular combination of current hardware, software, application and target hardware, software, applications, and so on.

usability

A measurement of how easy an application is for callers to use. The measurement is made by making observations and by asking questions. An application should have high usability to be successful.

USOC

universal service ordering code

UVL

unified voice library

V

VDC

video display controller

vi editor

A screen editor used to create and change electronic files.

virtual channel

A channel that is not associated with an interface to the telephone network (T1, LSE1/LST1, or PRI). Virtual channels are intended to run “data-only” applications which do not interact with callers but may interact with DIPs. Voice or network functions (for example, coding or playing speech, call answer, origination, or transfer) will not work on a virtual channel. Virtual channel applications can be initiated only by a “virtual seizure” request to TSM from a DIP.

vocabulary

A collection of words that the system is able to recognize using either WholeWord, FlexWord, or Natural Language Speech Recognition.

vocabulary activation

The set of active vocabularies that define the words and wordlists known to the FlexWord recognizer.

vocabulary loading

The process of copying the vocabulary from the system where it was developed and adding it to the target system.

Voice@Work

An optional software package that provides a graphical interface to assist in the development of voice response applications on the system (see also Script Builder).

voice channel

A channel that is associated with an interface to the telephone network (T1, E1, LSE1/LST1, or PRI). Any system application can run on a voice channel. Voice channel applications can be initiated by being assigned to particular voice channels or dialed numbers to handle incoming calls or by a “soft seizure” request to TSM from a DIP or the **soft_szr** command.

voice processing co-marketer

A company licensed to purchase voice processing equipment to sell based on their own marketing strategies.

voice response output process

A software process that transfers digitized speech between system hardware (for example, SSP circuit cards) and data storage devices (for example, hard disk, and so on).

voice response unit

A computer connected to a telephone network that can play messages to callers, recognize caller inputs, access and update a databases, and transfer and monitor calls.

voice system administration

The means by which you are able to administer both voice-related and nonvoice-related aspects of the system.

VPC

voice processing co-marketer

VROP

voice response output process

VRU

voice response unit

W

warning

An admonishment or advisory statement used in the system documentation to alert the user to the possibility of equipment damage.

WholeWord speech recognition

An optional feature, available in several languages, based on whole-word technology that can recognize the numbers one through zero, “yes”, and “no” (the key words). This feature is reliable, regardless of the individual speaker. This feature can identify the key words when spoken in phrases with other words. A string of key words, called *connected digits*, can be recognized. During the prompt announcement, the caller can speak or use touchtones (or dial pulses, if available). See also whole-word technology.

whole-word technology

The ability to recognize an entire word, rather than just the phoneme or a part of a word. Compare to subword technology.

wink signal

An interruption of current to a busy lamp indicating that there is a line on hold.

word

A unique utterance understood by the recognizer.

wordlist

A set of words available for FlexWord recognition by an application during a Prompt & Collect action step.

word spotting

The ability to search through extraneous speech during a recognition.

Index

Numerals

A

B

C

D

E

F

H

I

L

T

U

V

W