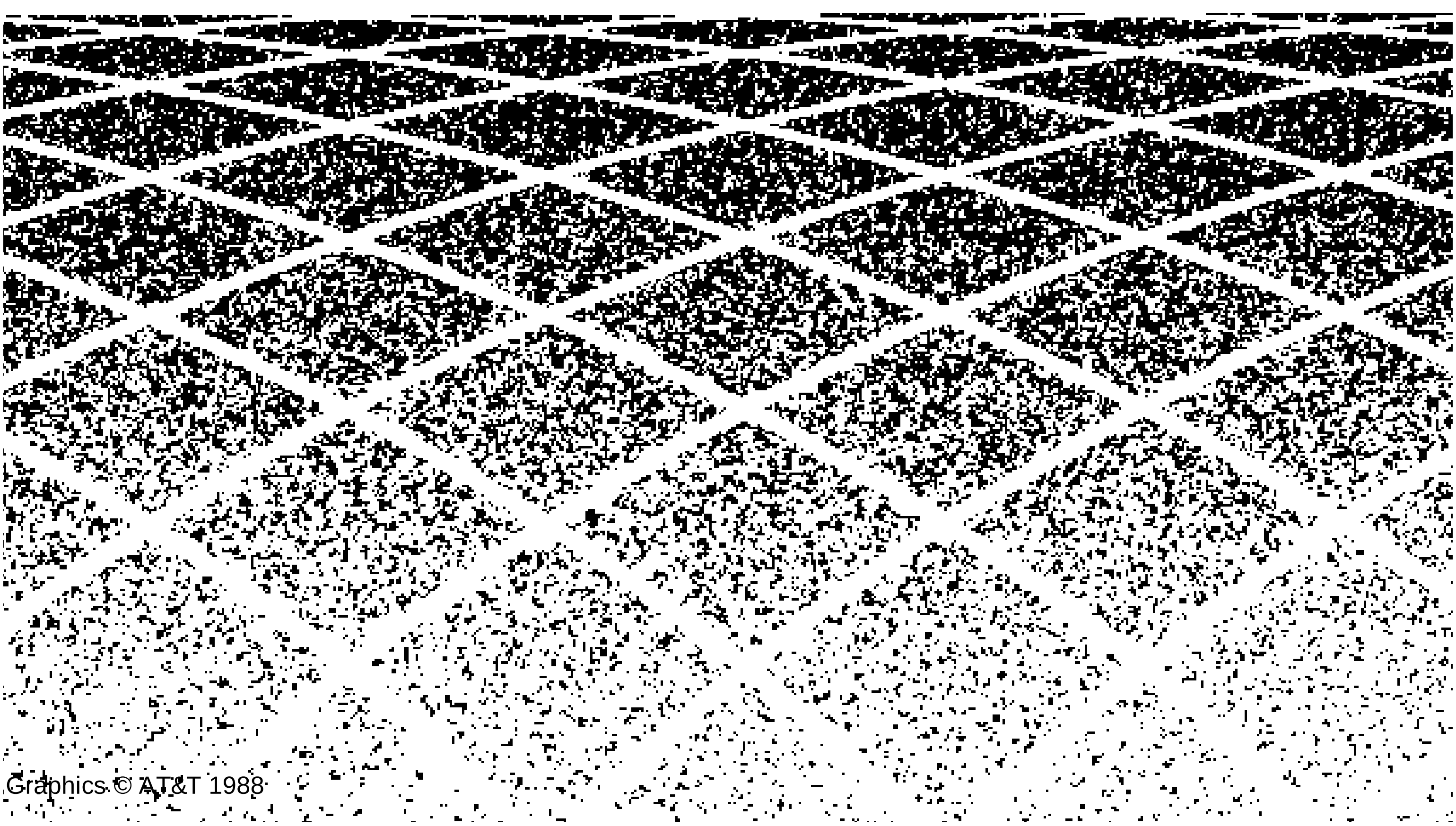
585-310-229
Issue 1
March, 1995

Conversant V5.0 Communication Development



Blank Page

Contents

Contents

Contents

Contents

Contents

Contents

Contents

Contents

Contents

Contents

About This Book

Purpose

This book is a reference manual for creating the necessary platform environment and applications to implement various communication interfaces between callers, administrators, and the AT&T Intuity™ CONVERSANT® Voice Information System (VIS).

How This Book Is Organized

This book is organized into the following sections:

- Chapter 1, "Intuity CONVERSANT VIS Communications Overview"

  This chapter provides a brief overview of the communications interfaces available within the VIS. This information includes both telephony and data network architectures.

- Chapter 2, "Analog Telephony Interfaces"

  This chapter 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 VIS.

- Chapter 3, "Digital Telephony Interfaces"

  This chapter 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 VIS.

- Chapter 4, "Adjunct/Switch Application Interface"

  This chapter 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 VIS.

- Chapter 5, "Converse Vector Step Routing"

  This chapter 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 VIS.

- Chapter 6, "Call Classification Analysis"

  This chapter 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 VIS.

- Chapter 7, "Data Network Communications"

  This chapter describes the potential uses of data network communications, discusses physical and logical protocol differences, and details what you must do on the VIS to implement this type of communication.

- Chapter 8, "Data Network Connectivity Alarms"

  This chapter describes the potential use of data network alarming and details what you must do on the VIS to implement this type of monitoring.

The book also includes a list of abbreviations, a glossary, and a cross-referenced index.

Conventions Used in This Book

The following typographic conventions are used in this book:

- Terminal keys
  - Terminal keys are shown in rounded boxes. For example, an instruction to press the enter key is shown as

    Press ENTER.
  - Function keys (also known as *soft* keys) are shown in rounded boxes followed by the function of that key in parentheses. For example, an instruction to press function key 3 is shown as

    Press F2 (CHOICES).
  - Two or three keys that you press at the same time (that is, you hold down the first key while pressing the second and/or third key) are shown as a series of rounded boxes. For example, an instruction to press and hold ALT while typing the letter **d** is shown as

    Press ALT D.
- User input
  - The word *enter* means to type a value and press ENTER. For example, an instruction to type **y** and press ENTER is shown as

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

    Type **y** to continue.

    Do *not* press ENTER after you type the value specified.
  - The word *select* is used to mean one of the following:
    a. Move to the desired menu item using the arrow keys and press ENTER. For example, an instruction to select an item from a menu and press ENTER is shown as

       Select Configuration Management from the Voice System Administration menu.

b. Type the first character of the item. The first menu item beginning with that letter is selected. If more than one item begins with the same letter, then type enough letters to identify the desired item. Press ENTER when the correct item is highlighted.

— Information that you enter or type from your terminal keyboard is shown in **bold** type; for example

Enter **root** at the `Console Login` prompt.

— Command and file names and their parameters are shown in **bold** type. Variable parameters are shown in ***bold italic*** type when they are part of a user input and in *regular italic* type when they are not. All are illustrated in the following example:

Use the **print** command to print your report. The command syntax is **print *reportname***, where *reportname* is the name of the report to be printed.

- Screen displays

— Information that is displayed on your terminal screen — including screen displays, prompts, script code, and system messages — is shown in `typewriter-style` type; for example

```
Installation is in progress -- do not remove
the floppy disk.
```

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

Begin at the CONVERSANT Administration menu, and select the following sequence:

> Voice System Administration

>Configuration Management

In this example, you would first access the CONVERSANT Administration menu. Then you would select the Voice System Administration option to display the Voice System Administration menu. From that menu, you would select the Configuration Management option to display the Configuration Management screen.

— The screens shown in the Intuity CONVERSANT library are only examples. Your screens may not appear exactly as illustrated.

Related Resources

The following books are to be used in conjunction with the *Intuity CONVERSANT VIS V5.0 Communication Development* book:

- *Intuity CONVERSANT VIS V5.0 Operations*, 585-310-550 — This book describes the Intuity CONVERSANT VIS V5.0 screen-based administrative interface. It describes each screen in detail, including the various fields and their possible values. Refer to this book when it is necessary to perform administrative procedures for various communication scenarios.
- *Intuity CONVERSANT VIS V5.0 Application Development*, 585-310-227 — This book describes application development using TSM script language. Refer to this book for information related to TSM script language.
- *Intuity CONVERSANT VIS V5.0 Script Builder*, 585-310-727 — This book describes application development using the VIS Script Builder. Refer to this book for information related to Script Builder, such as external actions and external functions.

Refer to the *Intuity CONVERSANT VIS Documentation Guide*, 585-350-002, for a complete list of VIS documentation.

Technical Updates

Every effort was made to ensure that the information contained in these books is technically accurate, and will guide readers in the normal operation of the system. There are instances however, when the Intuity CONVERSANT VIS V5.0 product may behave differently than is documented in the core library, or when hardware changes are made after these books have been published.

To help with this, an online bulletin board is available to all Intuity CONVERSANT VIS V5.0 customers that provides supplemental information about this product in an electronic, E-mail format. These updates include hints, tips, and exception conditions about all aspects of the Intuity CONVERSANT VIS V5.0 product that were discovered after the core library was published.

This service is called Access, and is available 24 hours-a-day, seven days-a-week to anyone who subscribes to it. To begin receiving electronic Intuity CONVERSANT VIS V5.0 Access articles, call 1-800-242-6005, and ask for department 186.

Trademarks and Service Marks

The following trademarked products are mentioned in this book:

- AUDIX, CONVERSANT, DEFINITY, 5ESS, and 4ESS are registered trademarks of AT&T.
- FlexWord, Intuity, and Voice Power are trademarks of AT&T.
- UNIX is a registered trademark of Novell, Inc.
- ORACLE, ORACLE*Terminal, OBJECT*SQL, SQL*FORMS, SQL*Menu, SQL*Net, SQL*Plus, PRO*C, and SQL*ReportWriter are trademarks of the Oracle Corporation.
- IBM is a registered trademark of International Business Machines.
- CLEO and LINKix are trademarks of CLEO Communications.
- Hayes and Smartmodem are trademarks of Hayes Microcomputer Products, Inc.

How to Make Comments About This Book

A reader comment card is included following the title page of this book. While we have tried to make this book fit your needs, we are interested in your suggestions for improving it and urge you to complete and return a reader comment card.

If the reader comment card has been removed from this book, please send your comments to:

AT&T
Product Documentation Development
Room 22-2C11
11900 North Pecos Street
Denver, Colorado 80234

Please include the name and document number of this book.

Intuity CONVERSANT VIS Communications Overview

1

What's in This Chapter

This chapter provides a brief overview of the communications interfaces available within the Intuity CONVERSANT Voice Information System (VIS). This information includes both telephony and data network architectures.

Communication Architecture

The Intuity CONVERSANT VIS connects to the public switch telephone network (PSTN) to communicate with external callers. In some VIS applications, it also connects to private data networks to access host computer databases for information required to complete certain types of calls.

This book provides descriptions for the following types of interfaces:

- Telephony
  - Analog
  - Digital
- Data Network
  - Asynchronous
  - Synchronous

This book also provides information on communications specifications, connectivity, typical cabling arrangements, administration, and switch integration parameters, and discusses application development issues.

Public Switched Telephone Network

The Intuity CONVERSANT VIS interface to the PSTN uses either an analog connection or a digital connection. An analog transmission is a method of sending signals in which the transmitted signal is similar to the original signal. A digital transmission is a method sending information in which the transmitted signal is encoded as a series of ones and zeros.

The following features use either anallg or digital connections:

- Adjunct/Switch Application Interface (ASAI)
- Converse Vector Step (CVS)
- Call Classification Analysis (CCA)

Chapters 2–6 should be read to learn more about the telephony interfaces used by the caller accessing the VIS application. Each of these chapters contain examples of how communications between the VIS and an external network are 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 a VIS application whether it was developed using Script Builder, transaction state machine (TSM) script language, or the Intuity Response Application Programming Interface (IRAPI).

Private Data Network

The Intuity CONVERSANT VIS supports two different forms of private data network interfaces: asynchronous and synchronous. These interfaces provide connections from the VIS to other computing devices such as remote monitoring systems or host computer databases. External customer interaction is not involved in these types of communications. Instead, the arrangement of these connections is based on the needs of the VIS application. These private data networks are transparent to the caller who is invoking the VIS application over the PSTN.

The following data network interfaces are supported and described in Chapter 7, "Data Network Communications", of this book:

- SNA 3270
- TCP/IP
- SQL*NET
- Physical asynchronous connections to the VIS platforms

Various data network alarming packages are also available for use in conjunction with the Intuity CONVERSANT VIS. The following packages are described in Chapter 8, "Data Network Connectivity Alarms", of this book:

- NetView
- CompuLert/SCCS
- External Alarms

Analog Telephony Interfaces

2

What's in This Chapter

This chapter describes the Tip/Ring (T/R) and FAX analog telephony interfaces available with the Intuity CONVERSANT Voice Information System (VIS) Version 5.0 base and optional software and the requirements that must be met to implement these interfaces.

This chapter also provides examples of typical analog connections.

Introduction to Analog Communications

In its analog configuration, the VIS provides nearly universal connectivity to existing private branch exchange (PBX) and automatic call distribution (ACD) premise equipment. It also allows standard interfaces to Centrex service offered by the domestic Local Exchange Carriers (LECs) and Public Switched Telephone Networks (PSTNs) maintained by countries outside of the U.S.

The following base analog telephony features make the Intuity CONVERSANT VIS compatible with a variety of domestic PBXs or ACDs (including the AT&T DEFINITY Communications Systems Generic 1, 2, and 3, System 85, System 75, System 25, Dimension 2000, etc.):

- The VIS can perform switch-hook-flash transfers (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, the system supports transfer with a bridge connection through the VIS. This bridging can be done with both digital and analog connections. Table 2-1 lists the analog line capabilities supported by call bridge.
- The VIS is capable of far-end caller disconnect detection through “wink signal” detection or an alternative time-out with dial-tone detection. Because these capabilities allow the VIS to know when a caller hangs up, it rarely transfers a “ghost” call.

  NOTE:
  Far-end caller disconnect detection through a wink signal or an alternative time-out must also be supported by the PBX or ACD. AT&T DEFINITY Communications Systems Generic 1, 2, and 3, System 85, System 75, System 25, and Dimension 2000 provide the signaling needed to notify the VIS of far-end caller disconnect. Other PBX systems may not.

- Outdialing for call transfer can be done with either touch tone or pulse.
- With custom software, the VIS 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 VIS flexibility

Table 2-1. Maximum Digital Trunks/Analog Lines Supported by Call Bridge

MAP/100, 100C		MAP/40	
Answer/Originate	**Outbound/Bridging**	**Answer/Originate**	**Outbound/Bridging**
60 digital T1 (linked)	60 digital T1	24 digital T1 (linked)	24 digital T1
36 analog T/R	36 analog T/R	24 analog T/R	24 analog T/R
72 analog T/R	48 digital T1	24 analog T/R	24 digital T1
96 digital T1	24 analog T/R	24 digital T1	24 analog T/R
96 digital T1	24 digital T1		

The following table details the maximum number of analog and digital line supported without call bridging.

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

	MAP/100, 100C Answer/Originate	MAP/40 Answer/Originate
Analog	72	48
Digital	96	48

Analog Connections to a 4ESS

Analog lines from the local service provider supply the physical interface between the switch and the Intuity CONVERANT VIS. The lines should be configured as a standard 2500 analog set on the switch. Refer to Chapter 6, “Switch Interface Administration,” of *Intuity CONVERSANT VIS V5.0 Operations*, 585-310-550, for an extended list of tunable parameters available with the various switch integration packages.

Analog Connections to AT&T PBXs

Analog connections between an Intuity CONVERSANT VIS 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 require analog connections. AUDIX Voice Power Coresidency and Adjunct/Switch Application Interface (ASAI) are two examples of features that rely on or use T/R interfaces between the VIS and the PBX.

The following settings and configuration data must be present on the PBX for analog T/R communication between the PBX and the VIS. The Intuity CONVERSANT VIS is designed to accommodate switch integration with AT&T System 75 switches as a default. Integration with other PBXs may require that you set specific switch integration values through the Voice System Administration menu. Refer to Chapter 6, “Switch Interface Administration,” of *Intuity CONVERSANT VIS V5.0 Operations*, 585-310-550, for an extended list of tunable parameters and valid values for various domestic 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 meet these standards. However, the LC03 circuit card on the DIMENSION and the SN229 circuit card on the System 85/G2 are *not* recommended for connection to the Intuity CONVERANT VIS.
- Each analog port on the switch must be configured to communicate as a standard 2500 analog set with the ability to transfer/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, the VIS ports must not have data restrictions in the COR, and “redirect notification” must be set to “y” if the Intuity CONVERANT VIS will be transferring calls to ACD splits staffed by Auto Answer (zip tone) agents.

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

- If you are using a MERLIN LEGEND:
  - All analog trunks receiving calls from and getting calls for the VIS must provide reliable disconnect.
  - All T/R lines originating from the MERLIN LEGEND switch connected to the VIS must be setup in a MERLIN LEGEND calling group as type “Generic VMI.”
  - You must administer the lines connected to the VIS with “outside line” dial tone. Refer to “Inside Dial Tone” in the *MERLIN LEGEND Communications System Installation, Programming, and Maintenance* for additional information.

Analog Connections to Other Switches

The Intuity CONVERSANT VIS 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 Intuity CONVERSANT VIS is essential for establishing communications between the two devices. Refer to Chapter 6, “Switch Interface Administration,” of *Intuity CONVERSANT VIS V5.0 Operations*, 585-310-550, for an extended list of tunable parameters. For specific values for each parameter, contact your local technical support organization.

T/R Interface

The T/R interface is provided through an analog (loop-start) T/R 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 T/R circuit card provides speech encoding and playback, dual tone multifrequency (DTMF) recognition, call supervision, and intraswitch call classification for intelligent transfers. Refer to "Introduction to Analog Communications" for additional information.

T/R Connectivity

Refer to the hardware installation book for your platform for information on installing a T/R circuit card. Figure 2-1 through Figure 2-3 show typical T/R connections from the Intuity CONVERSANT VIS.

NOTE:
The connectivity diagrams provide examples of T/R connections and are not the only method(s) of gaining connectivity to an external network. Actual network cabling varies on a site-by-site basis, and the cabling techniques used in each installation are the responsibility of the system administrator or installation technician.

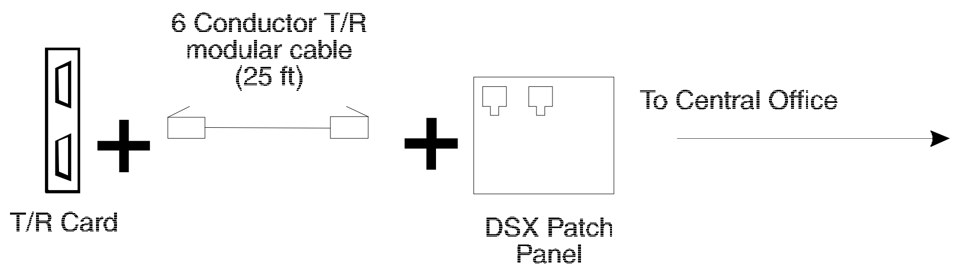


Figure 2-1. Analog T/R Interface Connection to a DSX Patch Panel

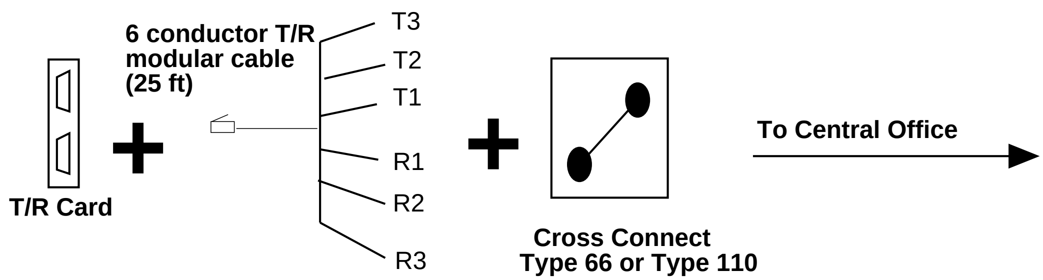


Figure 2-2. Analog T/R Interface Connection to a Type 66 or 110 Cross-Connect

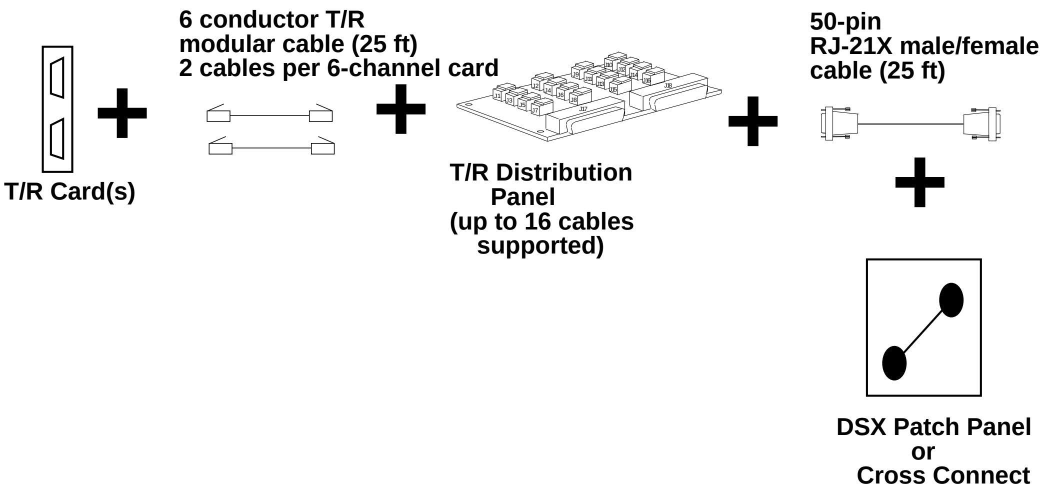


Figure 2-3. Analog T/R Interface Connection from Distribution Panel Using RJ21X Cable

T/R Telephony Interface Specifications

Table 2-3 through Table 2-6 detail the various T/R telephony interface specifications.

Table 2-3. T/R 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 T/R	0.5 B
Wink detection*†	80 – 800 msec
Flash duration*†	40 – 1550 msec
Register recall*	Timed break
Answer delay*	0 – 10 rings

*These attributes are adjustable through the Application Switch Interface (ASI) packages.

†These attributes can be changed via the Analog Interfaces screen described in Chapter 6, “Switch Interface Administration,” of *Intuity CONVERSANT VIS V5.0 Operations*, 585-310-550.

Table 2-4. T/R Circuit Card DTMF Tone-Detection Specifications

Attribute	Value
Digits	0 – 9, asterisk (*), pound sign (#), A – D
Amplitude	+3 to -22 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 %

Table 2-5. T/R Circuit Card DTMF Addressing Specifications

Attribute	Default Value
Digits	0 – 9, asterisk (*), pound sign (#), A – D
On/off timing*	100 msec on, 60 msec off
Frequency	Precise tones
Twist*	0 dB
Amplitude*	-3 dBm per frequency

*These attributes are adjustable through the Application Switch Interface (ASI) packages.

Table 2-6. T/R Circuit Card Call Progress Tone Detection Specifications

Tone	Frequency* (Hz)	Amplitude (dBm)	S/N Ratio (dB)	Maximum Twist (dB)	Frequency Deviation (%)	Cadence*
Dial	350 + 440 †	+1 to -24	55	+3	+/-0.5	Present for 1 sec
Stutter dial	350 + 440 †	+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	500–2000 msec on
Slow busy	480 + 620	+1 to -24	55	+3	+/-0.5	60 IPM, 250–500 msec on, 500–750 msec off
Fast busy	480 + 620	+1 to -24	55	+3	+/-0.5	120 IPM, 100–250 msec on, 250–450 msec off

*These attributes are adjustable through the Application Switch Interface (ASI) packages.

† These attributes are adjustable via the Analog Interfaces screen described in Chapter 6, “Switch Interface Administration,” of *Intuity CONVERSANT VIS V5.0 Operations*, 585-310-550.

Intuity CONVERSANT VIS Transmission Level Plan

A Transmission Level Plan (TLP) for a piece of telecommunications equipment is a set of specifications dictating the incoming/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 which is appropriate for listening without causing oscillations or distortions in the network.

Figure 2-4 shows most switch designs implement a TLP with a “built-in” gain of -3 dB (often called insertion loss) in each T/R loop of a station-set-to-station-set connection, for a total gain of -6 dB from end to end. The Intuity CONVERSANT VIS default TLP implements this same strategy; that is, the VIS default TLP attempts to make the end-to-end gain of voice signals passing through it equal to -6 dB.

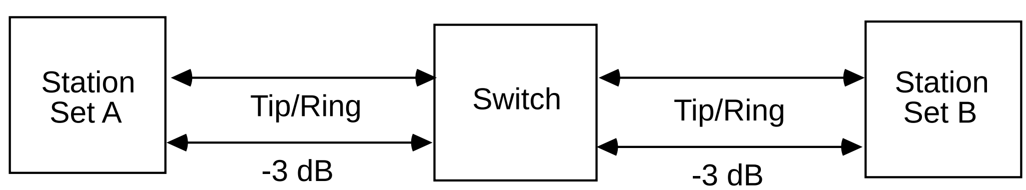


Figure 2-4. Typical Switch Transmission Level Plan for Station-Set-to-Station-Set Connection

Table 2-7. T/R 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 Interfaces screens as described in Chapter 6, “Switch Interface Administration,” of *Intuity CONVERSANT VIS V5.0 Operations*, 585-310-550. Gains can also be overridden on a per-channel basis by an Intuity Response Application Programming Interface (IRAPI) application. However, even with IRAPI, the IVOL cannot be overridden for speech recording on a T/R channel. Refer to the *Intuity CONVERSANT VIS V5.0 IRAPI Programming Guide*, 585-310-226, for the IRP_PLAYGAIN and IRP_RECORD_GAIN parameters under IrPARAMETERS(4IRAPI).

Intuity CONVERSANT VIS Network Interface Hardware

The Intuity CONVERSANT VIS connects to two types of telephone network facilities: analog (T/R) and digital (T1).

The Intuity CONVERSANT VIS default TLP is partially based on the following facts concerning VIS network interface hardware:

- VIS T1 interface circuit cards have a gain of 0 dB built into the hardware interface.
- VIS T/R 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 T/R and T1 network facilities have typical TLP characteristics associated with them. The VIS default TLP is partially based on the following typical network TLP characteristics:

- The VIS default TLP assumes a nominal 0-dB gain in each digital trunk connected to any T1 card in the system.
- The VIS default TLP assumes a nominal -3-dB gain in each analog loop connected to any T/R card in the system.

Incoming and Outgoing Speech Volume Nonbridging Modes

When a voice signal enters a VIS machine in a nonbridged connection, it is usually going to be coded and stored in the speech filesystem of the VIS machine. Before it is coded, its incoming volume can be adjusted by the IVOL parameter.

When a voice signal stored in the speech file system is played back from a VIS machine to a caller, its outgoing volume can be adjusted by the OVOL parameter.

The Intuity CONVERSANT VIS VERSION 5.0 Digital Interfaces screen allows the user to adjust both the incoming and outgoing speech volume for analog (T/R) and digital (T1) 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 circuit cards on a per card basis.

IVOL and OVOL should be thought of as volume multipliers (that is, +/- gain) of the incoming/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 -3 dB signal volume gain from the current volume (volume 3 dB lower); each multiplication of the current IVOL or OVOL setting by a factor of 1.414 results in a +3 dB signal volume gain from the current volume (volume 3 dB higher).

NOTE:
IVOL and OVOL affect only signals being coded or played back by the VIS. They do not affect end-to-end conversations in call bridge mode.

Table 2-8 shows the IVOL and OVOL settings required to implement the VIS default TLP along with the actual gain in decibels (shown in parenthesis) that each setting represents.

Table 2-8. Default System IVOL and OVOL Settings

Network Facilities	IVOL	OVOL	Text-to-Speech (TTS) OVOL*
Analog	2000(+6)	1000(0)	4000
Digital	1414(+3)	707(-3)	1000†

*The TTS OVOL is an option only when the TTS package is installed.

†The TTS OVOL default value may be too low in some cases. You may want to use a higher value. However, if a value is too high, it may cause distortion of the outgoing text.

Voice Coding and Play

As described above, most switches build in -dB of gain in a typical station-set-to-station-set connection. With a VIS in a nonbridging mode, station-set-to-station-set actually involves a signal being affected by IVOL while it is coded and stored on the disk, then affected by OVOL when it is played back. To be in accordance with the VIS TLP, the level the caller hears during should be dB lower than the level that was spoken when the signal was coded.

Voice Coding

Figure 2-5 shows how the default IVOL parameters control the level at which a voice signal is coded and stored in the VIS speech filesystem.

The top part of Figure 2-5 shows a T1 interface connected to the VIS; the bottom part shows a T/R interface connected to the VIS. 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 VIS.

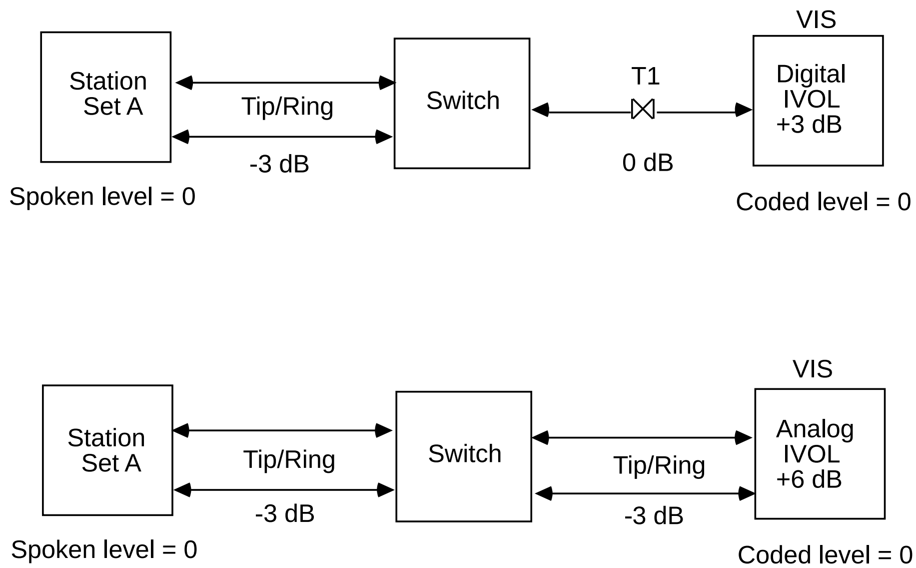


Figure 2-5. Effect of IVOL Parameters Voice Coding

Voice Play

Figure 2-6 shows how the default OVOL parameters control the level at which a previously coded voice signal stored in the VIS speech filesystem is played back.

The top part of Figure 2-6 shows a T1 interface connected to the VIS; the bottom part shows a T/R interface connected to the VIS. As you follow the signal from right to left, if the signal was coded in the manner depicted in Figure 2-5, 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 VIS. Since the initial spoken level shown in Figure 2-5 was 0, the heard level of -6 is in accordance with the VIS TLP.

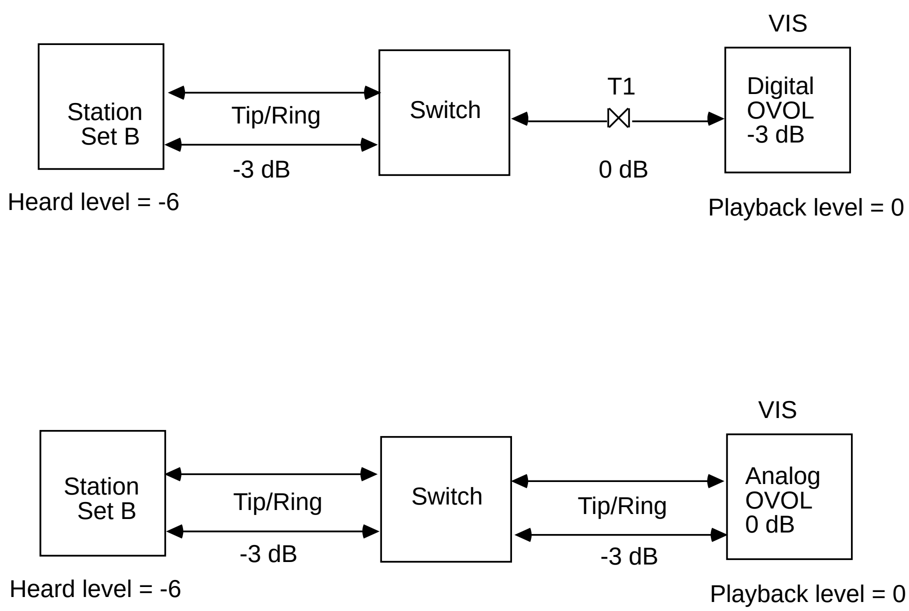


Figure 2-6. Effect OVOL Parameters Voice Play

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 2-5.

In many cases, however, speech prompts are coded in a studio at higher volumes than they would have been coded from a VIS network interface. In these situations, it may be desirable to decrease the applicable OVOL parameter (analog or digital, depending on whether playback is from T/R 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/or 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 equal to the additional insertion loss. For example, if the digital trunks connected to a VIS had insertion loss of -3 dB instead of 0 associated with them as the default VIS 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 (refer to Table 2-8 and the accompanying explanation). Making these changes results in meeting the TLP goal of -6 dB gain from end to end.

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 VIS, the callers on either end (station set A and station set B) can talk with each other through the VIS. In such a situation, the previously discussed IVOL and OVOL parameters do not apply. Instead, software on the VIS machine (specifically the TSM process) has built in rules for directing the VIS Network Interface cards to insert up to +6 dB gain in either direction of a call bridge connection.

Figure 2-7 through Figure 2-10 depict the rules governing the amount of gain inserted. Recall that the VIS 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 the previous section), Figure 2-7 through

Figure 2-10 show the amount of gain (in dB) that is automatically inserted in each direction for each of the four possible call bridging scenarios.

Figure 2-7 shows analog-to-analog (T/R-to-T/R) call bridging. Figure 2-8 shows digital-to-digital (T1-to-T1) call bridging. Figure 2-9 shows analog-to-digital (T/R-to-T1) call bridging. Figure 2-10 shows digital-to-analog (T1-to-T/R) call bridging.

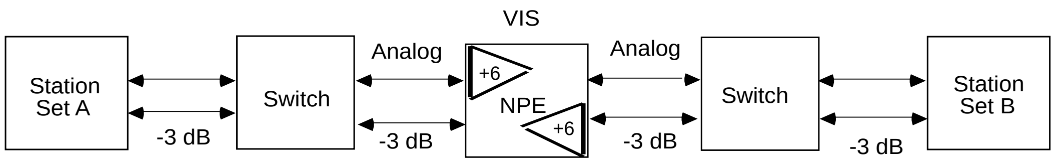


Figure 2-7. Analog-to-Analog Call Bridging

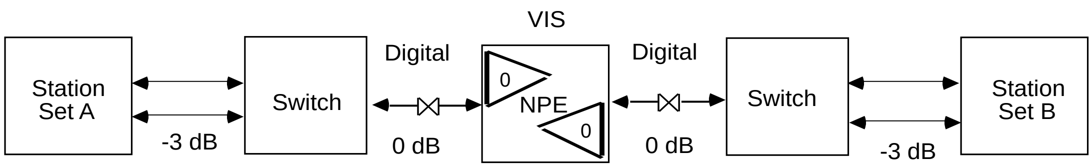


Figure 2-8. Digital-to-Digital Call Bridging

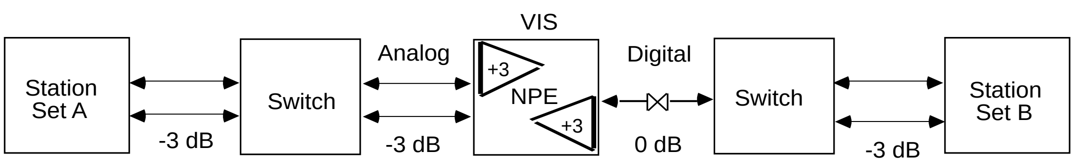


Figure 2-9. Analog-to-Digital Call Bridging

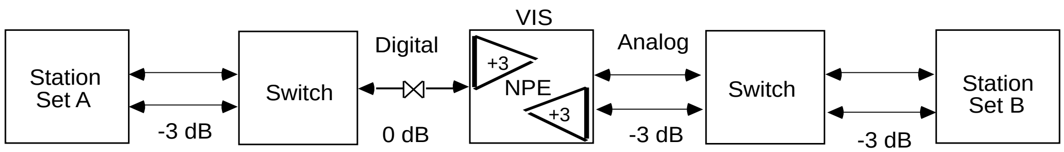


Figure 2-10. Digital-to-Analog Call Bridging

Possible Exceptions to the Intuity CONVERSANT VIS TLP

When a VIS is used as a network adjunct within the AT&T 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 Intuity CONVERSANT VIS TLP.

In addition to the 6-dB end-to-end loss described above, the FCC requires that the 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 AT&T 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 an Intuity CONVERSANT VIS is being used as a network adjunct and digital trunks are used, it is recommended that IVOL and OVOL settings be set to the non-default value of 1000 (for zero gain and that prerecorded speech be recorded at -21 dBm0. By using zero gain, the VIS 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 signalling (see Chapter 3, "Digital Telephony Interfaces"), where the least significant bits rather than voice data are used for signalling. 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 recorded via the long distance network. This tone is lost in an analog bridge or recording 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 disable.

To prevent problems with excessive volume levels from enhanced AT&T Truevoice processing, it is recommended that recordings and announcements be recorded in studio labs (rather than via the network) and that the low frequencies should not be enhanced by the studio.

T/R Switch Integration Issues

Switch integration for T/R circuit cards is done using the Analog Interfaces screen. This screen is described in Chapter 6, “Switch Interface Administration,” of *Intuity CONVERSANT VIS V5.0 Operations*, 585-310-550. The T/R interface is administered on a system-wide basis, that is, the T/R parameters apply to all T/R circuit cards. To administer the T/R interface, you may 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 6, “Switch Interface Administration,” of *Intuity CONVERSANT VIS V5.0 Operations*, 585-310-550. 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

T/R channel. Refer to the *Intuity CONVERSANT VIS V5.0 IRAPI Programming Guide*, 585-310-226, for the IRP_PLAYGAIN and IRP_RECORD_GAIN parameters under IrPARAMETERS(4IRAPI).

All T/R lines originating from the Merlin Legend switch connected to the VIS must be setup in a Merlin Legend calling group as type “Generic VMI.”

Refer to "Introduction to Analog Communications" at the beginning of this chapter for a discussion of the base analog telephony features of the VIS.

T/R Circuit Card Administration

Placing a card in the INSERV state allows it to be used for the purpose (play, code, etc.) for which it is allocated in the application. You may need to *manually* place a T/R 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 T/R cards to INSERV, use the steps described in Chapter 3, “Configuration Management,” of *Intuity CONVERSANT VIS V5.0 Operations*, 585-310-550.

FAX Interface

Facsimile (FAX) communications involve transmitting graphic and text images between FAX machines and other devices via standard telecommunications networks.

Intuity CONVERSANT VIS V5.0 supports the coresidency of AT&T FAX Attendant R2.5 hardware and software. The FAX Attendant system processes fax messages and controls announcements that are stored on disk. Integration with VIS applications occurs through the use of the Script Builder FAX Actions.

FAX Attendant provides FAX messaging and FAX-on-demand services. It works much like the VIS except that a fax message is the input and output of the system instead of voice and a FAX machine is the receiver of the message instead of a human. FAX Attendant uses the voice capabilities of the VIS to prompt callers through menu choices. Faxes are sent or received via a FAX circuit card installed in the VIS.

For a general discussion of FAX Attendant and the Script Builder FAX Actions, refer to the *Intuity CONVERSANT VIS V5.0 System Description*, 585-310-225.

FAX Provisioning

Intuity CONVERSANT VIS V5.0 supports a maximum of 12 FAX channels for FAX Attendant coresidency through the use of the Brooktrout TR114 and TR114+ (both are four-channel circuit cards). Up to three circuit cards may be used. Because of hardware limitations, Intuity CONVERSANT VIS V5.0 does not support the migration of a deployed FAX Attendant product to a new or existing Intuity CONVERSANT VIS V5.0 platform. It is not recommended that you assign FAX Attendant Services to T1 channels because T1 does not support flash-hook transfers. However, applications that use the Script Builder FAX Actions can be assigned to T/R, T1, or Line Side T1 (LST1) channels.

Refer to the hardware installation book for your platform for information on installing the FAX circuit card in the VIS.

FAX Administration

FAX Attendant administrative screens are accessible directly from the VIS main menu screens. The following must be administered to implement the FAX Attendant coresidency:

- Assign services to voice channels
- Assign extensions to FAX channels
- Administer parameters for FAX and the voice system
- Administer the PBX (in some cases)
- Load faxes into the system, if desired

Refer to the R2.1.1 FAX Attendant documentation for information about preforming the above listed procedures.

NOTE:
Any references to FACE in the R2.1.1 FAX Attendant documentation are obsolete. You should use the UnixWare System Administration screens to perform those functions that were previously done using FACE. Refer to the *Novell UnixWare Documentation Set*, 585-350-908, for information aobut using UnixWare.

FAX Switch Integration Issues

FAX Attendant can operate in either switch-integrated or non-switch-integrated mode. In both modes, spoken information is transferred over analog voice channels. In switch-integrated mode, caller and called-party identification information is transferred from the switch. In non-switch-integrated mode, the interface between the telephone switch and FAX attendant does not include caller or called-party identification information.

The necessary planning for and administration of the switch in a FAX Attendant system is discussed in the *Implementation and Switch Notes* for your telephone system that accompanies the R2.1.1 Fax Attendant documentation.

NOTE:
Extensive system and telephony/PBX administration may be necessary when using the full functionality of the AT&T FAX Attendant System. However, you may use the Script Builder FAX Actions without performing most of this administration. Refer to *Intuity CONVERSANT VIS V5.0 Operations*, 585-310-550, for information on administering FAX actions. Refer to the R2.1.1 FAX Attendant documentation for additional information on administering the FAX Attendant System and the PBX. Refer to Chapter 6, “Using Optional Features with Script Builder,” of *Intuity CONVERSANT VIS V5.0 Script Builder*, 585-310-727, for additional information on the using the Script Builder FAX Actions.

FAX Application Development Issues

With FAX Attendant coresidency, the VIS can invoke FAX Attendant services through Script Builder applications.

Script Builder FAX Actions

The Script Builder FAX Actions allow you to include FAX communications in any Script Builder application. The VIS must be coresident with the AT&T FAX Attendant System and have Script Builder installed. The Script Builder 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 Script Builder FAX Actions, including the benefits and uses, refer to the *Intuity CONVERSANT VIS V5.0 System Description*, 585-310-225. For detailed information about implementing Script Builder FAX Actions in Intuity CONVERSANT VIS applications, refer to Chapter 6, “Using Optional Features with Script Builder,” of *Intuity CONVERSANT VIS V5.0 Script Builder*, 585-310-727.

Digital Telephony Interfaces

3

What's in This Chapter

This chapter describes the T1, Line Side T1 (LST1), and Primary Rate Interface (PRI) digital telephony interfaces available with the Intuity CONVERSANT Voice Information System (VIS) Version 5.0 base and 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 circuit (trunk) allows the VIS to connect to digital network facilities such as a central office (CO) switch. 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.

T1 connections also provide dialed number identification service (DNIS) information to further automate incoming calls for customers with multiple 800 or 900 numbers. Table 2-2 shows the maximum number of digital lines that are supported on each of the Intuity CONVERSANT VIS platforms.

T1 and Integrated Services Digital Network (ISDN) PRI support trunk interfaces. PRI is the ISDN equivalent of a T1 circuit It contains 23 B+D channels. Each B channel operates at 64 Kbps, and is the equivalent of one incoming data line. The D channel does not provide normal telephony service, but is used to provide advanced control information, such as DNIS, for each of the other 23 incoming lines.

Digital T1 interfaces also support the line-side connection of a VIS and a PBX. This LST1 also supports the Adjunct/Switch Application Interface (ASAI) feature.

NOTE:
ASAI is only supported on LST1 when used with DEFINITY. Refer to "Line Side T1 Interface" later in this chapter for additional information.

The system supports call bridging through a digital connection. Call bridges can also be used to simulate a transfer, but this consumes channel resources. Table 2-1 lists the digital line capabilities that call bridge supports.

Digital Telephony Interface Specifications

Table 3-1 details the general digital telephony interface specifications for all T1 protocols.

Table 3-1. Digital Telephony Interface General Specifications

Attribute	Specification
Physical connector	Subminiature DB-15 male receptacle
FCC registration	AS593M-17926-VM-E
Safety approval	UL Type Approved
Signal regeneration	CSU required over 655 feet.
Loopback capability	CSU required for remote capability
Transmission Level Point (TLP) at DS-1 interface	0 ELP, 0 DLP
TLP at time-division multi-plexed (TDM) interface	0 ELP, 0 DLP
Call progress tone frequency	Precise tone frequencies
Call progress tone levels	-6 dBm total (nominal)
Call progress tone timing	Ringing - on/off: 2 sec on, 4 sec off Busy - on/off: 0.5 sec on, 0.5 sec off
Call progress tone detection	Not supported by T1 card; must use optional CCA feature if this capability is required. [Note: Even with Full Call Classification Analysis (CCA), LST1 does not detect dial tone.]
DS-1 timing source	Slave to DS-1 source (loop timed)
DS-1 timing (free running)	Stratum 4
Suggested channel service unit (CSU) types	AT&T Paradyne (PEC 2152-ESF), Verilink 551VST List 2, or equivalent
Supported configurations	Tie trunk (robbed-bit E&M), ISDN-PRI, LST1
Dual tone multifrequency (DTMF) output timing	70 msec on, 70 msec off
DTMF output levels	-8 dBm per frequency (nominal)
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).
Number of receivers	24 (one per DS-0 channel)

Channel Service Unit Connectivity

The T1 interface circuit card is connected to a channel service unit (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 Intuity CONVERSANT VIS is more than 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 Intuity CONVERSANT VIS is terminating the T1 trunk from outside the building. Here, the CSU provides the primary lightning and surge protection as required by FCC Part 68.
- The T1 loop is not dry (that is, 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 15-pin connector on the T1 cable can plug into the AYC3B or AYC11 circuit card and terminate via a 15-pin D subminiature connector to the CSU. On the other types, you must cut off the cable end with the 15-pin plug and slide latch and strip and connect the wires using the following information:

- Orange = our T1 = signals to CONVERSANT and should connect to a CSU or Network “T”
- White/Orange = our R1 = signals to CONVERSANT and should connect to a CSU or Network “R”
- Green = our T = signals from CONVERSANT and should connect to a CSU or Network “T1”
- White/Green = our R= signals from CONVERSANT and should connect to a CSU or Network R1

Refer to Figure 3-1 and Figure 3-2 for CSU connectivity examples.

T1 E&M Interface

The T1 circuit card accepts an ISDN PRI or DS-1 two-way digital trunk and converts it to two-way analog audio channels. Because of bandwidth and transmission differences of each trunk, ISDN PRI and DS-1 offer different amounts 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.

T1 Connectivity

The MAP/100 and MAP/100C support up to five T1 circuit cards. The MAP/40 supports two T1 circuit cards. A Signal Processor (SP) circuit card is required if you are using one or more T1 circuit cards in coding and playback situations.

NOTE:
Each SP circuit card supports up to 31 channels of simultaneous speech playback using adaptive differential pulse code modulation (ADPCM) 32-Kbps coded speech. Applications that require large amounts (more than 30 channels) of simultaneous speech coding may require additional SP circuit cards.

Refer to the hardware installation book for your platform for information on installing T1 and SP circuit cards.

Figure 3-1 and Figure 3-2 show examples of typical T1 connections to a T1 trunk. Table 3-2 details the digital telephony specification for the T1.5 Robbed-bit E&M protocol. Use this table in conjunction with Table 3-1.

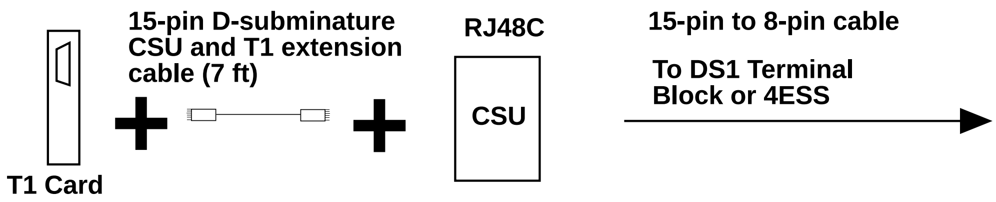


Figure 3-1. T1 Interface Connection to a CSU with a 15-Pin Connector

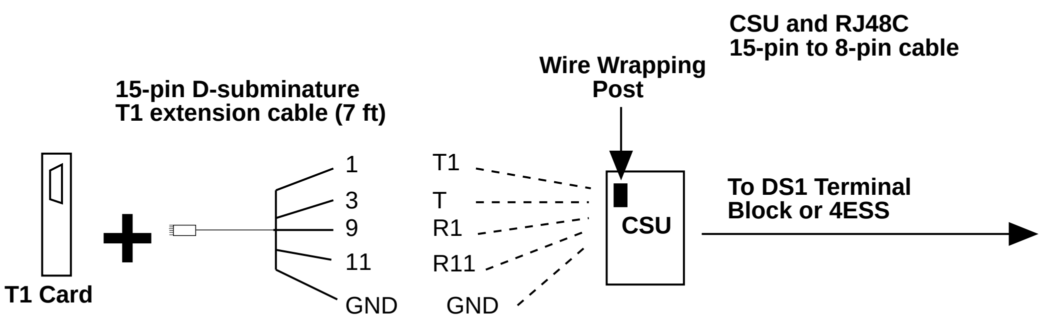


Figure 3-2. T1 Interface Connection to a CSU with Wire Wrapping Posts

Table 3-2. Digital Telephony Interface Specifications for Tie Trunk 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	100 – 350 msec
Addressing (outgoing)	DTMF (touch tone)
Number of digits	16-digit maximum
Addressing (incoming)	DTMF (touch tone)
Number of digits (DNIS)	Will wait for up to 16 digits (selectable). (This specification can also be provisioned not to wait for digits.)
Initial digit timer	Will wait up to 4 seconds for first digit. (This specification can also be provisioned not to wait for digits.)
Interdigital timer	Will wait up to 2 sec 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	Not supported

T1 Switch Integration and Administration

Switch integration for T1 is done using the Digital Interfaces screen. This screen is described in Chapter 6, “Switch Interface Administration,” of *Intuity CONVERSANT VIS V5.0 Operations*, 585-310-550. You must select T1 A/B Robbed-bit E&M Protocol from the Digital Interfaces screen. Refer to "Line Side T1 Interface" and "Primary Rate Interface" below 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 steps described in Chapter 3, “Configuration Management,” of *Intuity CONVERSANT VIS V5.0 Operations*, 585-310-550.

T1 Application Development Issues

The Intuity CONVERSANT VIS does not support flash transfers using T1 configurations (except when LST1 is accompanied by Full CCA, refer to "Line Side T1 Interface" below). Simulated T1 transfers can be performed only over call bridges. In the analog Tip/Ring (T/R) environment, the switch-hook-flash transfer releases the call from the Intuity CONVERSANT VIS once the transfer is made. A call bridge, however, ties up an incoming T1 port and an outgoing port until the call has concluded. Thus, with two ports being tied up simultaneously, more T1 ports may be necessary.

Script Language

The **tic** instruction is used for basic control of incoming and outgoing calls on T1 lines. For additional information about using the transaction state machine (TSM) script language on T1 lines, refer to the **tic** instruction in Chapter 3, “Script Instructions,” and Appendix A, “Summary of Script Instructions,” of *Intuity CONVERSANT VIS V5.0 Application Development*, 585-310-227.

Intuity Response Application Programming Interface

The **irCall()**, **irAnswer()**, **irDial()**, and **irDisconnect()** functions provide the basic call control capabilities for T1 interfaces with the Intuity Response Application Programming Interface (IRAPI). The **irStartSpeechED()** function is not supported for T1 interfaces. Refer to the *Intuity CONVERSANT VIS V5.0 IRAPI Programming Guide*, 585-310-226, for more information about these functions when developing IRAPI applications.

Line Side T1 Interface

Line Side T1 (LST1) allows the use of a 24-channel, 1.544-MHz digital interface between a customer PBX and the Intuity CONVERSANT VIS V5.0 platform. LST1 uses existing T1 card technology with new protocol-level software and VIS user interface modifications to improve system connectivity and reduce the number of circuit cards and cables required to support 24 channels of service.

LST1 is compatible with DEFINITY G3 PBX and Galaxy 8 Automatic Call Distributing (ACD) systems. LST1 also supports the ASAI feature when used with DEFINITY G3 PBX.

LST1 Provisioning

Existing configurations require 24 analog connections between the VIS and the PBX, whereas LST1 requires only one cable to provide 24 channels of service.

LST1 also significantly reduces the number of VIS cards required to support a VIS-PBX interface. Analog configurations require four IVP6 circuit cards or six IVP4 circuit cards to support 24 incoming channels. LST1 reduces the required hardware to only one T1 circuit card and one SP circuit card per 24 channels of digital service. Two T1 circuit cards and one SP circuit card provides 48 channels of voice on LST1. LST1 hardware and software support both existing T1 circuit cards (AYC3B and AYC11).

When LST1 is used to provide an ASAI link between the VIS and a PBX, an IPCI circuit card and connection must also be used to provide the Basic Rate Interface (BRI) D-channel link. A single LST1 channel is not held back for D-channel use as is done with ordinary PRI connections.

The following limitations apply when you use an LST1 interface:

- LST1 cannot provide dial tone or stutter dial tone detection prior to dialing. This is true whether LST1 is used with or without Full CCA.

  NOTE:
  If Full CCA is installed on the VIS, the **tic('O')** and **tic('D')** instructions use the return codes as described in Appendix A, "Summary of Script Instructions," of *Intuity CONVERSANT VIS V5.0 Application Development*, 585-310-227.

- Without the use of Full CCA, LST1 cannot detect call progress tones (CPTs) after dialing.
- AUDIX Voice Power and FAX Attendant coresidency scripts use dial-tone and call-progress-tone detection, and therefore are not compatible with systems using an LST1 VIS-to-PBX interface.
- When a switch is excessively loaded and a timed delay is used prior to dialing, a call can be lost if the PBX is not properly engineered and administered.
- The VIS cannot detect glare conditions caused by an incoming and outgoing call attempting to use the same channel due to the lack of dial-tone detection.
- Dial pulse is not supported on LST1 channels (or any T1); however, dialing of DTMF tones is supported.
- The DEFINITY G2 does not provide forward disconnect.

Table 3-3 details the digital telephony interface specifications for ISDN-PRI type configurations. Use this table in conjunction with Table 3-1.

Table 3-3. Digital Telephony Interface Specifications for LST1 Configurations

Attribute	Specification
DS-1 framing	D4 type only
DS-1 line coding	ZCS
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 converse vector step (CVS) or ASAI
ANI capacity	No supported unless used with CVS or ASAI
Transfer capability	Flash transfers supported

Using LST1 for Converse Vector Step

The in-band DNIS capability is available when using the converse vector step (CVS) feature of DEFINITY on LST1 channels. Refer to Chapter 5, "Converse Vector Step Routing", for additional information.

LST1 Switch Integration and Administration

Switch integration for LST1 is done using the Digital Interfaces screen. This screen is described in Chapter 6, “Switch Interface Administration,” of *Intuity CONVERSANT VIS V5.0 Operations*, 585-310-550. You must select Line Side T1 Protocol for DEFINITY or Galaxy from the Digital Interfaces screen.

In addition to the Digital Interfaces parameters, the following administration must be performed in the Analog Interfaces screen.

- In the Blind Transfer Actions field, you must set the To Initiate Transfer and To Complete Transfer fields to FP (flash and pause for a fixed delay) and H (hang-up).
- If you are using Full CCA, you must set the following under Intelligent Transfer Actions field:
  - — For DEFINITY, set the To Reconnect Caller field to FPF
  - — For Galaxy, set the To Reconnect Caller field o P.

Refer to Chapter 6, “Switch Interface Administration,” of *Intuity CONVERSANT VIS V5.0 Operations*, 585-310-550, for additional information.

Placing a card in the INSERV state allows it to be used for the purpose (play, code, etc.) for which it is allocated in the application. After performing switch integration on the T1 circuit card for the LST1 protocol, you may need to *manually* place a T/R 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 steps described in Chapter 3, “Configuration Management,” of *Intuity CONVERSANT VIS V5.0 Operations*, 585-310-550.

LST1 Application Development Issues

The following are LST1 application development issues for Script Builder, script language, and the IRAPI.

Script Builder

LST1 supports blind call origination (outcalling) and blind call transfers for DEFINITY PBXs and Galaxy ACDs only as normally performed on T/R lines. Blind transfers mean that the Intuity CONVERSANT VIS will not detect call-progress tones or provide any form of answer supervision. LST1 can provide CPT detection only when used with Full CCA. It cannot support CPT detection without Full CCA.

Because of this, LST1 does not support Script Builder requests for intelligent call originations and transfers. You must convert existing scripts which use intelligent call origination/transfer requests (DEFINITY PBXs and Galaxy ACDs only) to blind or Full CCA requests to achieve LST1 compatibility.

Script Language

The following script instructions support LST1 operations:

- **tic(‘o’)**
- **tic (‘O’)** (only supported with the use of Full CCA)
- **tic(‘f)**
- **tic(‘F’)**
- **tic(‘d’)**
- **tic(‘D’)** (only supported with the use of Full CCA)
- **tic(‘h’)**

Refer to the **tic** instruction in Chapter 3, “Script Instructions,” and Appendix A, “Summary of Script Instructions,” of *Intuity CONVERSANT VIS V5.0 Application Development*, 585-310-227, for additional information.

IRAPI

Refer to "T1 Application Development Issues" above for details on the supported IRAPI functions for T1 interfaces. Refer to the *Intuity CONVERSANT VIS V5.0 IRAPI Programming Guide*, 585-310-226, for more information about these functions when developing IRAPI applications using LST1.

Primary Rate Interface

The Intuity CONVERSANT VIS supports the ISDN-PRI 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 VIS supports this digital ISDN communication with ISDN-PRI Layer 1 protocol rather than the T1 A/B Robbed-bit E&M Protocol used with standard T1 communications. The ISDN-PRI Layer 1 protocol uses either D4 or ESF framing. Standard T1 circuit card connectivity, as described in the previous pages, is used to implement the physical connection between the VIS and the remote network entity when using ISDN-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 *non-inverted.* The ISDN-PRI service provider determines the method of framing used.

The Intuity CONVERSANT VIS does not support flash transfers using PRI configurations. Simulated T1 transfers can be performed only over call bridges. In the analog Tip/Ring (T/R) environment, the switch-hook-flash transfer releases the call from the Intuity CONVERSANT VIS once the transfer is made. A call bridge, however, ties up an incoming T1 port and an outgoing port until the call has concluded. Thus, with two ports being tied up simultaneously, more T1 ports may be necessary.

Table 3-4 details the digital telephony interface specifications for ISDN-PRI type configurations. Use this table in conjunction with Table 3-1.

Table 3-4. Digital Telephony Interface Specifications for ISDN-PRI Type Configurations

Attribute	**Specification**
DS-1 framing	D4 or ESF (selectable)
DS-1 line coding	ZCS (with D4 framing only) B8ZS (with ESF framing only)
B-channel capacities	23 B+D, 47 B+D, 71 B+D, 95 B+D, or 119 B+D when up to five T1 cards are provided (see the *Intuity CONVERSANT VIS V5.0 System Description*, 585-310-225, for a list of platform limitations). Note: These configurations are switch dependent as not all switches support all configurations.
D-channel capacities	Maximum of one D channel per system (for example, two 23 B+D interfaces are not supported)
Interface ID	1 (for a card with a D-channel, not selectable) 2 (for a card without a D-channel)
DNIS capacity	0–16 digits
ANI capacity	0–16 digits
D-channel backup	Not supported
Transfer capability	Not supported

PRI Provisioning

Supported B-channel capacities in PRI configurations are switch dependent (see Table 3-4) as not all switches support all configurations. For example, the 5ESS switch only supports the 23 B+D configuration, but the 4ESS switch can support all five configurations. Refer to the *Intuity CONVERSANT VIS V5.0 System Description*, 585-310-225, 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 VIS feature. In addition, the Intuity CONVERSANT VIS uses some Layer 2 and Layer 3 parameters that must be correct and matching in both machines. Table 3-5 and Table 3-6 show how to set these parameters on the switch.

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

Table 3-5. PRI Layer 2 Parameters

Parameter	Value
Retry Count N200	3
Timer T200	1 sec
Timer T203	30 sec
High-level data link control or HDLC (D4/ZCS)	Inverted
HDLC (ESF/B8ZS)	Noninverted

Table 3-6. PRI Layer 3 Parameters

Parameter	Value
Timer T302	10 sec
Timer T303	4 sec
Timer T305	4 sec
Timer T308	4 sec
Timer T310	10 sec
Timer T313	4 sec
Timer T316	120 sec
Timer TL3	120 sec
Timer T309	10 sec
Interface ID (with D-channel)	1
Interface ID (without D-channel)	2
Bearer capability	64 Kbps voice

PRI Switch Integration and Administration

Switch integration for the PRI feature is done using the Digital Interfaces screen. This screen is described in Chapter 6, "Switch Interface Administration," of *Intuity CONVERSANT VIS V5.0 Operations*, 585-310-550. You must select T1 ISDN-PRI Layer 1 Protocol from the Digital Interfaces screen.

Also, you must assign one SP circuit card to process the PRI protocol and one or more T1 circuit cards to provide the physical layer for the PRI.

To assign PRI functionality to an SP circuit card, refer to Chapter 3, "Configuration Management," of *Intuity CONVERSANT VIS V5.0 Operations*, 585-310-550.To assign PRI functionality to a T1 circuit card, refer to Chapter 6, "Switch Interface Administration," of *Intuity CONVERSANT VIS V5.0 Operations*, 585-310-550.

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. (Refer to the information on assigning the PRI Layer 1 Protocol to a T1 circuit card in Chapter 6, "Switch Interface Administration," of *Intuity CONVERSANT VIS V5.0 Operations*, 585-310-550, for more details). The D-channel cannot be used to run applications. It carries messages between the switch and the VIS. 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 has been 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 circuit T1 card, the additional T1 cards (the ones configured without a D-channel) will have 24 B-channels on which to run applications. The VIS can run applications on a total of 119 B-channels (that is, five T1 cards). 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.

Determining the D-Channel

If you do not know which channel is the D-channel, perform the following procedure. Refer to Chapter 3, “Configuration Management,” of *Intuity CONVERSANT VIS V5.0 Operations*, 585-310-550, for more information.

1. Begin at the CONVERSANT Administration menu, and select the following sequence:

   > Voice System Administration
   >Configuration Management
   >Equipment

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

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

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

Once you know which channel is 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 same steps for analog (T/R) and T1 channels described in Chapter 3, “Configuration Management,” of *Intuity CONVERSANT VIS V5.0 Operations*, 585-310-550.

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

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 additional information about integrating the PRI feature in your VIS application, refer to Chapter 6, “Using Optional Features with Script Builder,” of *Intuity CONVERSANT VIS V5.0 Script Builder*, 585-310-727.

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(‘O’)** instruction provides 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 additional information about integrating the PRI feature using TSM script language, refer to Chapter 3, “Script Instructions,” of *Intuity CONVERSANT VIS V5.0 Application Development*, 585-310-227.

Intuity 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. Refer to the *Intuity CONVERSANT VIS V5.0 IRAPI Programming Guide*, 585-310-226, for more information about these functions when developing IRAPI applications.

Adjunct/Switch Application Interface

4

What's in This Chapter

This chapter describes the use of the Intuity CONVERSANT Voice Information System (VIS) 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

ASAI provides an Integrated Services Digital Network (ISDN)-based interface between switches and adjunct processors. The CONVERSANT VIS ASAI feature supports this application interface for communications with the AT&T DEFINITY Communications System, Generic 3 (hereafter referred to as DEFINITY G3). This digital signaling interface allows the VIS to monitor and route calls on the DEFINITY G3. When used in conjunction with Tip/Ring (T/R) or digital Line Side T1 (LST1) interfaces, the ASAI interface allows the VIS to monitor and control the incoming calls it receives.

NOTE:
Various versions of DEFINITY G3 (such as G3i, G3r, etc.) are or will be certified with the ASAI feature. For the latest G3(x) versions certified for compliance with the ASAI feature, contact the AT&T Design Center.

Access to ASAI capabilities is provided through Script Builder. Refer to "ASAI Application Development" later in this chapter for additional information.

ASAI Connectivity

To support the ASAI capability on the VIS, the analog or digital connections between the VIS and PBX are provided through a point-to-point ISDN-Basic Rate Interface (BRI) via physical T/R or LST1 connections. One ASAI link per VIS is supported. The D-channel, which delivers control and supervisory messages about each T/R or LST1 channel, is connected to and processed by the ISDN Personal Computer Interface (IPCI) circuit card installed on the VIS.

The ISDN-BRI (Basic Rate Interface) line circuit card (TN556) must also be installed on the DEFINITY G3. For information on the TN556 circuit card, refer to the *DEFINITY Communications System Generic 1 and Generic 3i System Description,* 555-230-200, and the *DEFINITY Generic 1 and Generic 3i Wiring Manual,* 555-204-111.

Figure 4-1 shows a typical VIS and DEFINITY G3 configuration.

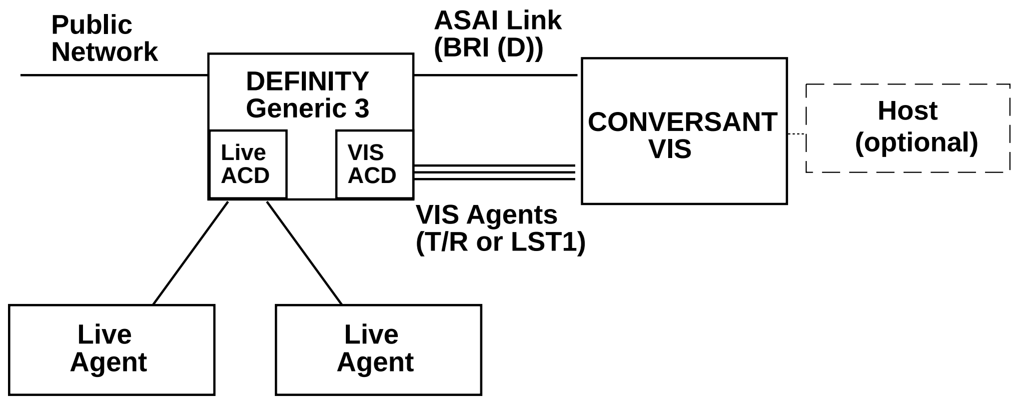


Figure 4-1. Typical VIS and DEFINITY G3 Configuration

NOTE:
The public network must provide an ISDN-PRI connection to the DEFINITY G3 for an application to receive calling number information.

Establishing the ASAI Link

The ASAI D-channel between the TN556 ISDN/BRI circuit card in the PBX and the IPCI circuit card in the VIS uses a separate connection. Figure 4-2 shows the D-channel connectivity.

You must connect an AT&T 440A4 8-pin terminating resistor (or equivalent) to the LINE connector of the IPCI circuit card using the DW8 cable provided. Use another DW8 cable to connect from the connecting block to the terminating resistor.

NOTE:
Total cable length from the DEFINITY G3 system to the VIS must not exceed 1900 feet. Refer to "ASAI Administration" below for additional information about administration of this communication arrangement.

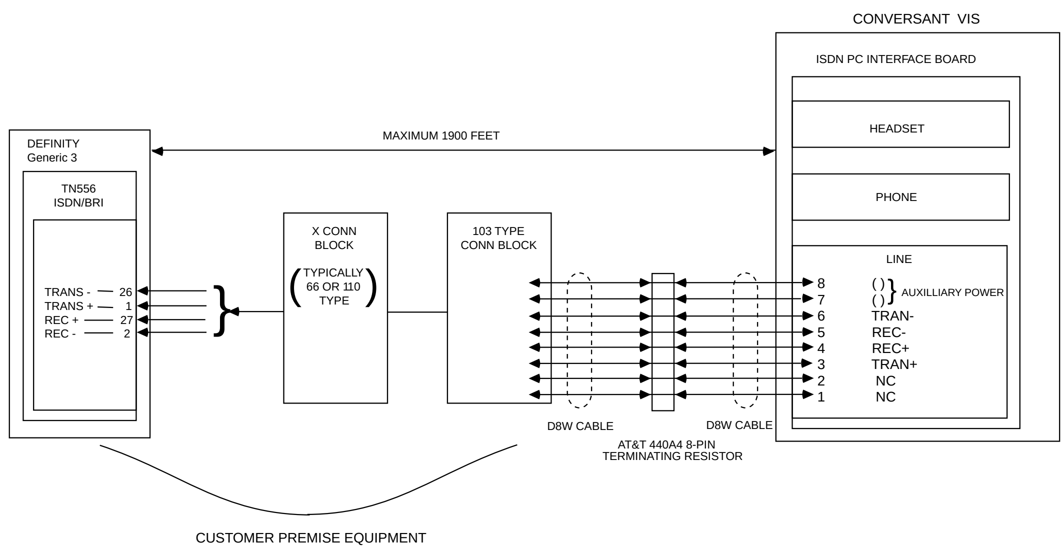


Figure 4-2. Typical D-Channel Wiring for an ASAI Link

Connecting the VIS Agents

The following information details making T/R and LST1 connections from the VIS to the PBX.

Analog T/R Connections

ASAI can be provisioned using analog T/R lines between the PBX and the VIS. Analog T/R circuit cards must be installed in the VIS and each line connected separately. Refer to the *Intuity CONVERSANT VIS V5.0 System Description*, 585-310-225, for information on T/R circuit card capabilities for ASAI.

Digital LST1 Connections

ASAI can also be provisioned with LST1, which allows digital T1 connections between the VIS and the line side of the PBX. This type of connection allows the utilization of various PBX features, such as call transfer and call progress tone (CPT) detection (in conjunction with Full CCA), which are not compatible with an ordinary T1 trunk connected between the VIS and PBX.

Analog configurations require 24 separate connections (four IVP6 or six IVP4 cards) to support an identical configuration provided by one LST1 cable. There is also a significant reduction in the number of VIS cards required to support the interface. One T1 and one SP circuit card support the same amount of traffic as four IVP6 or six IVP4 circuit cards, although the IPCI circuit card is still required.

ASAI Administration

Administering the ASAI feature is a four-step process. The following example assumes you are installing a voice response application with a configuration in which calls placed to an Automatic Call Distributor (ACD) on the PBX are directed to (agent) lines on the VIS. The VIS is used to select a service for the incoming call based on the dialed number identification service (DNIS), or called number. The service requests the DNIS number and automatic number identification (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 VIS:

1. Administer the BRI.
2. Administer the ACD domain (hunt group) on the VIS and the DEFINITY G3.
3. Administer the T/R or LST1 telephone lines.
4. Administer the VIS agent lines.

Once you have completed these steps, you can assign services to DNIS numbers. Refer to Chapter 3, “Configuration Management” of the *Intuity CONVERSANT VIS V5.0 Operations*, 585-310-550, for information on how to assign these services.

NOTE:
The following procedures assume you have installed the necessary hardware on the VIS and the DEFINITY G3. Refer to "ASAI Connectivity" earlier in this chapter and the hardware installation book for your platform for additional information.

NOTE:
The following procedures assume that you have completed the necessary administration on the PBX. Refer to the *DEFINITY Communications System Generic 3i Implementation*, 555-230-650, for additional information.

BRI Administration

With either a new VIS installation or a VIS upgrade, you must administer the DEFINITY BRI line to be used for ASAI connectivity between the DEFINITY and the VIS. Use the DEFINITY **add station** or **change station** commands to administer the BRI line. Use Table 4-1 for appropriate values.

Table 4-1. BRI Administration Field Name and Requirements

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

*To match the built-in administration of the IPCI circuit card and the ASAI software, the Type, XID, Fixed TEI, TEI, MIM Support, and CRV Length fields *must* have the contents indicated above.

†In previous releases, the CRV Length field required a value of 1. You *must* use the value 2 for Intuity CONVERSANT VIS V5.0.

Administering the ACD Split Domain

On the VIS

You must administer the ASAI feature to monitor the ACD hunt group extension and allow the VIS to receive information on calls placed to its VIS agent lines.In other words, you must administer the ASAI feature on the VIS so that it requests call events (information) from a *domain* on the PBX. In this case, the domain is the ACD hunt group or split, which is composed of the VIS agent lines. This domain is referred to as the *VIS ACD domain*. You can administer only one VIS ACD split domain on the system. Therefore, all VIS agent lines must be part of a single ACD split. Figure 4-1 shows this configuration. Refer to Chapter 4, “Feature Packages,” of *Intuity CONVERSANT VIS V5.0 Operations*, 585-310-550, to administer the VIS ACD Split Domain.

On the DEFINITY

Use the DEFINITY **add station** or **change station** command to administer the BRI line. Table 4-2 lists the values required for proper implementation of the DEFINITY for the ASAI link.

Table 4-2. DEFINITY Hunt Group Field Name and Values

Field Name	Contents
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
Controlling Adjunct:	none

Administering the T/R and LST1 Lines

Chapter 6, “Switch Interface Administration,” of *Intuity CONVERSANT VIS V5.0 Operations*, 585-310-550, describes how to administer the T/R and LST1. To be certain that you select options that are compatible with the DEFINITY G3 (only certain versions) system, select the “AT&T System 75” item in the PBX Defaults screen. AT&T System 75 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, refer to the information on changing maintenance state in Chapter 3, “Configuration Management,” of *Intuity CONVERSANT VIS V5.0 Operations*, 585-310-550. These lines or channels are referred to in the following text as *VIS Agent lines*.

CAUTION:
Do not proceed until the lines have been placed in the inserv state.

Administering the VIS Agent Lines

After creating and bringing the ACD split domain into service and administering the T/R and LST1 lines, you must administer and log in the lines as VIS agent lines. This is required if your service is going to use DNIS or the **A_Callinfo** or the **A_Tran** actions described in Chapter 6, “Using Optional Features with Script Builder,” of *Intuity CONVERSANT VIS V5.0 Script Builder*, 585-310-727. If you do not log in an agent line, the PBX 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 VIS by the ACD.) Refer to Chapter 4, “Feature Packages,” of *Intuity CONVERSANT VIS V5.0 Operations*, 585-310-550, for how to log in the VIS agent lines. Refer to Chapter 3, “Configuration Management,” of *Intuity CONVERSANT VIS V5.0 Operations*, 585-310-550, 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 VIS 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 may want to aggregate agents currently in multiple splits into a single split. This minimizes the number of domains that the VIS monitors and allows agents to be used more efficiently. Since DNIS is available in call events, you can have a single split of agents 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

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 have been integrated into Script Builder for use in ASAI applications.

NOTE:
The VIS 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 VIS ASAI feature to log T/R or LST1 channels in and out of an ACD split on the DEFINITY G3.

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

- Types of ASAI applications
- Using ASAI versus the Converse Vector Step (CVS)
- Using ASAI in a Call Center
- VIS 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 VIS. This implies that an Intuity CONVERSANT VIS ASAI system provides coresident voice response and DEFINITY G3-to-host gateway capabilities. A single call, for instance, can first be routed by the VIS, handled with a voice response application on the VIS, and then be monitored by the same VIS 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 VIS over T/R or LST1 channels via an ACD split on the DEFINITY G3. Figure 4-3 shows this class of application.

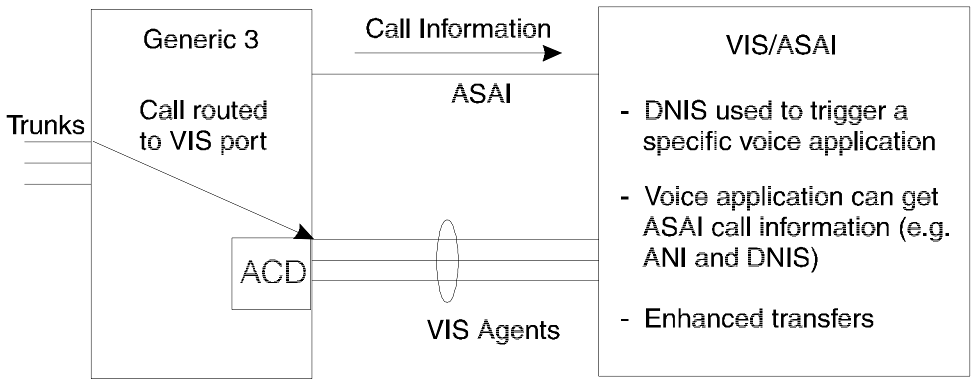


Figure 4-3. ASAI Voice Response Applications

As a call is delivered to the VIS, the VIS receives ASAI information related to the call through the D-channel connected to an IPCI circuit card in the Intuity CONVERSANT VIS. ASAI allows the VIS to receive the DNIS and/or ANI information of an incoming call to an analog T/R or digital 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 VIS 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 T/R 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 has been answered by the VIS, the ASAI information related to the call (such as ANI and DNIS) can be retrieved for use in the voice script 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 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. Refer to "Data Screen Delivery Applications" later in this chapter. 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, etc.
  - 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 could be used to determine where callers should be transferred if they need to speak to a live agent. This would be desirable if, for instance, agents handle calls from specific geographic regions.

Routing Applications

In routing applications using the ASAI feature, the VIS is used as a routing server to support the routing capabilities of ASAI and the call-vectoring feature on the DEFINITY G3. Figure 4-4 shows how a routing application on the VIS receives and responds to call-routing requests sent by the DEFINITY G3. The application uses routing information provided by the VIS to direct the call to a live agent or to a VIS agent via either a T/R or LST1 connection.

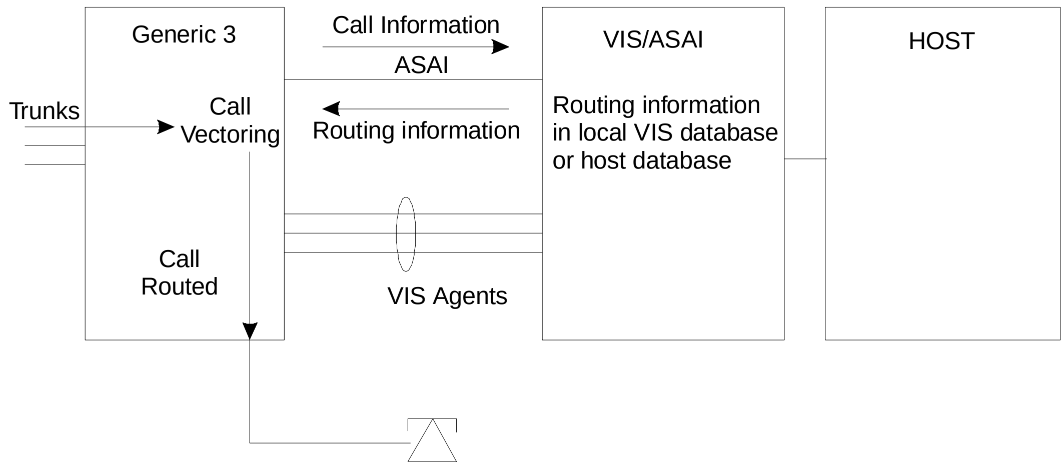


Figure 4-4. ASAI Routing Applications

These call-routing requests are generated by the DEFINITY G3 when a call is processed by specific call vectors on the DEFINITY G3.

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

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.
- Geographically-based service — Where service is provided on a regional basis, callers can be routed to the agent group 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 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 4-5 shows a data screen delivery application.

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

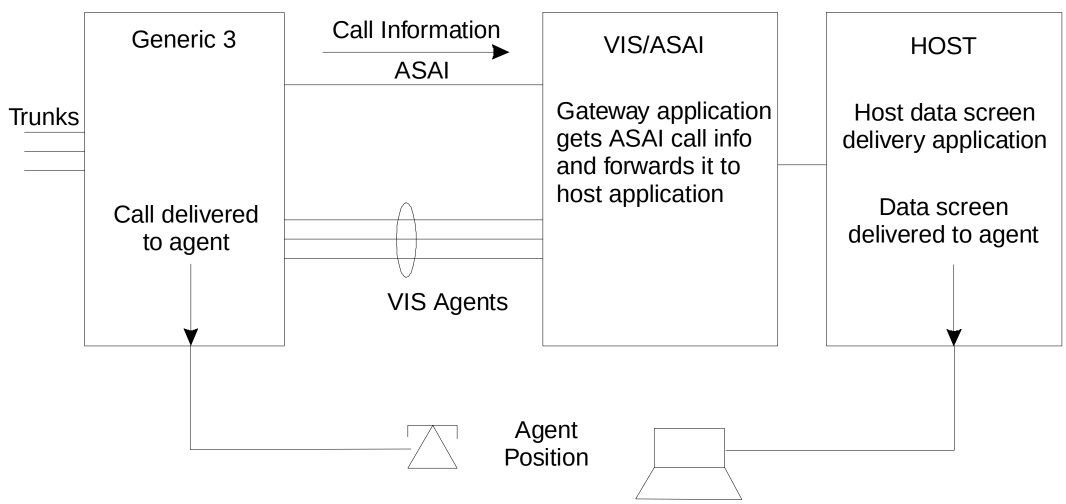


Figure 4-5. Data Screen Delivery Applications

Note that the delivery of data screens is not a function of the VIS itself. The VIS acts only as a communications gateway between the DEFINITY G3 and the host computer. A monitoring application on the VIS provides the ability to track the status of calls on the DEFINITY G3. 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 call-prompting information collected from the caller. In addition, the call may have been serviced by a VIS 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 VIS can therefore be used to support data screen delivery for three different call-flow scenarios:

- VIS-to-agent transfers — In this scenario, calls are delivered to the VIS 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.
- 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 VIS 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 call flow 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 system. Once a call is monitored by a particular VIS, the call cannot be redirected or transferred to a domain monitored by another VIS 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 VIS that monitors calls delivered to them. Also, for example, you may have VIS-to-agent transfers to support data screen delivery based on VIS collected data. In this case, you should configure multiple VIS systems to “front end” mutually exclusive sets of live agents. These considerations do not apply if you are using only one VIS ASAI system and it is the only ASAI adjunct.

The VIS-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 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, etc. 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 PBX-to-host gateway capabilities offered with the VIS 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 VIS 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 Converse Vector Step

The Converse Vector Step (CVS) allows the PBX to maintain control of a call while capabilities of the VIS 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 traffic volumes may require an ASAI-based solution due to the more efficient ASAI adjunct routing. Refer to Chapter 5, "Converse Vector Step Routing", for additional information about the CVS.

The following provides a list of the capabilities and limitations of using the two features on T/R or LST1 lines.

- Both ASAI and CVS provide the delivery of ANI, DNIS, and switch call prompting digits for T/R 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.

- ASAI and CVS cannot be used simultaneously for T/R or LST1 calls in an application. The splits identified by CVS must be vector controlled and cannot be monitored by using the VIS service available with ASAI. Conversely, the VIS ASAI A_Callinfo and A_Tran actions cannot be used in a script servicing calls delivered to the VIS via the CVS.

- The VIS 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 CVS have some unique properties that may influence the decision as to which feature to use:

- ASAI properties
  - When the VIS is used as a gateway for PBX-to-host applications, the **A_Tran** action simplifies call-flow development using screen pops based on VIS collected data.
  - 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 Intuity Response Application Programming Interface (IRAPI)-based start-up script written 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 VIS.
  - Queue position and administered digit string can be passed to the VIS using 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. This feature provides the following benefits:

- Enhanced customer service

  Caller- 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 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. This eliminates the need for callers to repeat information to each agent. For example, a caller may be directed initially to a VIS T/R 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 VIS 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/performance

  The coresidency of voice-response and PBX-to-host gateway applications with the VIS 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 VIS piece of the overall VIS/host application. In addition, the use of DNIS in voice-response applications to enable T/R or LST1 channel sharing means that the same number of VIS 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 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 may change when you add a VIS 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" in this chapter. 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 VIS 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 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 may 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.

VIS Script Design

The VIS ASAI feature provides four additional Script Builder actions that are used to access ASAI capabilities. These actions are discussed in detail in Chapter 6, "Using Optional Features with Script Builder," in *Intuity CONVERSANT VIS V5.0 Script Builder*, 585-310-727. A brief summary of these actions is provided below:

- **A_Callinfo** — This action is used within a voice response script to retrieve ASAI information about a call delivered to a T/R 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 VIS.
- **A_Event** — This action is used within routing scripts to receive information about call-routing requests sent by the DEFINITY G3 system. 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 VIS 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 T/R or LST1 channel on the VIS. 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, etc. "ASAI Application Examples" below includes 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 VIS T/R 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 VIS T/R or LST1 channel.

For ASAI voice response applications, incoming calls are routed to the VIS over T/R or LST1 channels configured as an ACD split on the DEFINITY G3 system. The VIS uses the Notification capability of ASAI to monitor this split. As a call is offered to this split, the VIS receives ASAI event reports indicating the status of the call (for example, call offered, queued, alerting, and connected event reports). The VIS 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 Vector Directory Number (VDN) on the DEFINITY G3 switch. Calls to these different VDN's can be routed to the same VIS 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 VIS allows the different dialed numbers to be associated with a specific voice response application. This allows T/R 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 VIS answers the call, the ASAI information related to the call can be retrieved for use in the voice script 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 VIS. Rather, special software is provided as part of the ASAI feature. This software processes the event reports and stores the information contained in these event 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 receiving 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 T/R 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 T/R 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 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 switchhook 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 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 VIS Data field of **A_Tran** is saved by the VIS even after the voice script terminates. The VIS 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 VIS. Since data collected in a voice script can be saved and is included in call events made available to the monitoring script, the host application is simplified. For instance, a CONNECT event (described later) made available to the monitoring script contains both the extension of the agent receiving the transferred call and the VIS data saved from the voice script which 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 system. In routing scenarios, calls are not physically delivered to T/R or LST1 channels on the VIS. Instead, incoming calls to the DEFINITY G3 are directed to a vector containing an *adjunct route* step. The adjunct route step causes a *route request* message to be sent to the VIS. The route request message contains information pertaining to the call (for example, ANI). The VIS uses this information to determine where to route the call.

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

Note that as a result of routing, the call may be directed to a VIS T/R 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 VIS 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 sent by the DEFINITY G3 system up to the script level. The **A_Event** action returns a ROUTE REQUEST event when the DEFINITY G3 system 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 system. 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 system. 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. These events can arrive at any point in time (controlled by vector processing on the DEFINITY G3 system). Hence, the primary difference between routing scripts and voice-response scripts is that once 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 VIS. Once the **A_Event** action retrieves a ROUTE REQUEST event, subsequent 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 VIS. This causes an ASAI route select message containing the routing information to be sent to the DEFINITY G3 system.

Steps 1 through 3 above 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 below in “ASAI Application Examples.”

A routing script may not contain any Script Builder actions that pertain to voice response capabilities (Announce, Prompt and Collect, etc.). A routing script is assigned by using the “RTE” domain designation as described in Chapter 4, “Feature Packages,” of *Intuity CONVERSANT VIS V5.0 Operations*, 585-310-550.

A routing script can use any of the information returned in the ROUTE REQUEST event. To route the call, refer Chapter 6, “Using Optional Features with Script Builder,” of *Intuity CONVERSANT VIS V5.0 Script Builder*, 585-310-727. 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 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 may 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 VIS 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 VIS 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 presented to agents receiving calls. Note, therefore, that the delivery of data screens is not a function of the VIS itself.

In data screen delivery applications, calls are not physically delivered to a T/R or LST1 channel on the VIS. Rather, calls are delivered to ACD agents on the DEFINITY G3 system. Note, however, that a call may have previously been delivered to a VIS T/R 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 6, “Using Optional Features with Script Builder,” of *Intuity CONVERSANT VIS V5.0 Script Builder*, 585-310-727. 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 the calls 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 system. 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 VIS. Once the **A_Event** action retrieves a call event, subsequent 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 above 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 below in “ASAI Application Examples.”

A monitoring script may not contain any Script Builder actions that pertain to voice-response capabilities (Announce, Prompt and Collect, etc.). To assign a monitoring script, use the “VDN”, “ACD”, or “CTL” domain designation as described in Chapter 4, “Feature Packages,” of *Intuity CONVERSANT VIS V5.0 Operations*, 585-310-550.

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 (DNIS, for example), calling party number (ANI, for example), and switch data (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 re-enable the domain. Refer to Chapter 4, “Feature Packages,” of *Intuity CONVERSANT VIS V5.0 Operations*, 585-310-550.
2. Stop and restart the voice system to activate the new script. Refer to Chapter 4, “Common Maintenance Procedures,” of *Intuity CONVERSANT VIS V5.0 Maintenance*, 585-310-153.

Call-Flow Scenarios

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

- VIS-to-agent transfers — In this scenario, calls are initially delivered to the VIS and then transferred from the VIS to a live agent. The transferred call can be monitored with a VDN 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 VIS Data field of **A_Tran** was used to save voice script data, this data is made available in the VIS Data field of call events sent to the monitoring script. Hence, 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, DNIS, and call-prompting information collected by the DEFINITY G3 system. Refer to "VIS-to-Agent Transfers" below for additional design considerations.
- Incoming call directly to agent — In this scenario, monitored VDN's or ACD splits deliver incoming trunk calls directly to live agents. Here, call events are passed to a VDN 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 collected in a voice script since a VIS voice script is not used to collect data from the caller. Since the VIS does not service calls in this scenario, no data is present in the VIS 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 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 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 or ACD type of monitoring script. Note that the call may first have been delivered to the VIS 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 available for the other two call-flow scenarios. ASAI-related information such as ANI, DNIS, and call-prompting information and VIS Data can be present in call events. This information can be used in conjunction with data entered by a live agent to provide the basis for data screens. Refer to "Agent-to-Agent Transfers" below for additional design considerations.

Call-Flow Design

VIS-to-Agent Transfers

VIS-to-agent transfers are accomplished by using the **A_Tran** action within a voice script servicing a caller. The use of **A_Tran** invokes ASAI Third Party Call Control operations to transfer a call away from the T/R or LST1 channel to which the caller is connected. The caller is transferred to the destination 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 VIS-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 monitored by the VIS with a monitoring script. Call events for the transferred call are passed to the script 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 Packages,” of *Intuity CONVERSANT VIS V5.0 Operations*, 585-310-550. In this case, the call can be transferred to nonmonitored domains and individual stations. Here, only call events for calls transferred from the VIS 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 VIS. Refer to Chapter 4, “Feature Packages,” of *Intuity CONVERSANT VIS V5.0 Operations*, 585-310-550, for the rules for which monitoring script receives call events when these two mechanisms are combined

In addition to monitoring the transferred call, the application developer can save data collected in the voice script for subsequent use in the data screen delivery application. To do this, use the VIS Data field of **A_Tran**. Any data saved in this field when the transfer is initiated from the voice script is presented in call events passed to the monitoring script that monitors the transferred call. The VIS Data field provides storage for twenty 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 VIS 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 VIS-to-Agent Transfers

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

1. A call arrives at a T/R or LST1 channel on the VIS. 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 the caller input is saved by using the VIS Data field of **A_Tran**. The voice script terminates after the transfer is complete and the T/R 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 VIS receives a CONNECT event for the call. The VIS Data field of this CONNECT event contains the account number 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 receiving 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 VIS Data information in addition to ASAI information related to the *original* call to the VIS. 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 originally delivered to the VIS T/R or LST1 channel. Call events for calls to T/R or LST1 channels on the VIS 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 pertinent to the original call. Refer to "Call Flow Examples" below for a detailed call flow example.

VIS-to-Agent Transfers Considerations

Additional considerations for VIS-to-agent transfers are as follows:

- In some cases, you may want to collect more data in a voice script than can be stored in the VIS 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 T/R 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 completed and the call terminates normally. As in the single transfer case, the END event contains information pertinent to the original call. Rules for how subsequent call events are reported are discussed below in “Agent-to-Agent Transfers.”
- The discussions on blind and consult transfers (refer to "Agent-to-Agent Transfers" below) do not apply to VIS-to-agent transfers completed using the **A_Tran** action. Also, the delay needed for agent-to-agent transfers discussed later does not apply to VIS-to-agent transfers completed using the **A_Tran** action.
- Transfers away from the VIS can still be accomplished by using standard flash transfer mechanisms. This type of transfer, however, precludes the ability to use the VIS Data field of **A_Tran** 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 (refer to "Agent-to-Agent Transfers" below). Hence, the discussions on blind versus consult transfer and the need to introduce delay for blind transfers from the VIS 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 system.

Call Monitoring in Transfer Scenarios

The VIS monitors VDN or ACD split domains by assigning monitoring scripts as described in Chapter 4, “Feature Packages,” of *Intuity CONVERSANT VIS V5.0 Operations*, 585-310-550. A call becomes monitored once it enters one of these monitored domains. The VIS *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 assigned to the domain the call has 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 Transfers Scenarios

The DEFINITY G3 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 VIS 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 completed as described in Step 4 below.
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 VIS is informed about the completed transfer immediately after the merge that occurs in Step 4. It is only after this merge, therefore, that the VIS 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 VIS can include information in the CONNECT event for the merged call which relates to the original call. In particular, the VIS retains the call ID of the original call and reports it in the Other Call ID of 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 phone when the second call is delivered to the second agent. Hence, for consult transfers, the VIS can only provide information 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 later.

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 VIS 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 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 calling customer.
3. Agent 1 transfers the call to Agent 2 by pressing the transfer button, dialing the VDN, 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. Refer to "Call Flow Examples" below 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 receiving the original call and any domains receiving 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 a 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 VIS assumes that a transfer to a station is taking place. Hence, two CONNECT events for the transferred call would be 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 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 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 VIS 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.

Refer to "Call Flow Examples" below 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 have been merged. This implies that the CONNECT event for the second call does not contain information 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 VIS 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 VIS, 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 pertinent to the first call. The Other Call ID field of this CONNECT event, for example, contains the call ID of the original call 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 VIS 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 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 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 VIS). 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 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 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 VIS ASAI system is integrated with a host computer. An application must be provided on the host that works with the VIS ASAI system. This host software application is not part of the VIS ASAI product. The host application can use the information it receives from the VIS 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 VIS 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 VIS 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 VIS 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 VIS ASAI system, ISDN services, and any additional communications system ACD features.

The voice response, routing, and data screen delivery applications enabled by a VIS 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 PBX 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 VIS T/R 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 may 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 passed by the DEFINITY G3 to the VIS.
- 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 PBX. 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, refer to "VIS-to-Agent Transfers" above 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 VIS 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 VIS 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 performing 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 VIS to drive the screens on agent's terminals. The interface to the VIS 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 has 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 have 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 VIS 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 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 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, 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 VIS 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 continues to deliver calls to agents even if the ASAI link to the VIS is down.

- If your call center application involves data screen delivery for VIS-to-agent transfers, refer to "VIS-to-Agent Transfers" above for additional design considerations.

- If your call center application involves data screen delivery for agent-to-agent transfers, refer to "Agent-to-Agent Transfers" above for additional design considerations.

- Your application should be able to accommodate cases where there are multiple CONNECT events received for the same call. This can occur, for example, in cases where direct agent calling is used. A call may 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 examples of scripts developed using the ASAI feature on the VIS. Included in this section is:

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

Sample ASAI Voice Script

The following is an example of an ASAI voice script 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
        # T/R 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 T/R 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 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 T/R 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 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 examples of data screen delivery call flows and the contents of the call events that result from these call flows. The following call flows are described:

- 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
- Call to ACD split that is redirected to AUDIX Voice Power
- 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
- VIS-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 have been selected for a call when the agent's telephone begins ringing or the agent hears a zip tone. CONNECT events may 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, then the incoming trunk group ID would be reported instead.

Also, since it is assumed that incoming calls are delivered via an ISDN facility, a 10-digit CPN is reported in call events. This number corresponds to the Called Party Number 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 dialed by the caller. The use of switch administration to modify DNIS digits does not affect the reporting of the called party number 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 would always be reported instead. Also, for non-ISDN calls, the called party number 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 called party number. 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 system 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 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 system 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 system 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 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 collected on the DEFINITY G3 system 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 associated with customer service. Calls to that 800 number are initially handled with a vector 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 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 system 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 associated with customer service. Calls to that 800 number are processed by a vector associated with VDN 7771. VDN 7771 is monitored.
2. The vector associated with VDN 7771 queues the call to a vector-controlled split with a “queue to” step.
3. The caller abandons 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 system 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 associated with customer service. Calls to that 800 number are processed by a vector associated with VDN 7771. VDN 7771 is monitored.
2. The vector 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 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 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.

Call to an ACD Split Redirected to AUDIX Voice Power

A call arrives at the DEFINITY G3 and is delivered directly to a monitored ACD split (no vector processing takes place for the call). The call is then redirected to and answered by AUDIX Voice Power. The caller then leaves a message and hangs up.

Example:

1. A caller calling from the telephone number 303-555-1726 calls a toll-free 800 number 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 redirected to the AUDIX Voice Power split and assigned to an AUDIX Voice Power port with the extension 5678.
4. A CONNECT event is passed to the monitoring script for ACD split 7777 with the following information:

Connected Party Number	5678
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	C

5. The caller leaves a message on AUDIX Voice Power and then hangs up.

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

Connected Party Number	5678
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

Note that in this scenario the CONNECT event identifies a connected party that the host application would not normally recognize. This is because the call is delivered to an AUDIX Voice Power port instead of a live agent.

This scenario illustrates just one of the many ways a call can be redirected to a second domain after having been offered to an initial domain. In these cases, call events for the redirected call are reported even though the call is not delivered to an endpoint in the targeted domain. The host application can accommodate these situations by filtering call events based on the connected party in addition to which monitoring script receives the events.

Notice that the CONNECT and END events are passed to the monitoring script for ACD split 7777 even though the call is redirected to the AUDIX Voice Power split. Until a call is presented to another monitored domain, the call continues to be monitored via the last monitored domain the call entered. This is a reflection on how ASAI software operates on the DEFINITY G3 system and shows how switch administration can influence how call events are 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 prior to when 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 associated with customer service. Calls to that 800 number are processed with a vector associated with VDN 7771. VDN 7771 is monitored.
2. The vector 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, 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 carried forward as well.

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 associated with customer service. Calls to that 800 number are processed with a vector associated with VDN 7771. VDN 7771 is monitored.
2. The vector 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 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 VIS 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 associated with the original call is carried forward as well.

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 associated with customer service. Calls to that 800 number are processed with a vector associated with VDN 7771. VDN 7771 is monitored.
2. The vector 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, 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 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 phone 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, VIS data, or switch data associated with the original call is carried forward as well and reported in call events 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 associated with customer service. Calls to that 800 number are processed with a vector associated with VDN 7771. VDN 7771 is monitored.
2. The vector 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 associated with a single, specialized agent.
7. The second 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 VIS 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 in a private conversation 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 phone 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, VIS data, or switch data associated with the original call is carried forward as well.

VIS-to-Agent Transfer via an ACD Split

A call is delivered to a VIS T/R 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 associated with customer service. Calls to that 800 number are processed with a vector associated with VDN 7771. VDN 7771 is not monitored.
2. The vector associated with VDN 7771 routes the call to the VIS T/R or LST1 split with a "route to" step. Note that this T/R or LST1 split is monitored but not by a monitoring script used to retrieve call events. This split is monitored internally by the VIS to support ASAI voice response applications making use of the **A_Callinfo** and **A_Tran** actions.
3. The call is answered by a T/R 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 VIS Data field of **A_Tran** is set to 987654321.
6. When the transfer is executed, the voice script terminates allowing the T/R 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 VIS-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 VIS Data field if such data was saved in the voice script via 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 associated with the original call is carried forward as well.

Note that the call may also be transferred from the VIS 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 monitoring script instead of a VDN- or ACD-split-type monitoring script. Also, A_Tran must be used to ensure that the CTL-type monitoring script receives the call events for the transferred call. Refer to Chapter 6, “Using Optional Features with Script Builder,” of *Intuity CONVERSANT VIS V5.0 Script Builder*, 585-310-727, for additional information.

Converse Vector Step Routing

5

What's in This Chapter

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 PBX to maintain control of a call while capabilities of the Intuity CONVERSANT Voice Information System (VIS) 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 (T/R) and Line Side T1 (LST1) lines.

Without the use of the CVS, once the call terminates on the VIS channel, it is no longer under the control of the switch. The VIS 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 touch tones to be passed to the VIS containing information such as the Calling Party Number/Billing Number (CPN/BN). At the end of the script, the VIS can pass back information relevant to further call vectoring via touch tones, such as a customer account number.

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

CVS Provisioning

The following information details the necessary provisioning for the CVS on the VIS and the PBX.

Provisioning within the VIS

The Converse Data Return (conv_data) action can only be implemented on the T/R and LST1 channels. Therefore, the application to be used must be assigned to the appropriate T/R and LST1 channels. Refer to the hardware installation guide for your platform for information about installing the necessary circuit cards. Refer to Chapter 3, “Configuration Management,” of *Intuity CONVERSANT VIS V5.0 Operations*, 585-310-550, for procedures on assigning service to channels.

The conv_data external action is a part of the base VIS V5.0 software. The Intuity CONVERSANT VIS base software must be installed prior to implementation of the CVS. Refer to Chapter 2, “Installing the Base System Software,” of *Intuity CONVERSANT VIS V5.0 Software Installation*, 585-310-151.

Provisioning within the PBX

The Converse Data Return (conv_data) action can only be used when the Intuity CONVERSANT VIS is used with the DEFINITY switch containing DS1 cards (version 767C 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 details the necessary administration of the CVS on the VIS and the PBX.

Administration within the VIS

The CVS return action executes a flash, and then transmits the digits contained in the Feature Access Code (FAC) and Data Return fields for conv_data. The duration of this flash must be set on VIS at 600 msec in the Switch Hook Flash Duration field. If you are using T/R 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. Refer to Chapter 6, “Switch Interface Administration,” of *Intuity CONVERSANT VIS V5.0 Operations*, 585-310-550.

The Dial Tone Delay in the Digital Protocol: Line Side T1 - DEFINITY screen must be set to 1000 msec.

NOTE:
The 1000 msec value may 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 may cause additional delays in the data return phase and the customer may hear dead air on the line.

Administration within the PBX

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

The Feature Access Code field in the conv_data action must match the corresponding FAC code setup on the switch. Refer to 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 VIS to send the collected data back to the switch

Script Builder

The Prompt and Collect action step and the conv_data external action are available to use the CVS within an application. Refer to Chapter 5, “Defining the Transaction,” of *Intuity CONVERSANT VIS V5.0 Script Builder*, 585-310-227.

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. Refer to Chapter 7, “Defining Parameters,” of *Intuity CONVERSANT VIS V5.0 Script Builder*, 585-310-227. This ensures that the Call Data Detail report (described in Chapter 5, “Reports,” of *Intuity CONVERSANT VIS V5.0 Operations*, 585-310-550) 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. Refer to Chapter 3, “Script Instructions,” and Appendix A, “Summary of Script Instructions,” in *Intuity CONVERSANT VIS V5.0 Application Development*, 585-310-227.

Intuity Response Application Programming Interface

It is possible to write an Intuity 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, refer to the *Intuity CONVERSANT VIS V5.0 IRAPI Programming Guide*, 585-310-226.

CVS Versus ASAI

For a discussion of the benefits of using the CVS over ASAI and vice versa, refer to Chapter 4, "Adjunct/Switch Application Interface".

CVS Examples

The following are typical ways in which VIS 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 equivalent to DNIS is available to the VIS via the first Prompt & Collect action. Based on the VDN, the VIS 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, T/R 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 via the second Prompt & Collection action. This information may be useful in a dealer or locator application.

- VIS announcement selection

  Hard-coded administered digit strings in the <data_1> and/or <data_2> fields can be used to instruct the VIS to play selected announcements.

- Expected delay announcement

  If the caller's position in the split queue is passed using the qpos keyword, the VIS can play an announcement informing the caller of the expected waiting time.

- ANI/routing

  Based on the CPN/BN, the VIS 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 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 processed by the VIS).

Call Classification Analysis

6

What's in This Chapter

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?

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 is needed to make call transfers and call bridges. It is provided with Intuity CONVERSANT Voice Information System (VIS) Version 5.0 software.
- Full — This type of call classification provides enhanced capabilities to intelligent call classification such as better answer detection, busy and audible ring tone detection, modem tone detection, etc. It is offered as an optional feature package with Version 5.0 software.

CCA Provisioning

Full CCA requires at least one AYC2C Signal Processor (SP) circuit card to be installed and operational prior to loading the Full CCA software. A single SP card can handle six simultaneous channels of CCA. This SP card must be dedicated to call classification (see "CCA Administration") and connected to the TDM bus. Refer to the appropriate hardware installation book for your platform for information on installing the SP circuit card.

Intelligent CCA on T1 or Primary Rate Interface (PRI) digital lines provides answer and disconnect supervision only. Intelligent CCA is not available on Line Side T1 (LST1) lines because there is no answer supervision or dial-tone detection. If detection of call progress tones (different from dial tone detection) with T1, LST1, or PRI lines is required, you must install Full CCA.

NOTE:
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 T/R lines instead of digital T1 or PRI lines. Analog lines tend to be more noisy than digital lines and may lead to occasional false identification of tones.

CCA Administration

You must assign CCA functionality to the SP circuit card for the CCA feature to operate properly. To assign CCA functionality to an SP circuit card, refer to Chapter 3, “Configuration Management,” of *Intuity CONVERSANT VIS V5.0 Operations*, 585-310-550, for the procedure to change the state of the SP circuit card.

CCA Application Development Issues

The following covers general, Script Builder, script language and Intuity 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 (six) are running. No further attempts to use Full CCA are made after the error is logged. Refer to system message TSM003 in Chapter 3, “System Message Listings,” of *Intuity CONVERSANT VIS V5.0 Maintenance*, 585-310-153.

Script Builder

Intelligent CCA or Full CCA can be activated when a call is dialed out during a 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 intelligent and Full CCA. Refer to Chapter 5, “Defining a Transaction,” of *Intuity CONVERSANT VIS V5.0 Script Builder*, 585-310-727, for additional information.

NOTE:
You must use the Make Call, Transfer Call, and Call Bridge actions to populate the database 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. Refer to Chapter 5, “Reports,” of *Intuity CONVERSANT VIS V5.0 Operations*, 585-310-550, for information about the Call Classification Report.

Script Language

The following instructions invoke Full CAA through script language:

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

The information below gives a brief discussion of these two instructions. For detailed information, refer to Chapter 3, “Script Instructions,” and Appendix A, “Summary of Script Instructions,” of *Intuity CONVERSANT VIS V5.0 Application Development*,
585-310-227.

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 showing 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)
```

Data Network Communications

7

What's in This Chapter

The following data network communication interfaces are available for use in conjunction with the Intuity CONVERSANT Voice Information System (VIS) V5.0 software.

- SNA 3270
- TCP/IP
- SQL*NET
- Asynchronous

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

Host Interface Software

The host interface is a combination of hardware and software 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 VIS to send and receive screens from applications running on the host mainframe. This offer was created in conjunction with CLEO Communications.

Host Interface Architecture Overview

The host interface software emulates an IBM 3274-41C or a 3174-01R cluster controller with up to 128 logical units (LUs) (that is, 3278 Model 2 terminals) connected to it. The host interface card (either a PC/XL, FIFO/SIB, or token ring circuit card) is typically linked to a 3725 or 3745 Front End Processor (FEP) and uses synchronous data link control (SDLC) or token ring data streams.

Hardware Environment Architecture

Figure 7-1, Figure 7-2, and Figure 7-3 illustrate typical VIS-to-host connections using modems, a modem eliminator, or a token ring.

Standard links from the host interface card to the FEP can be made through synchronous modems for distances over 100 feet and leased lines or modem eliminators for distances under 100 feet. The software supports speeds up to 56-Kbyte baud with the following restrictions:

- Certain line configurations are required to operate at speeds higher than 9600 baud. Refer to the information provided later in this section for assistance in modem configurations.
- High-speed connections for 56-Kbyte baud operation may use modems or modem eliminators with V.35 connectors. This requires an RS-232-to-V.35 interface converter since the PC/XL and FIFO/SIB circuit cards have only an RS-232 connector.
- If you have a PC/XL and want to support 56-Kbyte baud operations, you must have a Revision D or later version of the PC/XL card. To locate the revision letter of your card, look on the back side of the card near the RS232 connector. If your card is an earlier version than Revision D, you must upgrade to the FIFO/SIB circuit card.

NOTE:
The PC/XL circuit card is not an orderable replacement for CONVERSANT VIS V5.0. If a replacement SDLC card is needed, you must order the FIFO/SIB circuit card. Refer to the hardware installation book for your platform for additional information.

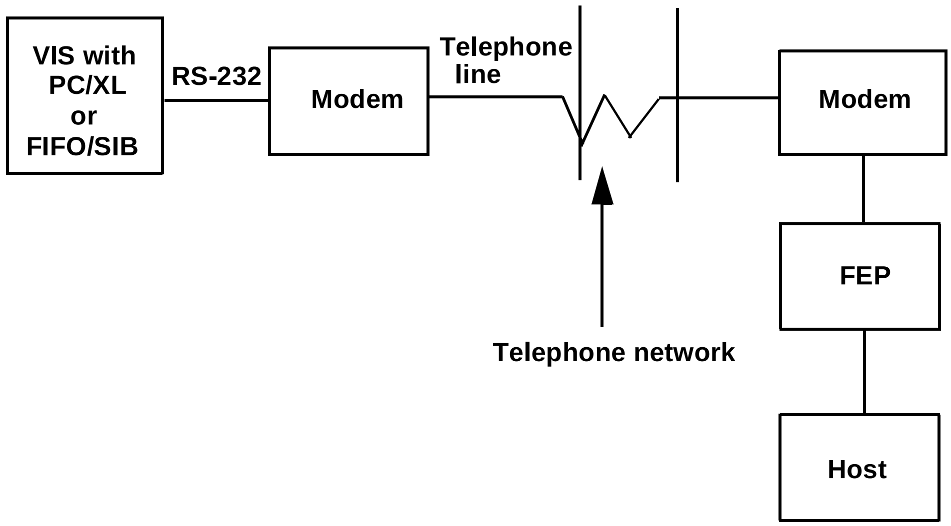


Figure 7-1. Sample Host Connection Using a Modem

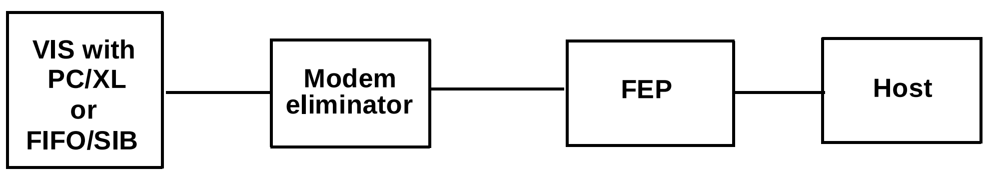


Figure 7-2. Sample Host Connection Using a Modem Eliminator

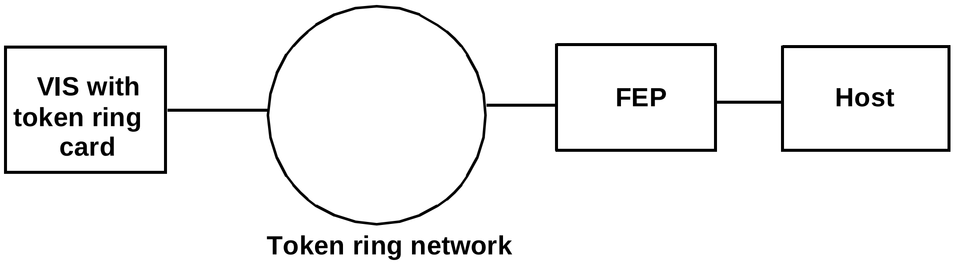


Figure 7-3. Sample Host Connection Using a Token Ring

The following list details the possible configurations for the host interface cards that are supported in a single platform for Intuity CONVERSANT V5.0. Note that a total of three host connections is only possible when using two FIFO/SIB cards and a Token/Ring card.

- 1 or 2 FIFO/SIB card(s)
- 1 PC/XL card
- 1 token ring card
- 1 or 2 FIFO/SIB card(s) and 1 token ring card
- 1 PC/XL card and 1 token ring card

Software Process Architecture

In Intuity CONVERSANT VIS Version 5.0 software, CLEO's LINKix product provides the Systems Network Architecture (SNA) software, including 3270, SNA and link levels. Two link-level protocols are supported:

- Token ring — A ring type of local area network (LAN) that allows any station in the network to communicate with any other station
- Synchronous Data Link Control (SDLC) — A line protocol that supports point-to-point communication

> **NOTE:**
> The token ring protocol has a higher throughput than the SDLC line protocol.

The following figure shows the current software process architecture for the host interface. Note that in Figure 7-4 the dashed line separates the process ownership between the VIS and LINKix.

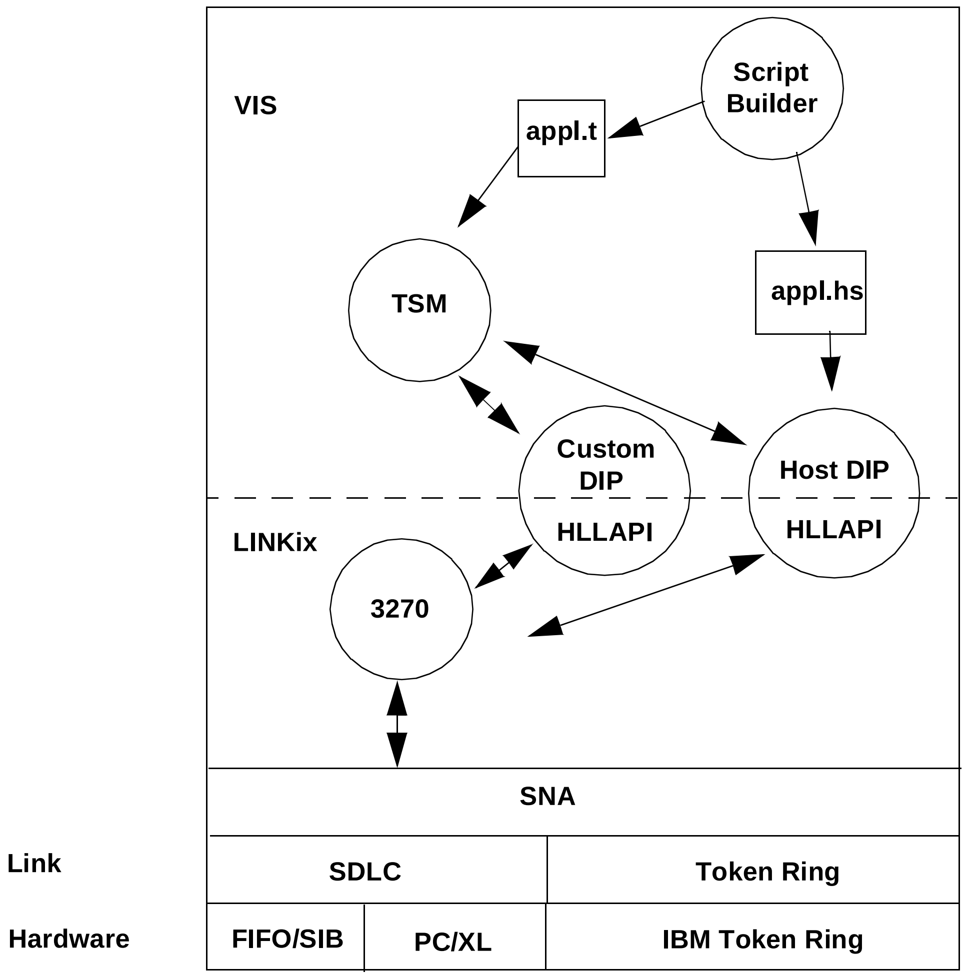


Figure 7-4. Host Interface Protocol

Host Interface Features

The following are the basic features of the Host Interface software available with Intuity CONVERSANT VIS V5.0 and CLEO Communications 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 on to the host computer and captures screen images, 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 3270 SNA or the High Level Language Application Programming Interface (HLLAPI),

- 3270 Terminal Emulation

  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 Intuity CONVERSANT VIS.

  NOTE:
  In past CONVERSANT VIS releases, this file transfer capability was a part of the 3270 Host Communication Package File Transfer System (FTS). It is now a part of the basic Host Interface software offer.

  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 as this is a change from previous product releases.

- Enhanced File Transfer

  Enhanced File Transfer uses the file transfer system to automatically transfer files between the VIS 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. Refer to the *HLLAPI Programmer's Guide* for additional information.

- NetView

  NetView provides the capability to send error messages to the host as Operator Generated Alerts (OGAs). LINKix also provides a more general NetView interface allowing messages to be sent to the host as other types of alerts. However, the VIS NetView package (see "Software Package Structure") does not use this more general interface.

Software Package Structure

The following packages comprise the Host Interface software for Version 5.0. Note that the Link, SNA, and Feature Levels are owned by CLEO Communications. Refer to the *Intuity CONVERSANT VIS V5.0 Software Installation*, 585-310-151, for information about the order in which you must install these packages.

- Link Levels — The link level package(s) you need depends on the type of protocol and the type of interface card(s) being used in the configuration.

  — linkix_coproc, Link Level (coproc). This package contains the CLEO LINKix PC/XL SDLC Link Level.

  — linkix_sib, Link Level (sib). This package contains the CLEO LINKix FIFO/SIB SDLC Link Level.

  — linkix_tkrn, Link Level (tkrn). This package contains the CLEO LINKix Token Ring Link Level.

  NOTE:
  The SDLC Link Level and Token Ring Link Level packages can be loaded and will operate simultaneously. However, the two SDLC packages (coproc and sib) *cannot* be loaded on the same system. The installation procedure prevents these packages from being loaded on the same system.

- SNA Levels — This package can be installed only *after* the link level package.

  — linkix_sna_128lu, SNA Level (sna128lu). This package provides support for 128 LUs.

- Feature Level 1 — The packages listed below, except for the NetView package (netman), are used in all SNA configurations. The NetView package is used only in an SNA configuration that uses NetView monitoring capabilities.
  - linkix_3270, Feature Level 1 (linkx3270). This package contains the CLEO LINKix 3270 Feature.
  - linkix_mgmt, Feature Level 1 (mgmt). This package contains the CLEO LINKix Management Utilities Feature.
  - linkix_netman, Feature Level 1 (netman). This package contains the CLEO LINKix Netview Package Feature.
- Feature Level 2 — This package can be installed only *after* the Feature Level 1 packages.
  - linkix_hte, Feature Level 2 (linkxHTE). This package contains the CLEO LINKix HLLAPI TE Feature.
- VIS Packages — These packages work in conjunction with the CLEO software.
  - Intuity CONVERSANT VIS V5.0 Synchronous Host Interface Package. This package includes basic voice system host communications software.
  - Intuity CONVERSANT VIS V5.0 Enhanced File Transfer. This package provides automatic transferring of files.
  - Intuity CONVERSANT VIS V5.0 3270 NetView Alarm Interface. This package provides the capability to send error messages to the host as OGAs. Refer to Chapter 8, "Data Network Connectivity Alarms", for additional information about the NetView Alarming.
- Package Updates

  Updates to the LINKix software may exist. When installing the software, be sure to install the package as well as any updates. Table 7-1 indicates the naming conventions for the update structure. Note that the package name is followed by a two digit number (*XX*). There will only be one update per LINKix package to be loaded at a time.

Table 7-1. LINKix Package Update Structure

Level	Package Name	Update Structure
Link Level	Token Ring	tkrn*XX*
	Coproc	coproc*XX*
	FIFO/SIB	sib*XX*
SNA Level	128 LUS SNA	sna128*XX* or sna*XX*
Feature Level 1	3270 Emulation	l3270s*XX*
	Advanced Netview	netman*XX*
	Management Utilities	mgmt*XX*
Feature Level 2	HLLAPI TE	linkxH*XX*

3270 Configuration Information

After the host is configured properly, you must set the VIS parameters to agree with the host's parameters. The following information details both SDLC and token ring configuration information, including the parameters that need to be set on the host and the VIS in the particular configuration.

SDLC Configuration Information

Host Sysgen Data for SDLC

The host sysgen data is the configuration information about the 3270 link on the host. The values of the following parameters in the host sysgen file are critical for the proper functioning of the 3270 software. Refer to "VIS Data for an SDLC Environment" for additional information concerning configuration values.

- DLOGMOD — Should be set to D4C32782 or the system default for the ISM 3278 Model 2 terminal
- DUPLEX — Can be either HALF or FULL. On multidrop lines (that is, when more than one terminal shares the line), use HALF duplex.
- MAXDATA — Determines the maximum path information unit for type2. The MAXDATA parameter should be less than or equal to 265.
- MAXOUT — Determines the maximum number of frames sent before the next polling. Set the MAXOUT parameter to 7.
- NRZ (Non Return to Zero) — Should be noted so that Encoding parameter on the VIS can be configured to match the host setting. Set this parameter to either NRZ (Non Return to Zero) or NRZI (Non Return to Zero Inverted).

- PU ADDR (Physical Unit Address) — Critical for host communication. This value is defined as a hexadecimal (that is, base 16) value. The PU address corresponds to the Poll Address parameter on the VIS.

  NOTE:
  In Version 5.0, the Poll Address *must* be a hexadecimal value. In previous releases, the VIS required the Poll Address to be a decimal value.

- PUTYPE — Sets the cluster controller type. Set the PUTYPE parameter to 2.
- SPEED — Can be any standard speed up to 56-Kbps baud that is supported by the attached modem or modem eliminator and the interface card. Make sure that it does not exceed the maximum speed of the modems or modem eliminators being used.
- TYPE — Can be either SWITCHED or LEASED. This value corresponds to the Line Type parameter on the VIS. It must match the setup for the modem or modem eliminator. Refer to the information provided later in this chapter for assistance in operating at speeds over 9600 baud.

VIS Data for an SDLC Environment

The SDLC configuration information is stored in a binary file called **/usr/lib/linkix/com.cfg**. The SDLC Protocol screen fields correspond to the configuration parameters described in Chapter 3, “Configuration Management,“ of *Intuity CONVERSANT VIS V5.0 Operations*, 585-310-550. Other configuration parameters are available through the LINKix command **comconfig**; however, they are rarely used and thus are not included in the following list. Refer to the *LINKix Administration Guide* for additional information on the **comconfig** command and other additional configuration parameters.

NOTE:
It is recommended that you use the screen available in the VIS menu to configure the SDLC environment. *Only* those with in-depth knowledge of SDLC connections should use the **comconfig** command.

- Connection Name — Specifies the SDLC connection. Valid values are SDLC*n*, where *n* is the card number. Default values are SDLC1 and SDLC2.
- Card Number — Indicates the card number to be used for this connection. Valid values are 1 (the default) and 2.
- Line Type — Specifies whether the connection is a leased or dial-up connection. Valid values are leased, switched manual dial, and switched autodial. Refer to the "Modem Configurations" information later in this chapter.
- Note Id to Send — Necessary only for host links that go through dial-up lines. The node identification is derived from the host system parameters.

- Encoding — Specifies the data link's encoding format. This parameter should match the settings in the host sysgen data for NRZ parameter. The valid values for encoding are nrz (Non Return to Zero) or nrzi (Non Return to Zero Inverted.)
- Constant Carrier — Specifies whether the modem supports constant carries. If this parameter is Yes, the voice system will keep the Request to Send (RTS) high.
- Poll Address — Specifies the cluster controller address. This value must be a hexadecimal value.

  NOTE:
  In Version 5.0, the Poll Address *must* be a hexadecimal value. In previous releases, the VIS required the Poll Address to be a decimal value.

- LU — Specifies which LUs should be defined as 3278 Model 2 terminals. The list of device numbers can range from 2 to 129. These numbers correspond to the LUs that are defined in the host sysgen data. They do *not* have to be consecutive numbers.

Modem Configurations

Certain line configurations must be present to use SDLC baud rates above 9600. Table 7-2 summarizes the affected configuration parameters.

Table 7-2. Modem Configuration Parameters

Constant Carrier	Line Type	Max. Baud	Comments
Yes	Leased	56 Kbyte	Ideal environment for the highest supported baud rates
No	Leased	56 Kbyte	Typical environment for a multi-drop configuration
No	Switched	9600 Kbyte	Not supported at line speeds above 9600 Kbyte

The configuration parameters Line Type and Constant Carrier must be set to reflect your modem environment. The three possible Line Type values are switched manual, switched autodial, and leased. The leased setting indicates that a line connection will always be present. The switched manual and switch autodial settings specify that one end must dial up the other end to establish for a line connection. Switched manual means a number must be manually dialed to make the connection. Switched autodial means that LINKix dials the number to make the connection if the modem allows it.

NOTE:
Defining the dial string for a switched autodial connection must be done through the LINKix utilities. It cannot be done through the Intuity CONVERSANT administrative screens.

The two possible Constant Carrier values are No and Yes. The No setting is used in most dial-up environments except when the modem is a V.32, V.22, or a V.42. The No setting must also be specified in multidrop environments. The Yes setting is used in single-drop, dedicated-line environments or when V.22, V.32, or V.42 modems are being used.

Table 7-3 summarizes the Constant Carrier and Line Type configuration parameter settings. The Request to Send (RTS)/Clear to Send (CTS) column indicates the action of the RTS modem signal. Toggled means that RTS is raised and lowered as required; Constant means that RTS is raised during protocol initialization and left high.

Table 7-3. Configuration Parameters

Constant Carrier	Line Type	RTS/CTS	Situation
No	Switched	Toggled	All dial-up modems *except* V.22, V.32 or V.42 which keep DCD constantly high
No	Leased	Toggled	Multidrop environments (not dialup)
Yes	Switched	Toggled	Dial-up environments using V.22, V.32, or V.42 modems which indicate line-connection via DCD
Yes	Leased	Constant	Single-drop, dedicated-line environments

Token Ring Configuration Information

Host Sysgen Data for a Token Ring Environment

The host sysgen data is the configuration information about the 3270 link on the host for a token ring environment. Refer to "VIS Data for a Token Ring Environment" for additional information.

- PU ADDR — An entry in this field must exist, but its value is not significant for the VIS.
- MAXDATA — This parameter determines the maximum size path information unit. With the default LINKix configuration, this should not exceed 1929. If LINKix and the UNIX kernel are tuned to allow a larger frame size, MAXDATA can grow to 4105.
- MAXOUT — This parameter determines the maximum number of frames sent before the next polling. Set the MAXOUT parameter to 7.
- PUTYPE — This parameter sets the cluster controller type. Set the PUTYPE parameter to 2.
- IDBLK, IDNUM — These values in combination must correspond to the “Node ID to Send” parameter on the VIS. See "VIS Data for a Token Ring Environment" for additional information on the Node ID to Send parameter.

VIS Data for a Token Ring Environment

NOTE:
It is recommended that you use the screens available in the VIS menu to configure the token ring environment. *Only* those with in-depth knowledge of token ring connections should use the **comconfig** command. Refer to “Adding Token Ring Protocol” in Chapter 3, “Configuration Management,” of *Intuity CONVERSANT VIS V5.0 Operations*, 585-310-550.

- Connection Name — Specifies the token ring connection. The default connection name is TKR*n,* where *n* is the first unused number starting from 1. (For example, if the current connection names are TKR1 and TKR3, TKR2 is the default.)
- Adapter Device name — Specifies the particular adapter used by this connection. The default is ibmtok_0.
- Local SAP Address — Specifies the Service Access Point (SAP) address the remote nodes use to contact the VIS. This is a 2-digit hexadecimal number ranging from 04 to EC. The value must be a multiple of 4. The default is 4.
- Remote SAP Address — Specifies the remote computer’s SAP address. This is a 2-digit hexadecimal number ranging from 04 to EC. The value must be a multiple of 4. The default is 4.
- Node ID to Send — Specifies the ID that is to be sent to the remote computer. This is an 8-digit hexadecimal number ranging from 00100001 to FFEFFFFF.

- Remote Network Address — Specifies the address of the host remote token ring node to which the VIS is connecting. This is a 12-digit hexadecimal number ranging from 000000000000 to FFFFFFFFFFFF. There is no default.
- LU — Specifies which LUs should be defined as 3278 Model 2 terminals. The list of device numbers can range from 2 to 129. These numbers correspond to the LUs that are defined in the host sysgen. They do *not* have to be consecutive numbers.

Administration Interfaces

The 3270 Synchronous Communications software can be administered from either the screen interface or the command line interface. This section details administration of the SDLC and token ring protocols from the command line. Many of the commands listed are also available from the VIS screen interface. For information on host administration via the screen interface, refer to Chapter 1, “User Interface,” and Chapter 3, “Configuration Management,” of *Intuity CONVERSANT VIS V5.0 Operations*, 585-310-550.

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 AT&T and CLEO Communications have developed commands to support the host interface software on the VIS.

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. VIS commands require host session numbers. LINKix commands require HLLAPI session IDs. The host session numbers range from 0–127. The HLLAPI session IDs range from 2–129 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 connection name and LU number to host session number is provided on the Display Host Sessions screen as described in Chapter 3, “Configuration Management,” of *Intuity CONVERSANT VIS V5.0 Operations*, 585-310-550. 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 SDLC 1 is configured with LUs 2–33 and SDLC 2 is configured with LUs 2–33, host session 0 equals HLLAPI session 2 which also equals LU2 on connection SDLC 1. Also, host session 32 equals HLLAPI session 34 which also equals LU2 on SDLC 2.

VIS Commands

For additional information concerning these commands, refer to the *Intuity CONVERSANT VIS V5.0 Command Reference*, 585-310-230.

- **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 0–127. If a session range is used, it can only include 10 sessions. If no session number is given, the command opens all available sessions 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 VIS 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 VIS development end. These discrepancies result in error messages being logged on the VIS 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.

  1. Turn on the modem or modem eliminator.
  2. 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. Refer to Appendix B, “Status Line Information,” of the *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 will display the HLLAPI session ID. This value equals the host session number plus 2.

3. If you have dial-up connections, connect with the host computer by dialing the telephone number of the host. If you have direct connections (leased-line), the host will probably identify itself soon after the communications card is loaded.

   You can now to send commands to the host.

sb_te executes the HLLAPI TE. (Refer to the *LINKix User's Guide* for additional information.) The Intuity CONVERSANT VIS V5.0 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 LINKix 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), this command shows what screen currently is being presented on that session. Make a note of this information; it will help 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 of the host before it goes down at night. It is a clean, convenient way to log out of 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.

- **hconfig**

  The **hconfig** command is the command interface you use to view and modify host configuration files. There are a number of options to the **hconfig** command. If any changes are made to the host configuration, you must restart the voice system for those changes to take effect.

- **hdiagnose conn <connection name>**

  The **hdiagnose conn** command determines which SDLC host interface card is installed (PC/XL or FIFO-SIB) and runs the appropriate diagnostic for that card and software. You must stop the voice system to run this command. If you do not stop the host interface process, **hdiagnose** runs **stop_hi** prior to running the diagnostics and then run **start_hi** after diagnostics are complete.

- **hdiagnose info <connection name>**

  The **hdiagnose info** command provides SDLC configuration information. You must stop the voice system to run this command. If you do not stop the host interface process, **hdiagnose** runs **stop_hi** prior to running the diagnostics and then run **start_hi** after diagnostics are complete.

- **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.*

LINKix Commands

For additional information about the following commands, refer to the LINKix documentation that accompanied your software. Specific references to the LINKix documentation for each command are provided.

- **comconfig** — Use this command to define complex configurations not supported through the screen interface provided on the VIS. Refer to Chapter 4, "Configuration Overview," of the *LINKix Administration Guide*.

  NOTE:
  It is recommended that you use the screen interface described in Chapter 3, "Configuration Management," of *Intuity CONVERSANT VIS V5.0 Operations*, 585-310-550, when configuring the host interface.

- **h3270** — Use this command to enter the HLLAPI TE. Invoking the **sb_te** command calls **h3270**. Refer to Appendix D, "HLLAPI Terminal Emulator," of the *3270 User's Guide*.

- **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. Refer to Chapter 4, "Transferring Files," of the *3270 User's Guide*. 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 as 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. Refer to Chapter 4, "Transferring Files," of the *3270 User's Guide*. 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 as this is a change from previous product releases.

- **comprintcfg** — Use this command to create a printable version of the configuration file. Refer to Chapter 4, "Configuration Overview," of the *LINKix Administration Guide*.
- **comservice** — Use this command to turn tracing on and off for LINKix processes. This command is used for SDLC or SNA protocol traces. Refer to Chapter 13, "Advanced Diagnostics," of the *LINKix Administration Guide.*
- **combrowse** — This command is a utility for viewing and filtering trace files.

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 VIS screens, use the File Transfer option provided via the Terminal Emulator selection in the Command Menu. For information on performing file transfer using either method, refer to Chapter 4, "Transferring Files," of the *3270 User's Guide*.

Interactive File Transfer

Refer to Chapter 3, "Controlling 3270 Emulation," of *3270 User's Guid*e 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:
You must be logged in to the host session and at the system ready prompt before you can execute the **comsend** and **comreceive** commands. You must use the **sb_te** command to establish the host session before using the file transfer command.

comsend

Use the **comsend** program to transfer a file from the VIS to the host mainframe; that is, to upload a file. 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 VIS 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. Refer to Table 7-4 for tips 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 provided in Chapter 4, “Transferring Files,” of the *3270 User’s Guide*. Note that some options are not available with all systems.

NOTE:
Mainframe systems vary in their requirements for the options list. Some mainframes 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 [* , ( ) etc.] must be preceded by a backslash (\) character in the **comsend** command line. Other characters may work, but the backslash is recommended in all cases.

comreceive

Use the **comreceive** program to transfer a file from the host mainframe to the VIS; that is, to download a file. 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 VIS 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. Refer to Table 7-4 for tips 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 Chapter 4, "Transferring Files," of the *3270 User's Guide*. Note that some options are not available with all systems and may not all be available with all systems.

NOTE:
Mainframe systems vary in their requirements for the options list. Some mainframes 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 [* , ( ) etc.] must be preceded by a backslash (\) character in the **comreceive** command line. Other characters may 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 get rid of the character in the file. This is typically a problem with TSO and VM systems.

When a binary file is received from the host, zeros (0) 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 7-4. Filename Guidelines for File Transfer

If Filename Contains	UNIX			Host3270		
	Syntax	Examples		Syntax	Examples	
		Original	Converted		Original	Converted
| & ; < > () ‘ \ ‘ * ? [] # ~ †	Precede each special character with a backslash (\)	ix’yy’a\bc	x\’yy’\a\\bc	Precede each special character with a backslash (\)	#AB~C*D E?cde#f*h	\#AB~C*D E\?cde#f*h
dollar sign ($)	Precede $ with backslash (\)	AB$tmp	AB\$tmp	Precede $ with backslash (\) ‡	XXyy$zz	XXyy\$zz
at sign (@)	Precede @ with backslash (\)	AB@tmp	AB\@tmp	Precede @ with backslash (\) §	XXyy@zz	XXyy\@zz
period (.)	No special syntax	s.xx.c	s.xx.c	Enclose filename first with a backslash (\) followed by an apostro-phe (‘) \’ ††	s.xx.c	\’.xx.c\’
Any char-acter not shown above	No special syntax	abcd	abcd	No special syntax	a123bcd	a123bcd

† Protect # and ~ with a backslash only if they begin the filename

‡ Protect $ with backslash only when the file transfer is done directly with the **comsend** or **comreceive** commands. Do not protect $ when file transfer is done through the 3270 terminal emulator.

§ Protect @ with backslash only when the file transfer is done directly with the **hsend**, **comsend** or **comreceive** commands. Do not protect @ when file transfer is done through the 3270 terminal emulator.

†† Protect . only if transferring files to/from a tso system and the dots in the filename are a fully qualified filename (containing the user id).

§§ You may *not* use an underscore when specifying a filename.

Administering Enhanced File Transfer

Local VIS Procedures

The user at the local VIS should do the following:

1. Develop, verify, and install a host maintenance script that initiates and maintains a host session; that is, provides procedures for login, logout, and recovery screen sequences. Note that the script should leave the host session at the host system ready prompt that will allow an interface with the host IND$FILE file transfer program. Refer to the *Intuity CONVERSANT VIS V5.0 Script Builder*, 585-310-727, for information on developing, verifying, and installing a host maintenance script.

   NOTE:
   After a file transfer, the host system-ready prompt may be in a different position on the screen. The recovery and logout sequences must take this into consideration. The user may need to define multiple screens for the host system-ready prompt.

2. Begin the file transfer by executing the **hassign** command to assign the host maintenance script to the host session. Following is the format of the **hassign** command:

   hassign <*application*> to <*session*> FTSCRT

   application is a required argument that specifies the host maintenance script for file transfer. *session* is a required argument that specifies the session number or a range of session numbers. *FTSCRT* is a required argument that assigns the session for file transfer. Refer to the *Intuity CONVERSANT VIS V5.0 Command Reference*, 585-310-230, for information on using the **hassign** command.

3. Execute the **hstatus** command to verify that the session is logged in to the proper screen for file transfer. If the session is logged in properly, **hstatus** displays “file transfer” as the session’s status. Refer to the *Intuity CONVERSANT VIS V5.0 Command Reference*, 585-310-230, for additional information on using the **hstatus** command.

4. If you are preparing to transfer a Script Builder application script to the remote site via the host, you must develop, verify, and install this application script using Script Builder. Refer to the *Intuity CONVERSANT VIS V5.0 Script Builder*, 585-310-727, for information on developing, verifying, and installing Script Builder applications.

5. If you are preparing to transfer a Script Builder application script to the remote site via the host, create a batch file to remove existing applications and install the new application script developed in the previous step. This batch file is sent with the application script to the remote VIS via the host. Once the remote VIS receives the batch file, it executes the commands in the batch file. The batch file can be any combination of regular UNIX commands, executable shell files, and executable program names.

For example, to automatically install an application received from the host, the batch file can execute the **remove_appl**, **restore_appl**, and **install_appl** commands. Note that the name of the batch file should end with **.vb**. Procedures and suggestions for creating batch files are described in detail later in this chapter under "Batch Files used in the Enhanced File Transfer System". Refer to the *Intuity CONVERSANT VIS V5.0 Command Reference*, 585-310-230, for information on the **remove_appl**, **restore_appl**, and **install_appl** commands.

6. If you are preparing to transfer a Script Builder application, execute the **backup_appl** command to create one file each for the transaction, speech, and database portion of the transaction. Next, bundle the Script Builder transaction, speech, and database files and the batch file into one bundle using the UNIX **cpio** command. If you are preparing to transfer a software package, bundle the software package and the batch file into one bundle by using the **cpio** command. Refer to the *Intuity CONVERSANT VIS V5.0 Command Reference*, 585-310-230, for information on the **backup_appl** command. Refer to the *UNIX System V Release 4.2 Command Reference (a–l)* for information on using the UNIX **cpio** command.

7. Name the file to be sent to the remote VISs and if necessary modify the DESTINATION parameter in the configuration file (**/vs/data/fts_config**) on the local VIS machine to include this filename. The DESTINATION parameter specifies the name of the bundle on the host 3270 mainframe. The DESTINATION parameter is required and must be set either in the configuration file or on the **hsend** command line. Refer to the *Intuity CONVERSANT VIS V5.0 Command Reference*, 585-310-230, for information on using the **hsend** command.

8. Send the file to the host by executing the **hsend** command. The format of the **hsend** command is as follows:

   hsend file=<filename> ***[dest=] [opt=]***

   <filename> is a required argument that specifies the full path name of the UNIX file or cpio bundle to be sent to the host. Refer to Figure 7-4 for filename guidelines for file transfers. *dest* is an optional argument that specifies the final destination of the file at the host. If a destination is not specified, the DESTINATION parameter from the **/vs/data/fts_config** file is used as the destination. *opt* is an optional argument that specifies either a list of options or the letter n (for no options). Note that the options must be separated by a space. If an option list is provided, it is sent to the host. If the option argument value is *n*, the PARAM1, PARAM2, and PARAM3 parameter values are not appended to the host IND$FILE file transfer program. If this argument is missing, the PARAM1, PARAM2, and PARAM3 parameter values are used.

The local VIS is now ready to send files to the remote VIS via the host and/or receive files sent from the remote VIS via the host. The procedures for sending files from the host to the remote VIS and sending files from the host to the local VIS are discussed later in this chapter.

Remote VIS Procedures

The user at the remote VIS should do the following:

1. Develop, verify, and install a host maintenance script that initiates and maintains a host session; that is, provides procedures for login, logout, and recovery screen sequences. Note that the script should leave the host session at the host system ready prompt that will allow an interface with the host IND$FILE file transfer program. Refer to the *Intuity CONVERSANT VIS V5.0 Script Builder*, 585-310-727, for information on developing, verifying, and installing a host maintenance script.

   NOTE:
   After a file transfer, the host system-ready prompt may be in a different position on the screen. The recovery and logout sequences must take this into consideration. The user may need to define multiple screens for the host system-ready prompt.

2. Execute the **hassign** command to assign the host maintenance script to the host session. Following is the format of the **hassign** command:

   hassign *<application>* to *<session>* FTSCRT

   application is a required argument that specifies the host maintenance script for file transfer. *session* is a required argument that specifies the session number or a range of session numbers. *FTSCRT* is a required argument that assigns the session for file transfer. Refer to *Intuity CONVERSANT VIS V5.0 Command Reference*, 585-310-230, for information on using the **hassign** command.

3. Execute the **hstatus** command to verify that the session is logged in to the proper screen for file transfer. If the session is logged in properly, **hstatus** displays "file transfer" as the session's status. Refer to *Intuity CONVERSANT VIS V5.0 Command Reference*, 585-310-230, for additional information on using the **hstatus** command.

4. Modify the **/vs/data/fts_config** configuration file on the remote VIS to poll the local VIS for the file. Procedures for modifying the **/vs/data/fts_config** file are described later in this chapter.

5. If necessary, create the APPL_FTS utility to preprocess the bundle received from the host. Preprocessing is optional and may be used to customize the file transfer feature by adding header information, special files, etc. to the files that will be handled by the Enhanced File Transfer system. Note that the full path name of the preprocessing file should be added to the APPL_FTS field in the configuration file.

The remote VIS is now ready to receive files sent from the local VIS to the host and/or send files to the host and the local VIS. The procedures for sending files from the host to the local VIS and from the host to the remote VIS are discussed later in this chapter.

Receiving Files from the Host on the Remote VIS

The Enhanced File Transfer system automatically transfers files from the host to the remote VIS. This procedure is part of this automatic transfer:

1. Poll the host at a time determined by the **/vs/data/fts_config** configuration file (POLL_START, POLL_FREQ, and POLL_END fields).
2. Receive a bundle sent by the local VIS.
3. Place the bundle in a temporary directory (for example, fts_tmp1) under the directory specified in the FROM_HOST_DIR field in the **/vs/data/fts_config** file. By default, each temporary directory is created under the **/usr/tmp** default directory.
4. Create a log file with the full path name of the bundle as specified in the FROM_HOST_DIR field in the **/vs/data/fts_config** file. All batch file command outputs are appended to the log file, with each record in the log file containing the original command line and the command output.
5. Execute either the APPL_FTS file (if one exists) or the UNIX **cpio** command (if an APPL_FTS file does not exist) to preprocess the bundle received from the host.
6. After preprocessing is complete, execute the batch file received from the host under the temporary directory. Note that the batch file name must end with .vb and must conform to UNIX standards.

   WARNING:
   If more than one batch file is sent in a bundle, the transfer is treated as an error and no further processing will take place for that bundle.

7. Record the status of all Enhanced File Transfer activities in the log file.
8. After executing the batch file, the Enhanced File Transfer system sends the log file to the host. At this time, the user may execute the **hsend** command to send other files to the host, including output files created during the execution of commands within the batch files. Refer to the information on “Sending Files to the Host” later in this chapter for information on this procedure.
9. Set the next polling time.

Sending Files from the Remote VIS to the Host

Use the **hsend** command to send files other than the log file from the remote VIS to the host. These files can include output files created during the execution of commands within the batch files. The format of the hsend command is as follows:

hsend file=<*filename*> *[dest=][opt=]*

<filename> is a required argument that specifies the full pathname of the UNIX file or cpio bundle to be sent to the host. Refer to Table 7-4 for filename guidelines for file transfers. *dest* is an optional argument that specifies the final destination of the file at the host. If you do not specify a destination, the DESTINATION parameter from the **/vs/data/fts_config** file is used. *opt* is an optional argument that specifies either a list of options or the letter "n" (for no options). Note that you must separate the options by a space. If an option list is provided, it is sent to the host. If the option argument value is *n*, the PARAM1, PARAM2, and PARAM3 parameter values are not appended to the host IND$FILE file transfer program. If this argument is missing, the PARAM1, PARAM2, and PARAM3 parameter values are used.

NOTE:
The Enhanced File Transfer system removes the log file on the remote VIS after the file is successfully transferred to the host. If the log file is not sent to the host successfully, it is stored at the FROM_HOST_DIR directory and renamed ***[unix time]*.log** where *[unix time]* indicates the current system time in seconds. It is the user's responsibility to remove the stored log file later to save disk space.

Batch Files used in the Enhanced File Transfer System

UNIX commands have two output files, stdout and stderr. Conventionally, stdout is used for expected output (often none) and stderr is used for error messages. You can discard the output of either the stdout, stderr, or both by directing it to **/dev/null**. Generally, a command line in a batch file behaves the same way as a command typed at a terminal; that is, the following occurs:

- Undirected stderr and stdout are collected and appended to the host log
- If stdout is redirected to **/dev/null**, the output is not appended to the host log (for example, **install_sw xmas_sale > /dev/null**)
- If stderr is redirected to **/dev/null**, the output is not appended to the host log (for example, **install_sw xmas_sale 2 > /dev/null**)
- If both stderr and stdout are redirected to **/dev/null**, nothing regarding the command is written to the host log (for example, **install_sw xmas_sale > /dev/null 2 > &1**)

All batch file command outputs are appended to the log file that is created in the FROM_HOST_DIR. Each record in the log file contains the original command line and the command output.

NOTE:
Do not include commands that are inherently interactive or that do not terminate automatically in batch files. Commands that are inherently interactive are difficult to execute on a noninteractive basis unless all the required responses are known in advance. Commands that do not terminate automatically can also cause a problem.

Configuring fts_config File for Enhanced File Transfer

The Enhanced File Transfer configuration file contains field settings that are used in configuring the IND$FILE file transfer program on the host.

Configuration information is stored in an ASCII file called **/vs/data/fts_config**. Use the following procedure to view and edit the contents of this file:

1. Log in as **root**.
2. Enter **vi /vs/data/fts_config**

   The default value for parameters in **/vs/data/fts_config** are shown in Figure 7-5.

```
POLL_START=-01:00
POLL_FREQ=04:00
POLL_END=24:00
DESTINATION=
ORIGINATION=
APPL_FTS=
HOST_OS=TSO
FROM_HOST_DIR=/usr/tmp
PARAM1=
PARAM2=
PARAM3=
Verbose=1
Max_receive=1
```

Figure 7-5. /vs/data/fts_config Example

Following is a description of each field in the **/vs/data/fts_config** file.

POLL_START

The POLL_START field specifies the time of day at which the Enhanced File Transfer system first polls the host. The default value is -01:00. This specifies that the Enhanced File Transfer system never polls the host but sends files only when a request is made. If you change the POLL_START value from the default (-01:00) to any value between 00:00 to 24:00, the Enhanced File Transfer system uses the new POLL_START value following the next polling period or the next **hsend** command.

NOTE:
You may not set the POLL_START field to a value greater than 24 hours (24:00). If you attempt to set the POLL_START field to a value greater than 24 hours, the value (00:00) is used.

POLL_FREQ

The POLL_FREQ field specifies the intervals at which the Enhanced File Transfer system polls the host. The default value is 04:00. This specifies that polling will occur every 4 hours. If you set the POLL_FREQ field to a value less than or equal to 00:00, the Enhanced File Transfer system polls only at POLL_START. For example, if the POLL_FREQ field is set to -01:00 and the POLL_START is set to 01:00, the Enhanced File Transfer system polls the host starting at 01:00. If you set the POLL_FREQ field to a value greater than 24 hours, the Enhanced File Transfer system polls the host at this offset from POLL_START. For example, if you set the POLL_START to 02:30 and POLL_FREQ to 50 hours, the Enhanced File Transfer system polls the host at 4:30 a.m. on alternate days. If you change the POLL_FREQ field just after the most recent POLL_START, the Enhanced File Transfer system changes the POLL_FREQ at the next POLL_START or the next execution of the **hsend** command. For example, if POLL_FREQ is changed from 01:00 to 00:30 at 2:20 p.m., the POLL_FREQ does not change until the next polling period begins at 3:00 p.m. or until the **hsend** command is executed.

POLL_END

The POLL_END field indicates the time of day after which the Enhanced File Transfer system will not poll the host. The default value is 24:00.

NOTE:
You may not set the POLL_END field to a value less than or equal to 00:00 or greater than or equal to 24:00. If you attempt to set POLL_END in this manner, the default value (-01:00) is used. This default value indicates that the POLL_END field should be ignored. Only the POLL_START field is used to determine whether to begin polling.

DESTINATION

The DESTINATION is a required field that specifies a dataset (file) name that is acceptable to the host. The DESTINATION specified in this field is used as the destination argument to the **hsend** command for sending a bundle to the host.

ORIGINATION

The ORIGINATION is a required field that indicates a dataset (file) name that is acceptable to the host. The ORIGINATION specified in this field is used as the origination argument to the **comreceive** command for receiving a bundle from the host.

APPL_FTS

The APPL_FTS field is used only if a program has been created to preprocess the bundle received from the host. The APPL_FTS field specifies the full path name of this program. The APPL_FTS default value is NULL. This default value indicates that a preprocessing program does not exist.

HOST_OS

The HOST_OS is a required field that indicates the name of a host application. You may specify either CICS, TSO, or CMS in this field. The HOST_OS default value is TSO.

FROM_HOST_DIR

The FROM_HOST_DIR field specifies the full pathname of the directory on the VIS where the Enhanced File Transfer system creates a temporary directory to receive a bundle from the host and executes the batch file from each of these temporary directories. The FROM_HOST_DIR default value is **/usr/tmp**.

PARAM1, PARAM2, PARAM3

PARAM1, PARAM2, PARAM3 are optional fields that are reserved for any additional parameters. Note that the parameters are sent in order of PARAM1, PARAM2, and PARAM3 with a space in between them (for example, PARAM1 PARAM3). Refer to Chapter 4, “Transferring Files,” of the *3270 User’s Guide* for a list of file transfer options.

Verbose

The Verbose field indicates the level of detail of the **/tmp/fts_trace** file. A Verbose setting of 1 (the default) indicates the most detailed level. This file is used for debugging purposes. A Verbose setting of -1 instructs the VIS to turn off tracing.

Max_receive

The Max_receive field specifies how many times the VIS attempts to receive a bundle from the host during each polling cycle. The Max_receive default value is 1. A Max_receive value of -1 specifies that the VIS will never poll the host.

Changes in the configuration file take effect the next time the host is polled. To make the changes take effect immediately, perform the “Stopping the Voice System” and “Starting the Voice System” procedures in Chapter 4, “Common Maintenance Procedures,” of *Intuity CONVERSANT VIS V5.0 Maintenance*, 585-310-153. You can also cause changes to take effect by using the **hsend** command. Refer to the information on sending files to the host in this chapter for additional information on using the **hsend** command.

Examples of Enhanced File Transfer

Sending a Single ASCII File to the Host

Enter:

hsend file=<*filename*> *[dest=filename on the host] [opt=ASCII CRLF]*

NOTE:
The above example assumes that PARAM1 and PARAM2 are set to “ASCII” and “CRLF” and DESTINATION is set to the host dataset name. If these values are not set, the *<dest>* and *<opt>* fields are not optional.

Receiving a Single ASCII File from the Host

1. Make sure that polling is on.
2. Create the file **/usr/tmp/appl** with the following contents, where */usr/tmp/hostfile* is the file received from the host:

   cp /usr/tmp/fts_tmp1/tmp1.pkg /usr/tmp/hostfile

3. Enter **vi /vs/data/fts_config**
   a. Change the APPL_FTS parameter to **/usr/tmp/appl.**
   b. Change the FROM_HOST_DIR parameter to **/usr/tmp.**
   c. Change the PARAM1 parameter to ASCII and the PARAM2 parameter to CRLF.
   d. Change the ORIGINATION parameter to the filename on the host.

Receiving a Package from the Host

Make sure that polling is on and modify the **/vs/data/fts_config file** as follows:

1. Keep the APPL_FTS parameter blank.
2. Change the FROM_HOST_DIR parameter to **/usr/tmp.**
3. Change the ORIGINATION parameter to the destination file name used in the **hsend** command.
4. Change the DESTINATION parameter to a desired host file name for later use. The Enhanced File Transfer system will use this file name in sending the trace log from the **tmp.vb** execution back to the host.

Sending/Receiving an Application

The following demonstrates the steps necessary to test sending an application to a host, and then receiving that same application back through the use of Enhanced File Transfer:

1. Enter **backup_appl -n *<appl_name>***

   This creates binary files for each component of an application, which include Transaction (Trans), Speech (Spch), and Database (Dbase).

2. Enter **cd /tmp/sb/BkUpAppl/*<appl_name>***

   This is the directory to which the Trans, Spch, and Dbase files are copied.

3. Enter **vi *<filename>*.vb**

   This is the file that will be run when it is received on the target system.

4. Enter **ls |cpio -oBcv > *<all_files_name>***

   This creates the <*all_files_name*> bundle that contains all the files together and will be sent using **hsend.**

5. Enter **vi /vs/data/fts_config**

   a. Update the Destination parameter with the name you want this application to be stored under on the host system. Remember, it must conform to the host file naming rules and special characters should be preceded with a backslash.

   b. Update the POLL_START with a positive value that you want to use to poll the host.

   c. Make sure that PARAM1, PARAM2, and PARAM3 are set to blank.

6. Enter **hassign *<eft_appl>* to *<session number>* FTSCRT**

   This assigns the Enhanced File Transfer script to a session and gets a session to the READY prompt, ready for a file transfer. To ensure that the session is ready, enter **hstatus *<session number>***. This session number must specify the the “file transfer” state.

7. Enter **hsend file=/tmp/sb/BkUpAppl/*<appl_name>*/*<all_files_name>***

   This starts the send of the <*all_files_name*> to the host, using the session hassigned in Step 6.

8. Enter **vi /vs/data/fts_config**

   Set DESTINATION to blank and set ORIGINATION to the name you stored the application under on the host in Step 5. Once the send has completed, this file is updated when the polling value is reached, and the receive command is initiated. Once the file is received the ***<name>*.vb** file is run. Some examples of what might be used in the ***<name>*.vb** file are **backup_appl**, **restore_appl**, and/or **install_appl**, to first make a backup of the original application, then to restore the new application, and to finally install the new application. Once the receive is complete, the dates on the appl files in **/att/trans/sb/*<appl>*** should be close to the current time.

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 5 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 1 or more of the 5 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. Perform the “Stopping the Voice System” procedure in Chapter 4, “Common Maintenance Procedures,” of *Intuity CONVERSANT VIS V5.0 Maintenance*, 585-310-153, for details.
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. Perform the “Starting the Voice System” procedure in Chapter 4, “Common Maintenance Procedures,” of *Intuity CONVERSANT VIS V5.0 Maintenance*, 585-310-153.

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.

NOTE:
Do not change this value.

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.

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(s) are executed.
- The session recovers and becomes logged-in.

Figure 7-6 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 will wait 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.

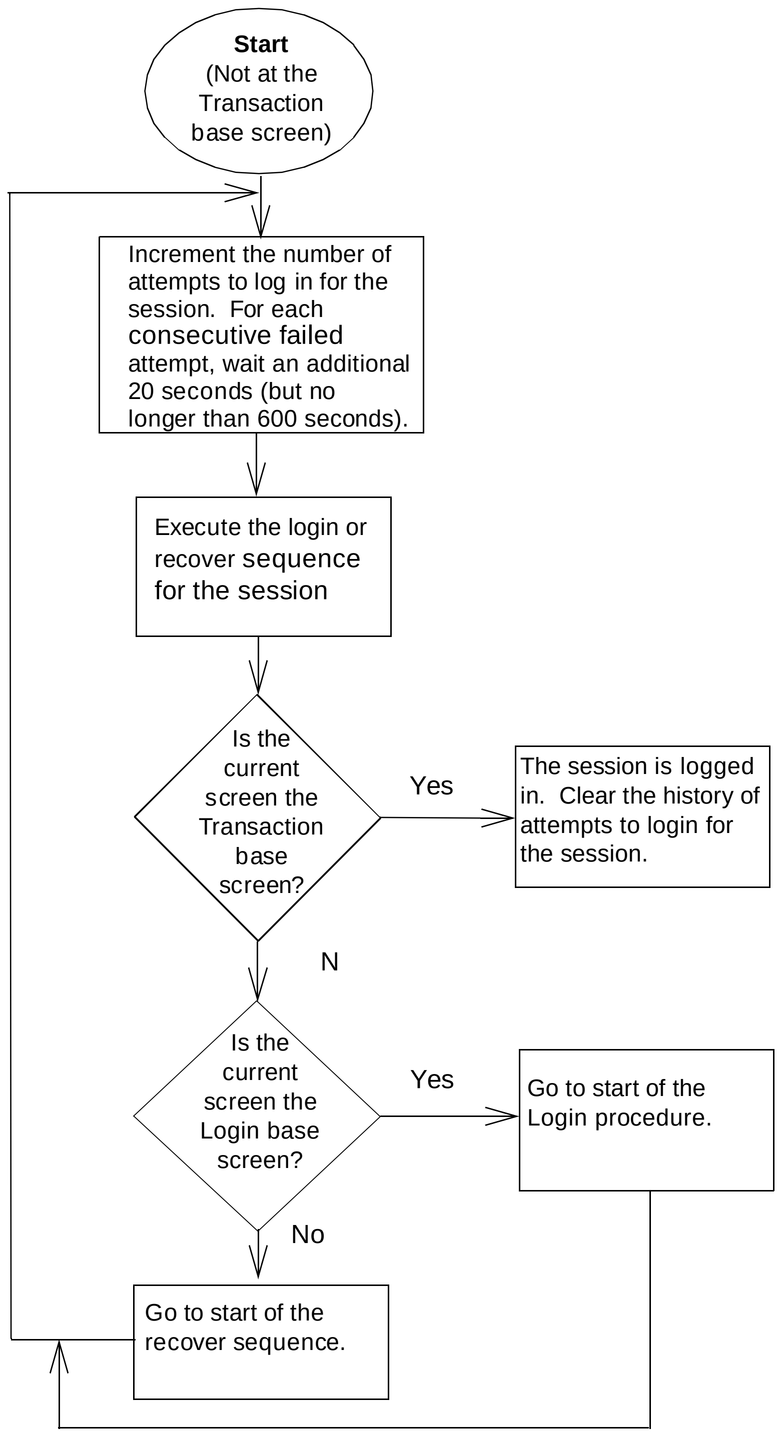


Figure 7-6. How a Session Tries to Log in

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 token ring networks are faster than SDLC connections, it is possible that a host application will experience more intermediate screens over a token ring network. If an application is moved from an SDLC environment to a token ring 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 VIS provides high-speed data transmission over an Ethernet or token ring network.

There are three areas that you must be address when using TCP/IP protocol with the VIS.

- Current network topology — Refer to "Network Architecture" below.
- Application structure — Refer to "Application Development Issues" below.
- Software installation — Refer to Chapter 3, "Installing the Optional Feature Software," of *Intuity CONVERSANT VIS V5.0 Software Installation,* 585-310-151.

Refer to *NFS/RPC/NIS Administration* and *TCP/IP Administration* for additional information on TPC/IP protocol. Refer to the *SQL*NET TCP/IP User's Guide* for additional information on using SQL*NET TCP/IP.

Network Architecture

UnixWare 1.1 includes an implementation of the TCP/IP protocol. The package has been internetworked successfully by AT&T and others with a wide variety of TCP/IP networks. Given this standard and compliant implementation, there is no reason that a VIS running this software cannot be connected successfully to a standard, compliant TCP/IP network.

Figure 7-7 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 illustrates 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 VIS 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

Refer to *UNIX SVR4.2 Network Administration* for additional 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.

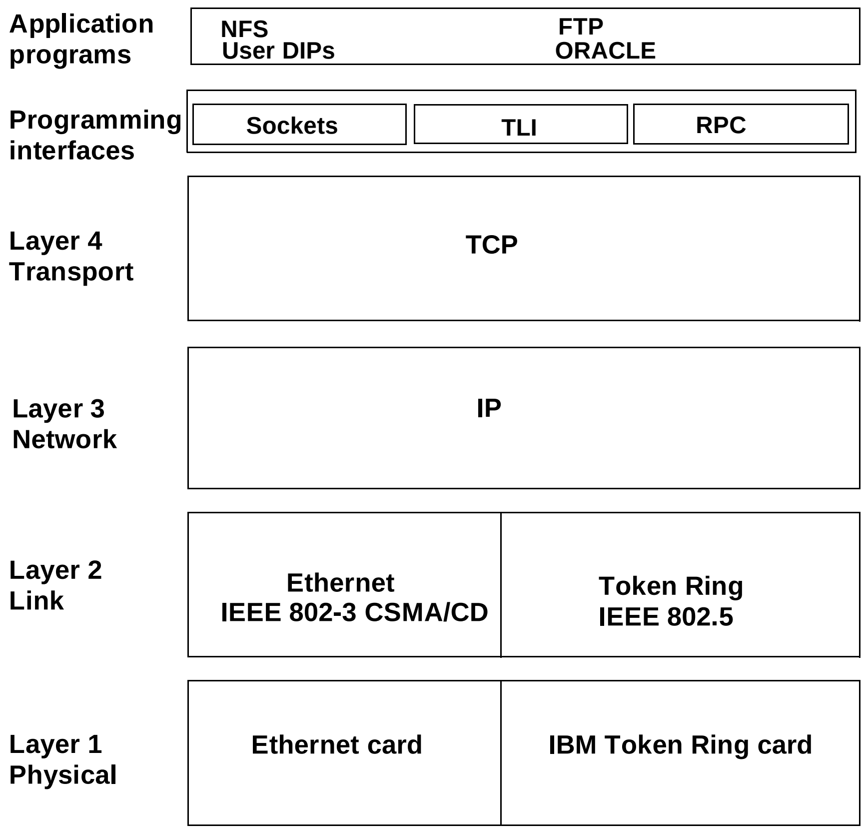


Figure 7-7. Network Layering

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 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.

Refer to Chapter 1, “Administering TCP/IP Networks,” of *TCP/IP Administration* in the *Novell UnixWare Documentation Set*, 585-310-908, for additional information on setting up the network.

Hardware Requirements

Using the TCP/IP protocol on the VIS requires either an Ethernet or Token/Ring circuit card, depending on the physical and link layer.

The SMC Ethernet card supports the following physical interfaces to the network:

- External transreceiver
- 10BASE2 (ThinNet)
- 10BASET (Twisted Pair)

The IBM token ring card supports the following physical interfaces to the network:

- IBM token ring Network PC Adapter cable
- Category 3, 4, or 5 cable

Refer to the hardware installation book for your platform for information on installing these network cards.

Software Requirements

The following software packages must be installed on the VIS to use TCP/IP protocol:

- UnixWare 1.1
- The driver specific to the circuit card installed on the VIS (either Ethernet or token ring)

Refer to the *Intuity CONVERSANT VIS V5.0 Software Installation*, 585-310-151, for installation procedures.

Application Development Issues

Typically, a VIS is added to a network that is already in place. Adding a VIS 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. Refer to *UNIX SVR4.2 Network Administration* for additional 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. Refer to Chapter 4, "Data Interface Process," of *Intuity CONVERSANT VIS V5.0 Application Development*, 585-310-227. When writing the DIP, you must use the Sockets, TLI, or RPC application programming interface (refer to *NFS/RPC/NIS Administration*). Within Script Builder you must create an external action to call the **dbase** script instruction to execute the DIP. Refer to "Using Advanced Features," of *Intuity CONVERSANT VIS V5.0 Script Builder*, 585-310-727.

It is also possible to use sockets, TLI, or RPC with an Intuity Response Application Programming Interface (IRAPI) application. Care must be used to determine who the process should block. Refer to the *Intuity CONVERSANT VIS V5.0 IRAPI Programming Guide*, 585-310-226, for information.

SQL*NET Communications

SQL*NET is the ORACLE communications component that allows the Intuity CONVERSANT VIS to share information stored in different remote ORACLE databases. With SQL*NET, you can run an ORACLE tool or another application on the VIS and be able to find, manipulate, and store data in an ORACLE database located on another machine.

For additional information on ORACLE SQL*NET communications, refer to *ORACLE SQL*NET TCP/IP Documentation*, 585-350-913.

Asynchronous Communications

Asynchronous communications 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 VIS supports two standard asynchronous connections and one standard parallel printer connection on each of the Multi-Application Platforms (MAP) via an EIA-232 serial port. One of the standard asynchronous connections is reserved for the Remote Maintenance board. This circuit card provides a standard modular connection for access to the built-in modem. This arrangement allows access to the VIS 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 VIS platform.

The VIS also supports both local alarm relay unit (ARU) and remote Switching Control Center System (SCCS) alarming systems, which communicate with the VIS through asynchronous connections. Refer to Chapter 8, "Data Network Connectivity Alarms", for diagrams for these types of connections.

Data transmission is limited to 9600 bps (maximum) for asynchronous communication established with any device.

Refer to the section “Ports” and “Printers” in Appendix A, “System Administration Features,” of the *Intuity CONVERSANT VIS V5.0 Operations*, 585-310-550, for information on setting up the ports and printers.

Standard Asynchronous Connections

The standard asynchronous ports are located on the back of each MAP/100C, MAP/100, and MAP/40 unit. These connections and their locations are described for each MAP later in this section.

Note that the distance between transmission devices (for example, the VIS and a terminal) should not exceed 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 50 to 5000 feet. Refer to the appropriate ADU documentation for maximum limits.

MAP/100C Asynchronous Communication Ports

You can connect the MAP/100C platform to a terminal, modem, or host computer through an asynchronous link connected to one or more ports on the platform. The system is connected to a printer through a single parallel port. The standard connections include:

- Two 25-pin D-subminiature male port, COM2, located on the ront and rear lower center of the MAP/100C. Both connectors provide access to the COM2 port on the CPU, for convenient access to central office rack-mounted systems.

  NOTE:
  The Remote Maintenance board uses the COM2 serial port.

- A 9-pin D-subminiature male port, COM1, is located on the faceplate of the CPU circuit card, which is accessed by opening the card cage access door.

  NOTE:
  The COM1/COM2 orientation is different between MAP/100C and the commercially available MAP/100 and MAP/40 machines. On the MAP/100C, the CPU-mounted connector is labeled "COM2," while the externally wired connector(s) are labeled “COM1.” This is reversed for the MAP/100 and MAP/40 machines.

- An Asynchronous Communications Interface feature package is available on the MAP/100C to provide eight additional RJ-45 type modular connector asynchronous ports. Prior to Version 5.0, this package included an optional circuit card, software, and T-adapter. Version 5.0 includes an 8-port asynchronous circuit card and a software driver.
- A parallel port connection is located on the faceplate of the CPU card. This port is a 25-pin male connector and is used as a printer interface.

MAP/100 Asynchronous Communication Ports

You can connect the MAP/100 platform to a terminal, modem, or host computer through an asynchronous link connected to one or more ports on the platform, including:

- A 9-pin D-subminiature male port, COM1, located on the faceplate of the CPU circuit card

  NOTE:
  The Remote Maintenance Circuit Card uses the COM1 serial port.

- A 9-pin D-subminiature male port, COM2, located at the rear, upper left corner of the MAP/100 chassis

- An Asynchronous Communications Interface feature package is available on the MAP/100 to provide eight additional RJ-45 type modular connector asynchronous ports.. Prior to Version 5.0, this package included an optional circuit card, software, and T-adapter. Version 5.0 includes an 8-port asynchronous circuit card and a software driver.
- A parallel port connection is located on the faceplate of the CPU card. This is a 25-pin male connector and is used as a printer interface.

MAP/40 Asynchronous Communication Ports

The MAP/40 platform can be connected to a terminal, modem, or host computer through an asynchronous link connected to one or more ports on the platform, including:

- A 9-pin D-subminiature male port, COM1, located on the faceplate of the CPU circuit card

  NOTE:
  The Remote Maintenance Circuit Card uses the COM1 serial port.

- A 9-pin D-subminiature male port, COM2, located at the rear, middle right side of the MAP/40 chassis
- An Asynchronous Communications Interface feature package is available on the MAP/40 to provide eight additional RJ-45 type modular connector asynchronous ports. Prior to Version 5.0, this package included an optional circuit card, software, and T-adapter. Version 5.0 includes an 8-port asynchronous circuit card and a software driver.
- A parallel port connection is located on the faceplate of the CPU card. This is a 25-pin male connector and is used as a printer interface.

8-Port Asynchronous Circuit Card Connections

Each of the MAP systems support connections to one or more asynchronous host computers or additional modems via an 8-port asynchronous interface. Depending on the version of the platform, these eight additional serial ports are provided by either an IPC-900, Gemini 1000, or Equinox 8-port asynchronous circuit card.

These serial connection ports are configured as data terminal equipment (DTE). DTE ports require a crossover or “null modem” cable to connect to serial devices such as a terminal, computer, or printer. The term “crossover” refers mainly to the transmit and receive lines. To communicate with any of the devices mentioned above, the transmit line on the serial port must ultimately be connected to the receive line of the terminal device. Conversely, the receive line on the serial port must be connected to the transmit line of the terminal device.

Connecting to a modem does not require a crossover cable. A modem is normally considered data communications equipment (DCE). DCE ports require a modem or straight-through cable. The crossover of transmit and receive are handled within the modem.

The following adapters are available to allow DCE equipment to communicate with DTE and vice versa:

- Null modem adapter or cable. This adapter “flips” the transmit and receive lines while still maintaining the functions of the other lines, that is, data terminal ready (DTR) and ground. This device is normally used to connect one DTE device to a another DTE device.
- Terminal/printer adapter. This adapter provides a crossover function much the same as a null modem adapter.
- ACU modem adapter. This is an adapter or cable that provides a straight-through connection.
- Gender changers. Gender changers convert a male connector to female and vice versa. There are two types of gender changers, male/male and female/female. The functionality of the incoming lines is maintained on the outgoing side.
- Modular extenders. Extenders allow you to connect two modular cables to each other without losing functionality. An extender consists of two female RJ-45 type ports linked to each other. The number of conductors in the extender must match the number of conductors in the cables used. There are three types of modular cables used with asynchronous communications within the VIS:
  - — A 6-conductor telephone hook-up cable (three pair) for analog Tip/Ring (T/R) connections.
  - — An 8-conductor cable is used for serial port peripheral connections (the standard serial ports provided on each VIS platform)
  - — A 10-conductor cable is used to connect devices with the modular ports provided on the 8-port asynchronous circuit card.

It is possible to connect 8-conductor to 10-conductor cables. The adapters used with the 8-conductor cable must be 8-pin adapters. Ten-pin adapters can be used with 10-conductor modular cables only. Eight-pin adapters can be connected to 10-pin adapters. However, check the wiring diagrams of both adapters to make sure that there is not loss of functionality when connecting 8- to 10-pin adapters.

In most cases, if transmit goes to receive (and vice versa) in connecting DTE devices, any combination of equipment can be used. For modems, it is most likely that a straight-through connection is required since they are DCE devices. However, you should confirm the pin positions of other functions (that is, DSR, DTR, carrier, etc.) on all connected devices to ensure proper functionality.

8-port Asynchronous Connections to Terminals

Figure 7-8 and Figure 7-9 show examples of external connectivity and cabling for a 8-port asynchronous connection to a terminal. Note that these are only examples and not an exhaustive list of possible connections.

NOTE:
The type of conductor cable (6- or 10-conductor) depends on 8-port asynchronous unit (Equinox, IPC900, or Gemini 1000) from which you are making the connection. Refer to the Appendix B, “Cable Connectivity,” of the hardware installation book for your platform for a list of the parts required.

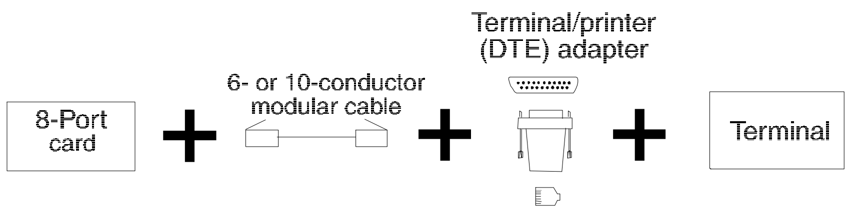


Figure 7-8. 8-port Asynchronous Terminal Connection Using 6- or 10-Conductor Modular Cable

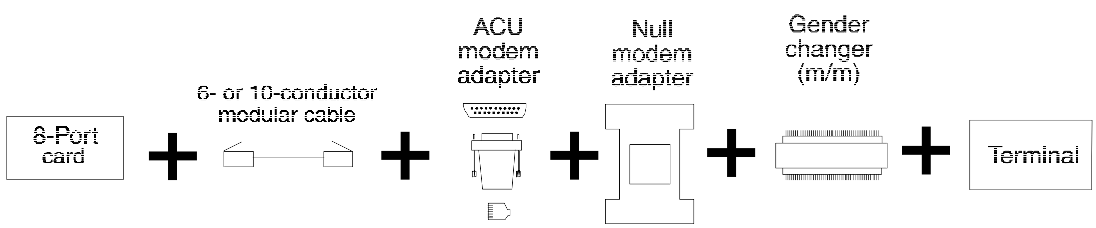


Figure 7-9. 8-port Asynchronous Terminal Connection Using 6- or 10-Conductor Modular Cable and a Null Modem

8-port Asynchronous Connections to Computers

Figure 7-10 and Figure 7-11 show examples of external connectivity and cabling for a multi-port asynchronous connection to a computer. Note that these are only examples and not an exhaustive list of possible connections.

NOTE:
The type of conductor cable (6- or 10-conductor) depends on 8-port asynchronous unit (Equinox, IPC900, or Gemini 1000) from which you are making the connection. Refer to the Appendix B, “Cable Connectivity,” of the hardware installation book for your platform for a list of the parts required.

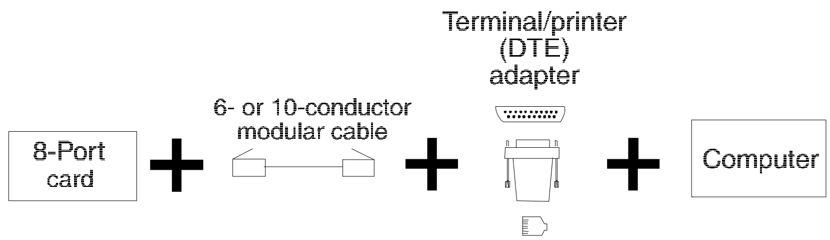


Figure 7-10. 8-port Asynchronous Computer Connection Using 6- or 10-Conductor Modular Cable

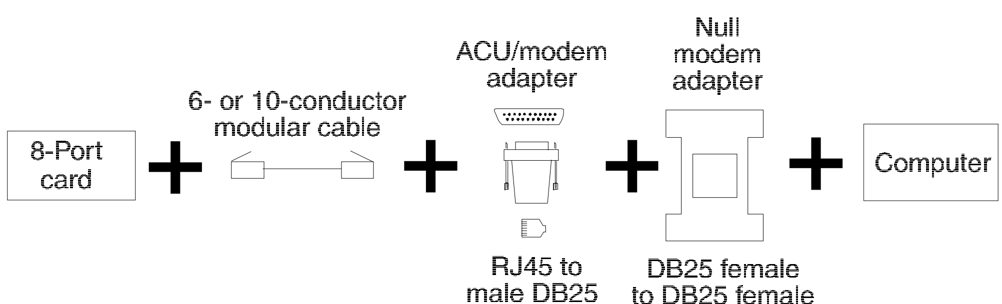


Figure 7-11. 8-port Asynchronous Computer Connection Using 6- or 10-Conductor Modular Cable and a Null Modem

8-port Asynchronous Connections to an External Modem

Figure 7-12 and Figure 7-13 show examples of external connectivity and cabling for a multi-port asynchronous connection to an external modem. Note that these are only examples and not an exhaustive list of possible connections.

NOTE:
The type of conductor cable (6- or 10-conductor) depends on 8-port asynchronous unit (Equinox, IPC900, or Gemini 1000) from which you are making the connection. Refer to the Appendix B, “Cable Connectivity,” of the hardware installation book for your platform for a list of the parts required.

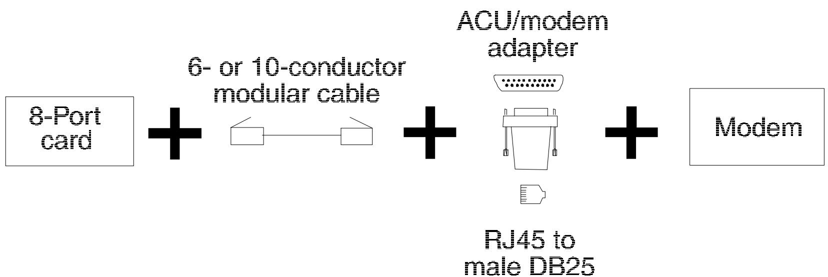


Figure 7-12. 8-port Asynchronous Modem Connection Using 6- or 10-Conductor Cable

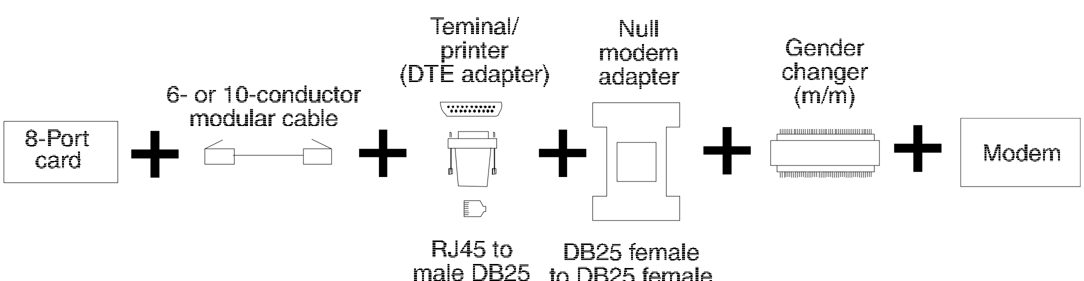


Figure 7-13. 8-port Asynchronous Modem Connection using 6- or 10-Conductor Cable and a Null Modem

8-port Asynchronous Connection to an ADU

Figure 7-14 shows an example of external connectivity and cabling for an 8-port asynchronous connection to an ADU.

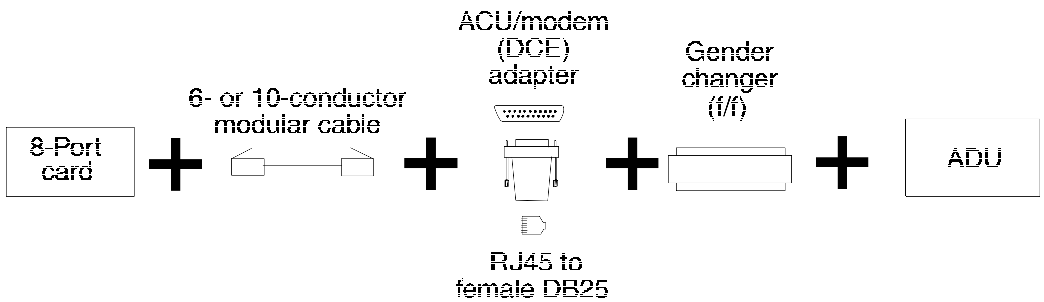


Figure 7-14. 8-port Asynchronous ADU Connection

8-port Asynchronous Connection to a Printer

There are two ways to connect the 8-port asynchronous unit to a printer. If you are connecting to the serial port on the printer, connect the 8-port card the same way that you would connect to a terminal as shown in Figure 7-8 on page 7-48. You can use a DTR printer adapter in place of the DTE terminal/printer adapter if the software uses the hardware flow control for the specified port.

Data Network Connectivity Alarms

8

What's in This Chapter

The following data network alarming packages are available for use in conjunction with the Intuity CONVERSANT Voice Information System (VIS) V5.0 software:

- NetView
- CompuLert/SCCS
- External Alarms

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

NetView Alarming

The NetView Alarming software package interacts with the Intuity CONVERSANT VIS V5.0 software to allow you to monitor VIS messages as part of your current NetView environment. The VIS logs alarms and events that occur during voice system operations. The VIS maintenance transmitter (mtcxmtr) process scans this log for error conditions and transmits critical, major and minor errors to the host as Operator-Generated Alerts (OGAs) over the 3270 host link.

Refer to the *NetView User's Guide* that is a part of the *CLEO LINKix Documentation Set*, 585-310-912, for information on accessing the NetView program, using the NetView screen display and commands, and using the Network Management Application Programing Interface (NM-API).

Configuring NetView

The NetView Alarming package is now bundled with the host interface offer (refer to Chapter 7, "Data Network Communications"). If you do not want NetView alarms sent to the host, remove the Intuity CONVERSANT VIS V5.0 3270 NetView Alarm Interface package. Refer to Chapter 3, "Installing Optional Features," of *Intuity CONVERSANT VIS V5.0 Software Installation*, 585-310-151.

By default, the host interface software is configured at installation to send all NetView alerts over the first host connection that is defined. To change the connection over which the alarms are sent

1. Enter **hconfig -a *conn_name***, where *conn_name* is the name of the connection you want to use.
2. Stop and start the voice system. Refer to Chapter 4, "Common Maintenance Procedures," of *Intuity CONVERSANT VIS V5.0 Maintenance*, 585-310-153.

When migrating to an environment with different NetView requirements, you must reconfigure NetView by editing the NetView configuration file. This configuration file contains flag settings that are used for configuring, monitoring, and testing the maintenance transmitter. The values set at installation should be satisfactory for normal operation in most environments. However, some environments may require a modification of some of these configuration flags.

The maintenance transmitter reads the configuration file when it starts up or when it receives the signal SIGUSR2. It is therefore possible to change the behavior of the maintenance transmitter by either sending it a SIGUSR2 signal or using it to restart and automatically read the configuration file.

NOTE:
For either of the following approaches, you must use the Process ID number (PID). Enter **ps -ef | grep mtcxmtr** to determine the PID. The PID is the leftmost number displayed on the output.

Changes in the configuration file take effect when you do one of the following:

- Enter **kill -17 *<PID>*** where *<PID>* is the process id number to send a SIGUSR2 signal to the maintenance transmitter. This approach is nondisruptive to the system (for example, queued alarms are saved).
- Enter **kill -9 *<PID>*** where *<PID>* is the process id number to restart the maintenance transmitter. This approach causes any accumulated error messages to be lost, while maintaining the host connection. The advantage of this approach is that it will start a new log, making it easier to locate the results of subsequent tests.

  NOTE:
  When you restart the maintenance transmitter, the previous log that records the transmitter's actions is moved to **/tmp/al_log.old**.

Configuration information is stored in an ASCII file called **/vs/data/alarm_flags**. To view or edit the contents of this file, log in as root and enter **vi /vs/data/alarm_flags**

Figure 8-1 shows the default values in the **/vs/data/alarm_flags** file. A description of each flag in the file follows the example.

```
Setbuf=1
Verbose=3
Tst=0
fail_mod=10
Wrap_ct=20,Rate=400
```

- Setbuf

  The Setbuf flag indicates whether messages sent to the log file will be buffered. A nonzero integer setbuf value indicates that messages will not be buffered.

  NOTE:
  Unbuffered writes to the log file are less efficient than buffered writes. Use unbuffered writes only when you want to guarantee the integrity of the log file through a system crash.

- Verbose

  The Verbose flag indicates the level of detail of the maintenance transmitter activity in the log. The following details the possible Verbose settings.

-1	No logging
3	Basic actions (default)
7	Detail of structure sent to the host
10–25	Increasing amount of detail in the log

NOTE:
The default value of 3 is strongly recommended for normal operation. The log is normally written without buffering from the UNIX operating system.

- Tst

  The Tst flag indicates the mode of operation. A Tst setting of 0 indicates remote operation. A Tst setting of 1 indicates local testing. A Tst setting of 2 indicates local testing but generates the maximum number of OGAs storable when any are read.

- fail_mod

  The fail_mod flag indicates the fraction of transmissions that will fail in test mode. The fail_mod flag causes the maintenance transmitter to simulate one transmission to fail. For example, if fail_mod is 4, an average of 25% of the transmissions will fail, with the transmissions of that file being randomly selected. The fail_mod flag if used only in test mode (that is, with Tst set to 1 or 2).

- Wrap_ct

  The Wrap_ct flag is used along with the Rate flag (described below) to define the maximum number of OGAs that can be transmitted. In the production environment (normal operation), set Wrap_ct to 10. In the test environment, set Wrap_ct to 20. To ignore Wrap_ct constraints established by NetView, set Wrap_ct to 0.

- Rate

  The Rate flag is used along with the Wrap_ct flag to define the maximum number of OGAs that can be transmitted. In the production environment (normal operation), set Rate to 1200. In the test environment, set Rate to 400. To ignore Rate constraints established by NetView, set Rate to 0.

  The Wrap_ct and Rate flags are used in the following manner: when an OGA arrives, NetView verifies that the maximum number of OGAs (as defined by the Wrap_ct flag) arrived in the allowable number of seconds (as defined by the Rate flag). For the test environment, Wrap_ct is 20 and Rate is 400. Consequently, when an OGA arrives in the test environment, the 20th previous OGA should have arrived not less than 400 seconds ago. For the production environment, Wrap_ct is 10 and Rate is 1200. Consequently, when an OGA arrives in this environment, the 10th previous OGA should have arrived not less than 1,200 seconds ago.

  NOTE:
  The host interface card (the PC/XL and/or the FIFO-SIB) accepts OGAs at a maximum average rate of approximately one per second. Within these limits, the maintenance transmitter sends OGAs as soon as possible in the first-in, first-out order.

Testing the Maintenance Transmitter

To test the maintenance transmitter, instruct the VIS software to send a known set of error messages to it and observe the resulting OGAs.

1. Use the command **logit** to generate error messages. For example, you can construct a script that contains the following lines to drive NetView testing:

   ```
   logit -p minor -d 0xffff “This is test message 1”
   logit -p minor -d 0xffff “This is test message 2”
   logit -p minor -d 0xffff “This is test message 3”
   logit -p minor -d 0xffff “This is test message 4”
   logit -p minor -d 0xffff “This is test message 5”
   ```

   Refer to the *Intuity CONVERSANT VIS V5.0 Command Reference*, 585-310-230, for additional information on using the **logit** command.

2. View the transmitter’s output either by using NetView (if there is a live host connection) or by examining the log **/tmp/al_log**. Input to the maintenance transmitter can be determined by examining the log.

   The maintenance transmitter should pass the following tests:

   - Every logged message or priority — Critical, Major, and Minor — should generate an OGA.
   - Only error messages of priority Critical, Major, and Minor should generate OGAs.

- The maintenance transmitter should follow the NetView constraints as expressed in the Wrap_ct/Rate line of the configuration file.
- If the connection to the host is lost, the transmitter should check the link at 5-minute intervals and resume sending messages 5 minutes after the connection is reestablished. If the maintenance transmitter receives up to 100 OGAs, these messages are stored and sent at a later time. If the transmitter receives more than 100 OGAs, the messages that were received first are overwritten and lost. Consequently, when the link is restored, not all of the messages received while the link was down are sent.
- OGAs should be in a format defined in "Configuring NetView" earlier in this chapter.

NOTE:
You can test everything but the NetView constraints in local mode. If you test in local mode, however, the failure rate due to the host interface card blocking may not be realistic.

Figure 8-1 shows the maintenance transmitter sending five OGAs to the host computer. The text of the set of alarms to send is enclosed by the SEND_2_Host and N alarms in the alarmbuf line.

```
                  SEND_2_HOST:  Feb 22 04:48:21

0 <*  LOGIT GEN002 -- -- --- root: This is test message 1 04:55>:
 Feb 22 04:58:21

1 <*  LOGIT GEN002 -- -- --- root: This is test message 2 04:55>:
 Feb 22 04:58:21

2 <*  LOGIT GEN002 -- -- --- root: This is test message 3 04:55>:
 Feb 22 04:58:21

3 <*  LOGIT GEN002 -- -- --- root: This is test message 4 04:55>:
 Feb 22 04:58:21

4 <*  LOGIT GEN002 -- -- --- root: This is test message 5 04:55>:
 Feb 22 04:58:21


                     5 alarms in alarmbuf:  Feb 22 04:58:21
```

Figure 8-1. Maintenance Transmitter Log Example

CompuLert/SCCS

The CompuLert/SCCS interface package allows error messages to be sent to the VIS error log and to the SCCS link. The data connection between the VIS and the SCCS is a standard serial asynchronous data connection operating at 9600 bps. (Refer to Chapter 7, "Data Network Communications") This same connection makes it possible to log in to the VIS from the SCCS and monitor or administer the VIS. It also provides for the transmission of a date-and-time *heartbeat* message from the VIS to the SCCS every 15 minutes as an indication of a functional link between the two systems.

The CompuLert/SCCS interface package also supports an optional Alarm Relay Unit (ARU) for local monitoring. The ARU is a microprocessor-based external unit attached to an asynchronous port on the VIS and driven by control character sequences that activate its features. The ARU generates alarms based on error messages sent from the VIS. It provides contact closures for Critical, Major, Minor, and power failure alarms and displays the alarm type (“CRIT,” “MAJOR,” “MINOR,” or “POWER”, respectively, on an LED strip on its front panel. You will also see the letter “A” or “B” following the alarm type, depending on whether the VIS is connected to port A or port B on the ARU. The ARU can also sound an audible tone to indicate an alarm condition.

The ARU also features a watchdog timer that displays an alarm message (“WDOGA” or “WDOGB”) and sends a tone if the VIS fails to send the ARU a date-and-time message once every 2 minutes. This would occur, for example, if the voice system is stopped or in the event of a power failure on the VIS. A heavily loaded system that is slow to come up can also cause the watchdog alarm message and tone.

Compulert/SCCS/ARU Connectivity

Figure 8-2 shows two different connectivity arrangements for the SCCS/ARU interface. Both alarm monitors can interface with the VIS through a serial port. Refer to the Appendix B, “Cable Connectivity,” of the hardware installation book for your platform for a complete list of the parts required to make these connections.

NOTE:
The 6-, 8- and 10-conductor cables shown in Figure 8-2 must be gray straight-through cables for this configuration to work properly. The black null-modem cables will not work in this configuration.

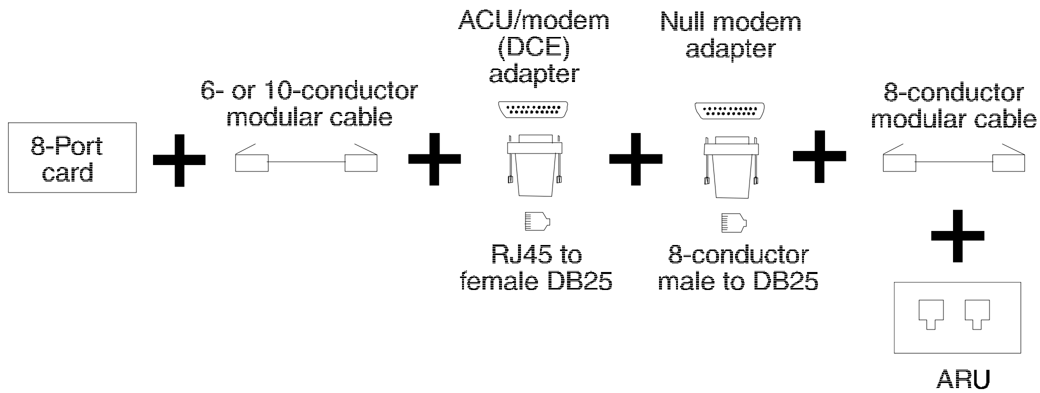


Figure 8-2. VIS and SCCS/ARU Connectivity

SCCS and ARU Provisioning

There are several requirements that must be met to correctly provision the CompuLert/SCCS interface on the VIS:

- The CompuLert/SCCS Interface software package must be installed on the V5.0 VIS. Refer to Chapter 3, “Installing the Optional Feature Software,” of *Intuity CONVERSANT VIS V5.0 Software Installation*, 585-310-151.
- If you require an ARU, it must be installed and operational. The ARU should be attached to a dedicated asynchronous port on the VIS. Refer to the hardware installation book for your platform for additional information.
- The serial port(s) on the VIS must be set to use the CompuLert/SCCS and ARU. Refer to the section “Ports” in Appendix A, “System Administration Features,” of the *Intuity CONVERSANT VIS V5.0 Operations*, 585-310-550, for information on port set up.
- The ARU and SCCS connections must be tested to ensure proper operation of the monitor mode from the SCCS. Refer to Chapter 3, “Installing the Optional Feature Software,” of *Intuity CONVERSANT VIS V5.0 Software Installation*, 585-310-151.

SCCS Interface Parameters

A standard options file controls a number of parameters for the sccsDaemon process. Table 8-1 describes some of the key dynamic options contained in the **/vs/etc/default/sccsDaemon** file.

After changing any parameter in this file, you must restart the daemon process if the system is running. This is not necessary if the system is stopped. To restart the daemon process, enter **/vs/bin/vrs/sccsDaemon -c quit**

NOTE:
Before making any changes to the **/vs/etc/default/sccs/Daemon** file, make a copy of the original and name it differently.

Table 8-1. Dynamic Options of the sccsDaemon Process

Option	Description	Default
CONSOLE_OUTPUT_DEFINED	Route messages to the SCCS device to the console as well	FALSE
CONSOLE_DEVICE	The name of the device or file to which console messages are directed	**/dev/console**
ARU_ENABLED	Enables sending of alarm messages and watchdog reset messages to the ARU (this parameter can be disabled entirely when the system does not include an ARU)	TRUE
SCCS_MODES	The **stty** command used to condition the line to the SCCS	**stty sane 9600 erase ‘^h’ echoe echok**
ARU_MODES	The **stty** command used to condition the line to the ARU	**stty sane 9600 erase ‘^h’ echoe echok**

External Alarms

The External Alarms Interface package provides a means for administering external alarms in a central office environment. This package can be used only on the MAP/100C. This section describes how to provision the External Alarms feature for monitoring the VIS software on a MAP/100C.

External Alarms Relay Contacts

To provision this feature correctly, you must understand the relay contacts used by the External Alarms Interface card. The External Alarms Interface card includes relays controlled by a sanity timer, by power failure, and by the VIS software. The relays can alert receptors external to the VIS of problems with the VIS.

Sanity Timer Relay Contacts

The Sanity Timer controls relay number 8 or relays 7 and 8 depending on how the card is configured (refer to the hardware installation book for your platform for additional information). The Sanity Timer is used to indicate that the software on the VIS is running. The Sanity Timer must be reset periodically by a process on the VIS. As long as the sanity timer is reset by this VIS process, the Sanity Timer will not time out, and the relay(s) associated with the timer will not close. The sanity timer is updated by the alerter process on the VIS. The alerter process runs at run-level 2 so the Sanity Timer will not time out even if the voice system is stopped. The most likely cause of Sanity Timer time-out is a system crash or a system lockup.

Power Fail Relay Contact

The Power Fail Relay Contact, relay 1 on the External Alarm Interface card, remains closed as long as there is power to the External Alarm Interface card. Power comes to the card from the VIS backplane. The Power Fail Relay Contact opens if power is cut off from the VIS, or if the External Alarm Interface card is not seated properly in the VIS backplane. There is no software control available for the Power Fail Relay Contact.

Software-Controllable Relay Contacts

The remaining relay contacts are software controllable. That is, the VIS resident software can send commands to the External Alarm Interface card to open and close relay contacts. Software controllable relay contacts include relays 2 through 6 and possibly 7 depending on how the card has been configured (refer to the hardware installation book for your platform for additional information).

External Alarms Interface Software Features

The primary function of the software supplied with the External Alarms Interface Package is to close relay contacts when the VIS generates certain alarm-level messages. The software supports mapping messages to one or more relay contacts, or none at all. The software also provides an administrative command set. This command set supports the enabling and disabling of message-produced relay contact closures and state changes to the relay contacts themselves.

Software Interface to the External Alarms Interface Card

The software implementing the External Alarms Interface consists of a process that monitors system messages (the alerter) and a command set. The alerter is also responsible for updating the sanity timer at a regular interval. (The default is every 20 seconds.) The alerter uses a notion of Alarm Contact Sets. An Alarm Contact Set is a set of software controllable relay contacts. The file **/vs/data/alarms/masks** specifies the External Alarms card relays associated with a given Alarm Contact Set.

System messages are then assigned to Alarm Contact Sets through inclusion in one or more of the alarm files in **/vs/data/alarms**. For example, all messages assigned to Alarm Contact Set 1 are specified in **/vs/data/alarms/alarm1**.

When the system generates a message, the alerter reads it. If its ID is in one of the alarm files, the Alarm Contact Set associated with that file is closed. Note that a message ID can reside in more than one alarm file.

Note that Alarm Contact Sets and External Alarms Card relay contacts are not necessarily the same thing. Alarm Contact Sets provide a level of indirection between the software and the hardware. This allows more than one External Alarms Card relays to be assigned to a single Alarm Contact Set, and it allows more meaningful numeric identifiers to be associated with the relays. For instance, with the defaults settings, Critical, Major and Minor alarms are assigned to alarm contact sets 1, 2, and 3, respectively. However, Alarm Contact Sets 1, 2, and 3 map to Alarm Card relays 6, 5, and 4, which is nonintuitive. See "Mapping Alarm Contact Sets to Alarm Card Relays" later in this chapter.

External Alarms Connectivity

Figure 8-3 shows a possible External Alarms central office configuration. In this example, a machine alarm light is illuminated for the Sanity Time (Relay 8) as well as for Critical, Major, or Minor alarm occurrences (Relay 2). In addition, an aisle alarm grid is illuminated for the Sanity Timer (Relay 7), Critical (Relay 6), Major (Relay 5), and Minor (Relay 4) alarm occurrences. Relay 3 is unused in this configuration. The Grid Power, Sanity Time, or critical alarm lights the grid Critical light. The major alarm lights the grid Major light and the minor alarm lights the grid Minor light.

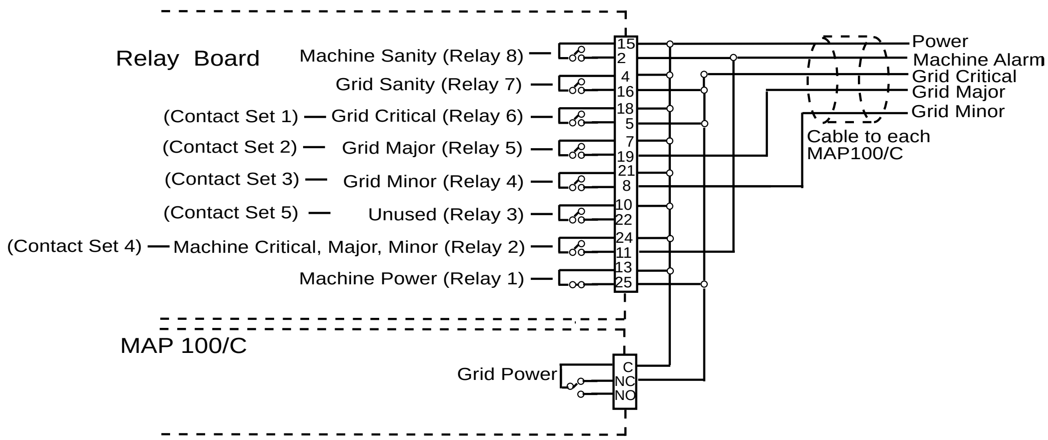


Figure 8-3. Alarm Relay Card Configured for a Central Office Application

External Alarms Administration

External Alarm Operational Commands

The External Alarms Interface package is delivered with a software command set for administration of the Alarm Contact Sets. The command set is implemented in the command **alarm**, which is executable from the UNIX system prompt. The External Alarms Interface provides the capability to enable, disable, display, reinitialize (reinit), retire, or test external alarms using the commands included with the External Alarms Interface package. Refer to the *Intuity CONVERSANT VIS Command Reference*, 585-310-230, for additional information on each of these commands.

The External Alarms Interface enable and disable features allow you to enable or disable a specified Alarm Contact Set. By default, all Alarm Contact Sets are set to enable; that is, the Alarm Contact Set is operational. Note that if the Alarm Contact Set is enabled, the contacts close upon receiving an assigned message or performing the alarm test command.

The **alarm display** command allows you to display the state of the external Alarm Contact Sets. The external Alarm Contact Sets are either OFF (the contacts are open indicating that no assigned message has occurred) or ON (the contacts are closed due to the occurrence of an assigned message).

Mapping Alarm Contact Sets to Alarm Card Relays

The software delivered with the External Alarms package provides a mapping from Alarm Contact Sets to Alarm Card Relays. When the software is installed, Alarm Contacts are assigned as follows:

alarm contact set 1:Alarm Card Relay 6
alarm contact set 2:Alarm Card Relay 5
alarm contact set 3:Alarm Card Relay 4
alarm contact set 4:Alarm Card Relay 2

It is possible that your application may require a different mapping. You can change the mapping of alarm contact sets to alarm card relays by editing the **/vs/data/alarms/masks** file.

The basic format of this file is:

<alarm contact set> <relay> [<relay>] ...

where *<alarm contact set>* is of the form alarm *X* and *X* is a single-digit number (for example: 1, 2, 3, ...) and *<relay> [<relay>]* ... is one or more software controllable alarm card relay numbers (2, 3, 4, 5, and/or 6)

Note that alarm contact sets must be disjointed; that is, two Alarm Contact Sets may not contain the same alarm relay number.

Note that there must be a **/vs/data/alarms/alarm*X*** file for each **alarm***X* defined in **/vs/data/alarms/masks**. The following shows a **/vs/data/alarms/masks** default file:

```
alarm1    6
alarm2    5
alarm3    4
alarm4    2
```

Another possible **/vs/data/alarms/masks** file could be as follows:

```
alarm1    6 5
alarm2    4 3
alarm3    2
```

In this scheme, if a message is generated that has an ID is in **/vs/data/alarms/alarm1**, Alarm Contact Set 1 is set; that is, Alarm Card Relays 6 and 5 are closed.

Setting the Sanity Timer Update Time

The External Alarms card is equipped with a sanity timer. The sanity timer is used to inform you that the voice system may have stopped operating. This timer must be reset before it times out or relays 7 and 8 will close. The sanity timer is reset every 20 seconds by default. The voice system periodically accesses the External Alarms card to reset the sanity timer. To change the reset period of the external alarms software, place a time-out value in the file **/vs/data/alarms/timer**. This value should be a positive integer that represents a number of seconds. To change the update time to every 30 seconds, change the 20 to 30 in the **/vs/data/alarms/timer** file. In the absence of a **/vs/data/alarms/timer** file, the VIS uses a 20-second time-out value.

It is also possible to change the time-out value on the External Alarms card. See the hardware installation book for your platform for information.

NOTE:
The software time-out value should be less than the hardware time-out value.

Voltage and Current Capacities for External Alarms Interface

Table 8-2 provides the current capacities for the External Alarms Interface hardware. If an inductive or capacitive load is being switched, the

- Peak turn-off or turn-on surge current must not exceed the DC current limit.
- Maximum AC or DC root mean square (RMS) load current must be less than the AC or DC current limit.

Table 8-2. Current Capacities for External Alarms

Voltage	Current (A)
250 VAC	5
30 VDC	5
125 VDC	1

Abbreviations

A

AC
Alternating current

ACD
Automatic call distributor

AD
Application Dispatch

AD-API
Application dispatch application programming interface

ADPCM
Adaptive differential pulse code modulation

ADU
Asynchronous data unit

AGL
Application generation language

ALERT
VIS Alerter process

ANI
Automatic number identification

API
Application programming interface

ARU
Alarm relay unit

ASAI
Adjunct/Switch Application Interface

ASCII
American Standard Code for Information Interchange

ASI
Analog switch integration

B

BB
Bulletin board

bps
Bits per second

BRDG
Call bridging process

BSC
Binary synchronous communication

C

CCA
Call classification analysis

CDH
Call data handler

CELP
Continuously Excited Linear Prediction

CGEN
Voice system general message class

CICS
Customer Information Control System

CMP
Companion circuit card

CMS
Call Management System

CO
Central office

CPE
Customer provided equipment or customer premise equipment

CPN
Calling party number

CPT
Call progress tones

CPU
Central processing unit

CSU
Channel service unit

CVS
Converse vector step

D

dB
Decibels

DB
Database

DBC
Database checking process

DBMS
Database management system

DC
Direct current

DCE
Data communications equipment

DCP
Digital communications protocol

DIO
Disk input and output process

DIP
Data interface process

DMA
Direct memory access

DNIS
Dialed number identification service

DSP
Digital signal processor

DTE
Data terminal equipment

DTMF
Dual tone multi-frequency

DTR
Data terminal ready

E

EBCDIC
Extended Binary Coded Decimal Interexchange Code

EIA
Electronic Industries Association

EISA
Extended Industry Standard Architecture

EMI
Electromagnetic interference

ESD
Electrostatic discharge

ESDI
Extended Serial Data Interface

ESS
Electronic Switching System

ET
Error tracker

EXTA
External alarms feature message class

F

FCC
Federal Communications Commission

FDD
Floppy disk drive

FEP
Front end processor

FFE
Form Filler Plus feature message class

FIFO
First-in-first-out processing order

foos
Facility out-of-service state

FTS
File transfer process message class

G

GEN
PRISM logger and alerter general message class

GSE
Graphical Speech Editor

GUI
Graphical user interface

H

HDD
Hard disk drive

HLLAPI
High Level Language Application Programming Interface

HOST
Host interface process message class

hwoos
Hardware out-of-service state

Hz
Hertz

I

IBM
International Business Machines

ICK
Integrity checking process message class

ID
Identification

IDE
Integrated Disk Electronics

IE
Information element

INIT
Voice system initialization message class

inserv
In-service state

IPC
Interprocess communication

IPC
Intelligent Ports Card (IPC-900)

IPCI
Integrated personal computer interface

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

IVP4
Integrated Voice Processing card with 4 analog channels

IVP6
Integrated Voice Processing card with 6 analog channels

IVPSS
Integrated Voice Processing System Software

K

Kbps
Kilobites per second

Kbyte
Kilobyte

L

LAN
Local area network

LDB
Local database

LED
Light-emitting diode

LIFO
Last-in-first-out processing order

LN
Load number

LOG
VIS logger process message class

LST1
Line side T1

LU
Logical unit

M

manoos
Manually out-of-service state

MAP/100
Multi-Application Platform 100

MAP/100C
Multi-Application Platform 100C

MAP/40
Multi-Application Platform 40

Mbps
Megabits per second

Mbyte
Megabyte

ms
Millisecond

msec
Millisecond

MHz
Megahertz

MTC
Maintenance process

N

NCP
Network Control Program

NEBS
Network Equipment Building Standards

NEMA
National Electrical Manufacturers Association

netoos
Network out-of-service state

NFAS
Non-Facility Associated Signaling

NFS
Network file sharing

NMVT
Network Management Vector Transport

NM-API
Network Management - Application Programming Interface

nonex
Nonexistent state

NRZ
Non Return to Zero

NRZI
Non Return to Zero Inverted

O

OEM
Original equipment manufacturer

OGA
Operator generated alert

P

PBX
Private branch exchange

PC
Personal computer

PCB
Printed circuit board

PCM
Pulse code modulation

PEC
Price element code

PRI
Primary rate interface

PSTN
Public switch telephone network

PS&BM
Power supply and battery module

R

RAM
Random access memory

RECOG
Speech recognition feature message class

RDBMS
ORACLE relational database management system

REN
Ringer equivalence number

RFS
Remote file sharing

RM
Resource manager

RMB
Remote maintenance board

RTS
Request to send

S

SBC
Sub-band coding

SCCS
Switching Control Center System

SCSI
Small Computer System Interface

SDLC
Synchronous Data Link Control

SDN
Software Defined Network

SID
Station identification

SIMM
Single inline memory module

SLIP
Serial Line Interface Protocol

SNA
Systems Network Architecture

SNMP
Simple Network Management Protocol

SP
Signal processor circuit card

SPIP
Signal processor interface process

SPPLIB
Speech processing library

SQL
Structured Query Language

SR
Speech recognition

SYS
UNIX system calls message class

sysgen
System generation

T

tas
Transaction assembler

TCC
Technology Control Center

TCP/IP
Transmission control protocol/internet protocol

TDM
Time division multiplexing

TE
Terminal emulator

THR
Threshold message class

TKR
Token Ring

TLI
Transport layer interface

TLP
Transmission level plan

T/R
Tip/Ring circuit card

TRIP
Tip/Ring interface process

TSO
Technical Service Organization

TSO
Time Share Operation

TSM
Transaction state machine process

TTS
Text-to-Speech

TWIP
T1 interface process

U

UK
United Kingdom

USOC
Universal service ordering code

UVL
Unified Voice Library

V

VDC
Video display controller

VIS
Intuity CONVERSANT Voice Information System

VPC
Voice processing comarketer

VRU
Voice response unit

VROP
Voice response output process

Glossary

Numerics

3270 interface
A link between one or more Intuity CONVERSANT Voice Information System (VIS) machines and a host mainframe. In Intuity CONVERSANT VIS documentation, the 3270 interface means the link between one or more VIS machines and an IBM host mainframe.

4ESS
A large AT&T central office switch used to route calls through AT&T's telephone network.

A

ACD
See "automatic call distributor."

ADPCM
See "adaptive differential pulse code modulation."

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, Adjunct/Switch Application Interface) that the Intuity VIS administers via cut-through access to the inherent management capabilities of the product itself; this is in opposition to CONVERSANT VIS's ability to administer the switch directly.

Adjunct/Switch Application Interface
An optional feature package that provides an Integrated Services Digital Network-based interface between AT&T PBX's and adjunct processors.

affiliate
A business organization that AT&T controls or which with AT&T is in partnership.

alarm relay unit
A unit used in central office telecommunication arrangements that transmits warning indicators from telephone communications equipment (like the Intuity CONVERSANT VIS) to audio.

alerter
A system process that responds to patterns of events logged by the "logdaemon" process.

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.

application
Made of several components that provide an automated version of the communication between a caller and an attendant. The Intuity CONVERSANT VIS provides several methods for creating applications, including Script Builder, the Intuity Response Application Programming Interface (IRAPI), and transaction state machine (TSM) script language.

application administration
The component of the Intuity CONVERSANT VIS that provides access to the applications currently available on your system and helps you to manage and administer them.

application installation
A two-step process in which the Intuity CONVERSANT VIS invokes the TSM script assembler for the specific application name and files are moved to the appropriate directories.

application verification
A process in which the Intuity CONVERSANT VIS verifies that all the components needed by an application are complete.

ASCII
An acronym for American Standard Code for Information Interchange, a standard for data representation. ASCII code represents alphanumeric characters as binary numbers. The code includes 128 upper- and lowercase letters, numerals, and special characters. Each alphanumeric and special character has an ASCII code (binary) equivalent that is 1 byte long.

asynchronous communication
A method of data transmission in which bits or characters are sent at irregular intervals and are spaced by start and stop bits and not by time. See also “synchronous communication.”

asynchronous data unit
An electronic communications device that allows computer systems to communicate over asynchronous lines more than 50 feet in length.

AUDIX Voice Power
A complete voice-mail messaging system accessed and operated by touch-tone telephones and integrated with a switch or “Private Branch Exchange.”

automatic call distributor
A telephone system that recognizes and answers incoming calls and completes these calls based on a set of instructions contained in a database. The Automatic Call Distributor 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
A utility that makes an archive copy of a completed application or makes an interim copy of an application in progress. The backup copy can be restored to the VIS if the online version is damaged, or if you make revisions and wish to go back to the previous version.

barge-in
A capability provided by WholeWord speech recognition that allow callers to speak their responses to the VIS prompt and have those responses recognized before the prompt has finished playing.

batch file
A file containing one or more lines, each of which is a command executable by the UNIX shell.

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.

bridging
The process of connecting one telephone network connection to another telephone network connection over the Intuity CONVERSANT VIS TDM bus. Bridging decreases the processing load on the system since an active bridge does not require speech processing, database access, host activity, etc., for the transaction.

BSC
See “binary synchronous communication.”

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), the equivalent of one character of text.

C

call classification analysis
An optional feature package that allows application developers to classify the disposition of originated and transferred calls.

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 someone making a telephone call. It can be used by telephone switching equipment to selectively route an incoming call to a particular department or agent.

caller
The party that calls for a service, gets connected to the Intuity CONVERSANT VIS, and interacts with the system. As the Intuity CONVERSANT VIS is also capable of making outbound calls for service, the caller can also be the person who responds to those outbound calls.

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 Intuity CONVERSANT VIS 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/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.

caution
An admonishment used when there is a possibility of a service interruption or a loss of data.

CCA
See “call classification analysis.”

CDH
See “call data handler process.”

central office
An office or location in which large telecommunication machines such as telephone switches and network access facilities are maintained. These locations follow strict installation and operation requirements.

central processing unit
A component of the Intuity CONVERSANT VIS that is based on either the Multi-Application Platform 100 (MAP/100), MAP/40, or MAP/100C.

channel
See “port.”

CICS
See “Customer Information Control System.”

circuit card upgrade
A new circuit card that replaces an existing one in the platform. Usually the replacement is an updated version of the other card, and the replacement is designed to deal with technology made obsolete by industry trends or a new VIS release.

cluster controller
A bisynchronous interface that provides a means of handling remote communication processing.

command
An instruction or request given by the user to the VIS software to perform a particular function. An entire command consists of the command name and options.

CompuLert/SCCS interface
An optional feature that enables remote or console monitoring of error messages generated from the Intuity CONVERSANT VIS. CompuLert is a centralized maintenance system for monitoring minicomputers, computer mainframes, etc. The Switching Control Center System (SCCS) is similar to the CompuLert system, but is used to support 4ESS local switching systems.

configuration
The arrangement of the software and hardware of a computer system or network. The Intuity CONVERSANT VIS configuration includes either a standard or custom processor, peripheral equipment (for example, printers, modems), and software applications. Configuration also refers to the way 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 VIS 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 SP and T1 cards, and perform various maintenance functions.

Converse Data Return (conv_data)
A Script Builder action that supports the DEFINITY call vectoring (routing) feature by enabling the switch to retain control of vector processing in the VIS environment. It supports the DEFINITY “converse” vector command to establish a two-way routing mechanism between the switch and the VIS 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 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. An example of this is the use of AUDIX Voice Power along with Intuity CONVERSANT on the same VIS platform.

CPU
See “central processing unit.”

crash
An interactive utility for examining the operating system core and for determining if system parameters are being exceeded.

custom speech
Unique words or phrases to be used in Intuity CONVERSANT VIS voice prompts that AT&T records for a customer on a custom basis.

custom vocabulary
A specialized package of unique words or phrased 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.

D

danger
An admonishment used when there is a possibility of personal injury.

data interface process
A software process that communicates with Script Builder 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 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 permanently store, in the transaction outline.

debug
The process of locating and correcting errors in computer programs. This process is 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 Tip/Ring, T1, or SP circuit cards or a bus.

dialed number identification service
A service that allows incoming calls to contain information about the telephone number for which it is destined.

directory
A type of file used to group and organize other files or directories.

DNIS
See “dialed number identification service.”

DIP
See “data interface process.”

display errdata
A command that displays system errors sent to the logger.

DTMF
See “dual tone multi-frequency.”

dual 3270 links
A feature that provides an additional physical unit (PU) to allow a cost-effective means of connecting to two host computers. The customer can connect a VIS 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 touch tone.

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 upon system crashes. The dump can be fetched after rebooting for analysis of what may have caused the crash.

E

editor system
A system that allows speech phrases to be displayed and edited by a user. See “Graphical Speech Editor.”

Enhanced File Transfer
A feature that allows the transferring of files automatically between the Intuity CONVERSANT VIS and a synchronous host processor on a designated logical unit.

Enhanced Serial Data Interface
A software- and hardware-controlled method used to store data on magnetic peripherals.

error message
A message on the screen indicating that something is wrong and possibly suggesting how to correct it.

Error Tracker process
See “etStub.”

Ethernet
A name for a local area network that uses 10BASE5 or 10BASE2 coaxial cable and InterLAN signaling techniques.

etStub
A system process that processes pre-Version 3.1 error message logging requests. These requests are transformed and passed on to the “logdaemon” process.

event
The notification given to an application when some condition occurs.

external actions
Specific tasks and interfaces controlled by Intuity CONVERSANT VIS software that allow a Script Builder application script to invoke processes and interact with other products or services. For example, a Intuity CONVERSANT VIS application script can invoke AUDIX Voice Power functionality through the used of an external action within an application script.

F

feature
A function or capability of a product or an application within the Intuity CONVERSANT VIS.

feature package
An optionally purchased package that may contain both hardware and software resources, which provides additional functionality to a standard system.

feature_tst script package
A standard CONVERSANT VIS software program that allows a VIS user to perform self-tests of critical hardware and software functionality.

field
A “slot” in a VIS window that holds one column of information in a row.

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.

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 of American English vocabularies. See “subword technology.”

Form Filler Plus
An optional feature package that provides the capability for application scripts to record caller’s responses to prompts for later transcription and review.

function key
A key, labeled F1 through F8, on your keyboard to which the Intuity CONVERSANT VIS software gives special properties for manipulating the user interface.

G

Graphical Speech Editor
A window-driven, X Windows/Motif based, graphical user interface (GUI) that can be accessed to perform different functions associated with the creation and editing of speech files to be used by VIS applications.

H

hard disk drive
A high-capacity data storage/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 floppy drives, etc., are all hardware.

hardware upgrade
Replacement of one or more fundamental platform hardware components (for example, the CPU or hard disk drive), but the existing platform and other existing optional circuit cards remain.

High Level Language Applications Programming Interface (HLLAPI)
An application programming interface that allows user to write custom applications that can communicate with the host via an API.

HLLAPI
See “High Level Language Applications Programming Interface.”

host computer
A computer linked to a network providing 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.

I

iCk
The system integrity checking process.

idle channel
A channel that either has no owner or is owned by its default owner and is onhook.

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.

indexed table
A table that, unlike a nonindexed table, can be searched via a field name that has been indexed.

initialize
To start up the system for the first time.

Integrated Services Digital Network
A network that provides end-to-end digital connectivity to support a wide range of voice and data services.

Integrated Voice Processing circuit card
The IVP4 or IVP6 circuit card.

intelligent transfer protocol
A transfer protocol that monitors the line after dialing is complete to determine whether a busy, reorder (fast busy), or other failure has been encountered. It also recognizes when the extension is answered or if the extension is not answered after a specified number of rings.

interface
The access point of a system. With respect to the Intuity CONVERSANT VIS, the interface is designed to provide you with easy access to the software's capabilities.

interrupt
The termination of voice and/or telephony functions when some condition occurs.

Intuity Response Application Programming Interface
A library interface the provides a standard development interface for voice-telephony applications.

ipcs
A command that reports interprocess communication facilities status.

IRAPI
See "Intuity Response Application Programming Interface."

ISDN
See "Integrated Services Digital Network."

K

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 that allows the VIS to recognize a single word in the middle of an entire phrase spoken by a caller in response to a prompt.

L

LAN
See "local area network."

library states
The state information about channel activities maintained by the IRAPI.

line side T1
A digital method of interfacing a Intuity CONVERSANT VIS 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 local area network provides communications between computers and peripherals.

local database
A database residing on the Intuity CONVERSANT VIS.

logical unit
A type of SNA Network Addressable Unit.

logdaemon
System information and error logging process.

logger
See “logdaemon.”

logging on/off
Entering or exiting the Intuity CONVERSANT VIS software.

LU
See “logical unit.”

M

magnetic peripherals
Data storage devices that use magnetic media to store information. Such devices include hard disk drives, floppy disk drives, and cartridge tape drives.

main screen
The Intuity CONVERSANT VIS VERSION 5.0 screen from which you are able to enter System Administration or Voice System Administration.

maintenance process
A software process that runs temporary diagnostics.

Manual Configurator Program
A software program that resolves or blocks the allocation of CPU and memory resources for controlling and optional circuit cards.

masked event
An event that an application can ignore (that is, the application can ask not to be informed of the event).

master
A board that provides clock information to the TDM bus.

megabyte
A unit of memory equal to 1,048,576 bytes (1024 x 1024). It is often rounded to one million.

Microsoft
A company that manufactures software products, primarily for IBM-compatible computers.

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 crashes 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-DOS
A personal computer disk operating system developed by the Microsoft Corporation.

MTC
See “maintenance process.”

multi-threaded application
A single process/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

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.

new error logging environment
A more flexible and informative environment for logging errors and status messages (introduced in CONVERSANT VIS Version 3.1). Customer applications created earlier than V3.1 that log messages require conversion to this new environment.

new operating system
The UnixWare operating system being introduced in Intuity CONVERSANT VIS V5.0.

nonindexed table
A table that may be searched only in a sequential manner and that cannot be searched via a field name.

nonmasked event
An event that must be sent to the application. Generally, an event is nonmaskable if the applicaiton would likely encounted state transition errors by trying to ignore the event.

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

obsolete hardware
Hardware that is no longer supported on Intuity CONVERSANT VIS V5.0.

on-line help
Messages or information that appear on the user’s screen when a “function key” (F1 through F8) is pressed.

Operator Generated Alerts
System monitoring messages transmitted from the CONVERSANT VIS or other computer system to an IBM host computer that are 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 will execute 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

PBX
See “private branch exchange.”

PCM
See “pulse code modulation.”

peripheral (device)
Equipment such as printers or terminals that is in addition to the basic processor.

permanent process
A process that starts and initializes itself before it is needed by a caller.

phoneme
A single basic sound of particular spoken language. The English language contains 40 phonemes that represent all basic sounds used with the language. As an example, the 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 filtering
The rejection of unrecognized speech. The WholeWord and FlexWord speech recognition packages can be programmed to reprompt the caller if the spoken response was not recognized by the VIS.

phrase tag
A string of up to 50 characters that identify 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.

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 to send.

polling
A network arrangement whereby a central computer asks each remote location whether they wish 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.”

Primary Rate Interface
An optional feature package that provides a digital interface capable both of receiving and originating telephone calls directly from/to an AT&T 4ESS switch.

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 Intuity CONVERSANT VIS documentation, the computer on which UnixWare and Intuity CONVERSANT VIS software runs. In general, the part of the computer system that processes the data. Also known as the “central processing unit.”

ps
A command that shows active processes. This command displays the process table and can be used to determine which processes are consuming large amounts of system resources, such as CPU time.

pseudo driver
A driver that does not control any hardware.

pulse code modulation
A digital modulation method of encoding voice signals into digital signals. See also “adaptive differential pulse code modulation.”

R

recovery
The process of using copies of the VIS software to reconstruct files that have been lost or damaged. See also “restore.”

remote database
The component of the VIS that provides access to information not currently on the VIS.

remote maintenance board
A Intuity CONVERSANT VIS board that is equipped standard on all new MAP/100 and MAP/40 platform purchases. This card, available with a built-in modem, allows remote personnel (for example, field support) to access all Intuity CONVERSANT VIS machines with a standard simplified process.

reports administration
The component of the VIS that provides access to system reports, including VIS call classification reports, call data detail reports, call data summary reports, message log reports, and traffic reports. In addition, if AUDIX Voice Power R2.1.1 is installed on your system, the reports administration component gives you access to AUDIX Voice Power 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 reusing an existing system component after a software upgrade or platform migration.

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.

S

sar
A command that is associated with the system activity report package.

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 Intuity CONVERSANT VIS 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 VIS.

SCSI
See “Small Computer System Interface.”

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.

Single Inline Memory Modules
A method of containing random access memory (RAM) chips on narrow circuit card strips that attach directly to sockets on the CPU circuit card. Multiple SIMMs are sometimes installed on a single CPU 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.

Small Computer System Interface
A disk drive control technology in which a single SCSI adapter card plugged into a PC slot is capable of controlling as many as seven different hard disks, optical disks, tape drives, etc.

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 Intuity CONVERSANT VIS Version 5.0 software.

software upgrade
The installation of a new version of software. The existing platform and circuit cards are kept.

source system
The system from which you are upgrading (that is, your system as it exists *before* you upgrade).

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-Kbyte blocks for efficient management and retrieval of talkfiles. The Intuity CONVERSANT VIS speech filesystem is not consistent with standard UNIX filesystems, and can not be referenced with standard UNIX commands such as **ls**, **cat**, etc.

speech modeling
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 phrase
A continuous speech segment encoded into a digital string.

speech space
An area that contains all digitized speech used for playback in the applications loaded on the system.

standard speech
The speech package containing simple words and phrases produced by AT&T for use with an Intuity CONVERSANT VIS. 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 the voice prompts played by the VIS.

standard vocabulary
A standard package of simple word speech models provided by AT&T and used for WholeWord speech recognition purposes. These phrases include the digits “zero” through “nine,” “yes,” “no,” and “oh.”

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 that recognizes phonemes or parts of words of American English vocabularies. See “whole-word technology.”

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 (on hook). The device is raised when the handset is picked up (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 VIS that enables you to define the interaction between the VIS and switches by allowing you to establish and modify switch interface parameters and protocol options for both analog and digital interfaces.

switch network
Two or more interconnected 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. See also “asynchronous communication.”

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 VIS software processing, performing daily system operations and preventive maintenance, and troubleshooting errors as required.

system architecture
The manner in which the Intuity CONVERSANT VIS software is structured.

system message
An event or alarm generated by either a VIS or end-user process.

system monitor
A component of the VIS in which tests are performed to verify that each incoming telephone line and its associated tip/ring or T1 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
A collection of records that are logically grouped together.

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.

target system
The system to which you are upgrading (that is, your system as you expect it to exist *after* you upgrade).

TDM
See “time-division multiplex.”

telephone network connection
The point at which a telephone network connection terminates on an Intuity CONVERSANT VIS. Supported telephone connections are Tip/Ring and T1.

Terminal Emulator
Software that allows the VIS 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 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. Text-to-Speech application development is supported through Script Builder.

ThickNet
A 10-millimeter (10BASE5) coaxial cable used to provide InterLAN communications.

ThinNet
A 5-millimeter (10BASE2) coaxial cable used to provide InterLAN communications.

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.

Tip/Ring
A term used to denote analog telecommunications using four-wire media.

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
Comprised of the exchanges between the caller and the voice system. A transaction can involve one or more telephone network connections and voice responses from the Intuity CONVERSANT VIS. It can also involve one or more of the VIS optional features, such as speech recognition, 3270 host interface, FAX response, etc.

transaction state machine process
A multi-channel IRAPI application that runs applications driven by script information.

transient process
A process that is created dynamically only when needed.

troubleshoot
The process of locating and correcting errors in computer programs. This process is also referred to as debugging.

TSM
See “transaction state machine process.”

TTS
See “Text-to-Speech.”

U

UNIX Operating System
A multiuser, multitasking computer operating system developed by the Bell Telephone Laboratories division of AT&T.

UNIX shell
The command language that provides a user interface to the UNIX operating system.

upgrade image tape
A tape, optionally provided to you by the Technical Service Organization, containing the new operating system and Intuity CONVERSANT VIS V5.0 base software in a standard configuration which is compatible with your target system.

upgrade scenario
The particular combination of current hardware, software, application and target hardware, software, applications, etc.

V

vi editor
A screen editor used by the Intuity CONVERSANT VIS to create and change electronic files.

virtual channel
A channel that is not associated with an interface to the telephone network (Tip/Ring, T1, 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 may be initiated only by a “virtual seizure” request to TSM from a DIP.

VIS
See “Voice Information System.”

vocabulary
A collection of words that a VIS is able to recognize using either WholeWord or FlexWord 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 channel
A channel that is associated with an interface to the telephone network (Tip/Ring, T1, or PRI). Any Intuity CONVERSANT VIS application can run on a voice channel. Voice channel applications may 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 data interface process (DIP) or the **soft_szr** command.

Voice Information System
A computer connected to a telephone network that handles touch-tone input, voice response, and line transfer. The Voice Information System uses a screen-based, menu-driven user interface to interact with the system operator or administrator.

voice processing co-marketer
A company licensed to purchase voice processing equipment, such as the Intuity CONVERSANT VIS, to market and sell based on their own marketing strategies.

voice response output process
A software process that transfers digitized speech between system hardware (for example, Tip/Ring and SP cards) and data storage devices (that is, hard disk, etc.)

Voice System Administration
The means by which you are able to administer both voice- and nonvoice-related aspects of the system.

VROP
See “voice response output process.”

W

warning
An admonishment used when there is a possibility of equipment damage.

WholeWord speech recognition
An optional feature based on whole-word technology that provides speaker independence, connected digit recognition, key word spotting, prompt interrupt, and DTMF support functionality. See “whole-word technology.”

whole-word technology
The ability to recognize an entire word, not the phoneme or a part of a word. See “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 identified by a wordlist name. If the wordlist is part of an active vocabulary, the wordlist name appears as a recognition type in the Prompt & Collect mode field.

word spotting
The ability to search past extraneous speech during a recognition.

Index

Numerics

A

B

C

D

E

F

G

H

I

L

M

N

O

P

R

S

T

V

W