Lucent Technologies
Bell Labs Innovations

INTUITY™ CONVERSANT® System

Version 7.0

Speech Development, Processing, and Recognition

585-313-201
Comcode 108231986
Issue 2
April 1998

Notice
Every effort was made to ensure that the information in this book was complete and accurate at the time of printing. However, information is subject to change.

Your Responsibility for Your System's Security
Toll fraud is the unauthorized use of your telecommunications system by an unauthorized party, for example, persons other than your company's employees, agents, subcontractors, or persons working on your company's behalf. Note that there may be a risk of toll fraud associated with your telecommunications system and, if toll fraud occurs, it can result in substantial additional charges for your telecommunications services.

You and your system manager are responsible for the security of your system, such as programming and configuring your equipment to prevent unauthorized use. The system manager is also responsible for reading all installation, instruction, and system administration documents provided with this product in order to fully understand the features that can introduce risk of toll fraud and the steps that can be taken to reduce that risk. Lucent Technologies does not warrant that this product is immune from or will prevent unauthorized use of common-carrier telecommunication services or facilities accessed through or connected to it. Lucent Technologies will not be responsible for any charges that result from such unauthorized use.

Lucent Corporate Security
Whether or not immediate support is required, all toll fraud incidents involving Lucent products or services should be reported to Lucent Corporate Security at 1 800 821-8235. In addition to recording the incident, Lucent Corporate Security is available for consultation on security issues, investigation support, referral to law enforcement agencies, and educational programs.

Lucent Technologies Fraud Intervention
If you *suspect that you are being victimized* by toll fraud and you need technical support or assistance, call the Lucent Technologies National Customer Care Center Toll Fraud Intervention Hotline at 1 800 643-2353.

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

Part 68: Network Registration Number. This equipment is registered with the FCC in accordance with Part 68 of the FCC Rules. It is identified by an FCC registration number.

Part 68: Answer-Supervision Signaling. Allowing this equipment to be operated in a manner that does not provide proper answer-supervision signaling is in violation of Part 68 Rules. This equipment returns answer-supervision signals to the public switched network when:

- Answered by the called station
- Answered by the attendant
- Routed to a recorded announcement that can be administered by the CPE user

This equipment returns answer-supervision signals on all DID calls forwarded back to the public switched telephone network. Permissible exceptions are:

- A call is unanswered
- A busy tone is received
- A reorder tone is received

Canadian Department of Communications (DOC)
Interference Information
This digital apparatus does not exceed the Class A limits for radio noise emissions set out in the radio interference regulations of the Canadian Department of Communications.

Le Présent Appareil Nomérique n'émet pas de bruits radioélectriques dépassant les limites applicables aux appareils numériques de la class A préscrites dans le reglement sur le brouillage radioélectrique édicté par le ministére des Communications du Canada.

Trademarks
See the section titled "About This Book."

Ordering Information

Call:	Lucent Technologies Publications Center	
	Voice 1 800 457-1235	International Voice 317 322-6791
	Fax 1 800 457-1764	International Fax 317 322-6699
Write:	Lucent Technologies Publications Center 2855 N. Franklin Road Indianapolis, IN 46219	
Order:	Document No. 585-313-201 Comcode 108231986 Issue 2, April 1998	

You can be placed on a standing order list for this and other documents you may need. Standing order will enable you to automatically receive updated versions of individual documents or document sets, billed to account information that you provide. For more information on standing orders, or to be put on a list to receive future issues of this document, contact the Lucent Technologies Publications Center.

Warranty
Lucent Technologies provides a limited warranty on this product. Refer to the "Limited Use Software License Agreement" card provided with your package.

European Union Declaration of Conformity
Lucent Technologies Business Communications Systems declares that the equipment specified in this document conforms to the referenced European Union (EU) Directives and Harmonized Standards listed below:

EMC Directive	89/336/EEC
Low-Voltage Directive	73/23/EEC
CE	The "CE" mark affixed to the equipment means that it conforms to the above directives.

Comments
To comment on this document, return the comment card at the back of the document.

Acknowledgment
This document was prepared by Product Documentation, Lucent Technologies, Columbus, OH.

Contents

About This Book

Purpose

Most applications, whether generated with the Script Builder or the Voice@Work application development tool or written using the transaction state machine (TSM) script-level language or C language, involve some form of speech, for example, playing recorded phrases, recognizing speech input, recording spoken phrases, or playing synthesized speech. An application may use one, some, or all forms of speech the INTUITY™ CONVERSANT® system has to offer. This book offers detailed information about the features available for developing speech.

Intended Audience

The primary audience for the Lucent *INTUITY™ CONVERSANT® System Version 7.0 Speech Development, Processing & Recognition* book is as follows:

- End customer developers

  This group is responsible for creating and maintaining applications in the INTUITY CONVERSANT system environment.

- Custom application developers

  This group is responsible for creating applications to be used in the system environment for end-user customers. This audience includes any Lucent Technologies custom application developers.

- Application distributors

  This group distributes and implements applications for end-users. This audience includes independent software vendors (ISVs) and voice processing co-marketers (VPCs).

How This Book Is Organized

This book is contains the following sections:

- Chapter 1, ''Overview of Speech'' — Provides an overview of speech on the INTUITY CONVERSANT system, information about the speech file system and a discussion of all of the speech-related features available for the development of voice response applications.
- Chapter 2, ''Developing Speech'' — Describes the procedures involved in producing speech, from determining the transaction and planning the script to recording and encoding the speech.
- Chapter 3, ''Editing Speech'' — Discusses use of the Graphical Speech Editor (GSE) and features of the Script Builder and Voice@Work application development tools available for editing speech.
- Chapter 4, ''Recognizing WholeWord Speech Input'' — Discusses the concepts behind WholeWord speech recognition and includes sections on creating WholeWord vocabularies and improving the accuracy of speech recognition.
- Chapter 5, ''Recognizing FlexWord™ Speech Input'' — Includes a detailed discussion of FlexWord™ speech recognition and its features.
- Chapter 6, ''Recognizing Dial Pulse Input'' — Discusses the recognition of dial pulses on rotary or push button telephones.
- Chapter 7, ''Putting It Together'' — Discusses how to make the various speech-related features and work well together in a voice response application.
- Appendix A, ''Enhanced Basic Speech Formats'' — Provides complete lists of Enhanced Basic Speech formats for all available languages.
- Appendix B, ''Speech File Formats'' — Includes information about speech files and speech file formats, including pulse code modulation (PCM) and adaptive differential pulse code modulation (ADPCM).
- Appendix C, ''Calculating the OS Index'' — Includes the procedure for computing channel numbers for the GSE.
- Appendix D, ''Text-to-Speech Formats'' — Provides a complete list of the Text-to-Speech (TTS) formats (available for US English only).
- Appendix E, ''Advanced Text-to-Speech Features'' — Discusses how to customize synthesized speech by using escape sequences to add silence delays, change the speaking rate, select the speaking voice gender, and mark text as belonging to a specific text category.
- Appendix F, ''Recognition Post-Processing'' — Describes the data interface process (DIP) associated with WholeWord and FlexWord speech recognition and discusses five processing features that increase the system accuracy of recognition (Return Top 4, Prefix List Check, Luhn Check, Expected Value and Close Match).

- "Glossary" —Defines the terms, abbreviations, and acronyms used in system documentation.
- "Index" — Alphabetically lists the principal subjects covered in the book.

How to Use This Book

This book covers two overall concepts regarding speech on the system:

- Developing speech that prompts callers in an application
- Recognizing speech input or caller input during a transaction

This book is organized in a procedural manner in which you learn about speech on the INTUITY CONVERSANT system (Chapter 1), followed by how to create speech (Chapter 2) and edit speech (Chapter 3). Next, the focus turns to recognizing call input an using speech recognition features (Chapter 4, Chapter 5, and Chapter 6). Finally, you learn how to effectively use speech-related features together (Chapter 7).

Conventions Used in This Book

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

Terminology

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

  Type **y** to continue.

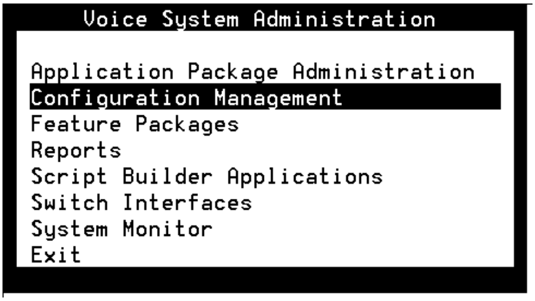
- The word "enter" means to type a value and then press ENTER. For example, an instruction to type the letter "y" and press ENTER is shown as

  Enter **y** to continue.

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

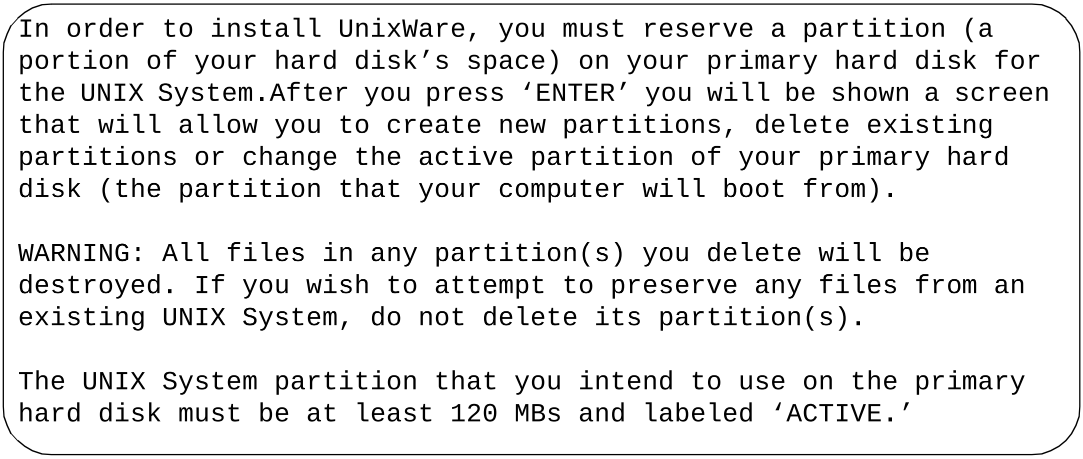
  Select `Start Test.`

- The INTUITY CONVERSANT system displays *menus, screens,* and *windows*. Menus (Figure 1) present options from which you can choose to view another menu, a screen, window. Screens and windows both show (Figure 2 and Figure 3) and request (Figure 4 and Figure 5) system information.



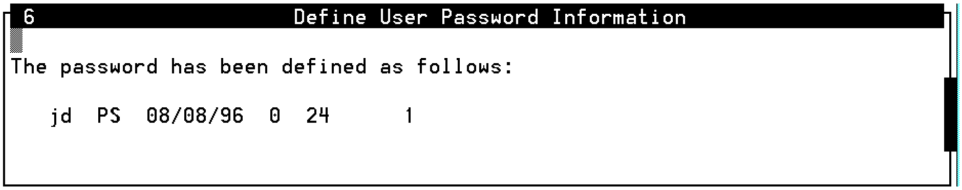
```
Voice System Administration

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

Figure 1. Example of INTUITY CONVERSANT Menu



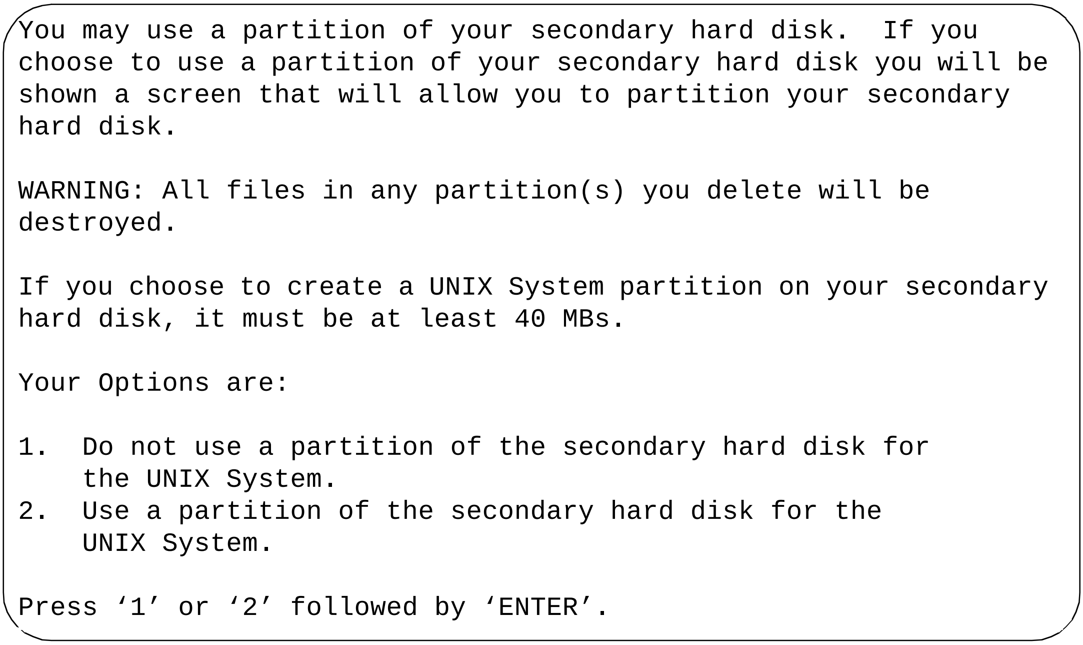
```
In order to install UnixWare, you must reserve a partition (a
portion of your hard disk's space) on your primary hard disk for
the UNIX System.After you press 'ENTER' you will be shown a screen
that will allow you to create new partitions, delete existing
partitions or change the active partition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

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

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

Figure 2. Example of INTUITY CONVERSANT Screen Showing Information



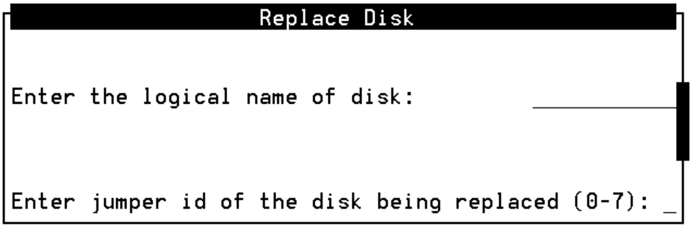
```
6                    Define User Password Information

The password has been defined as follows:

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

Figure 3. Example of INTUITY CONVERSANT Window Showing Information



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

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

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

Your Options are:

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

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

Figure 4. Example of INTUITY CONVERSANT Screen Requesting Information



```
                    Replace Disk

Enter the logical name of disk:          ________


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

Figure 5. Example of INTUITY CONVERSANT Window Requesting Information

Keyboard and Telephone Keypad Representations

- Keys that you press on your terminal or PC are represented as rounded boxes. For example, an instruction to press the enter key is shown as

  Press ENTER.

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

Press ALT D.

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

  Press F3 (Choices).

- Keys that you press on your telephone keypad are represented as square boxes. For example, an instruction to press the first key on your telephone keypad is shown as

  Press 1 to record a message.

Screen Displays

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

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

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

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

  Start at the INTUITY CONVERSANT main menu and select:

  ```
  > Application Configuration Management
        > Database Administration
  ```

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

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

Other Typography

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

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

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

- Command variables are shown in ***bold italic*** type when they are part of what you must type in and *regular italic* type when they are not, for example

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

Safety Labels

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

⚠ **CAUTION:**

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

⚠ **WARNING:**

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

⚠ **DANGER:**

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

Related Resources

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

Training

For information on INTUITY CONVERSANT training, call the BCS Education and Training Center at one of the following numbers:

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

Documentation

The *INTUITY™ CONVERSANT® System Version 7.0 System Description*, 585-313-204, describes in detail all books included in INTUITY CONVERSANT documentation library.

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

The following titles are referenced in this book:

- *INTUITY™ CONVERSANT® System Version 7.0 System Description*, 585-313-204
- *INTUITY™ CONVERSANT® System Version 7.0 New System Installation*, 585-313-106
- *INTUITY™ CONVERSANT® System Version 7.0 Administration, 585-313-501*
- *INTUITY™ CONVERSANT® System Version 7.0 Application Development with Script Builder*, 585-313-206
- *Voice@Work*, 585-313-207
- *INTUITY™ CONVERSANT® System Version 7.0 Application Development with Advanced Methods*, 585-313-203
- The maintenance books for various platforms:
  - — *INTUITY™ CONVERSANT® System Version 7.0 MAP/5P Maintenance*, 585-313-107
  - — *INTUITY™ CONVERSANT® System Version 7.0 MAP/40P Maintenance*, 585-313-108
  - — *INTUITY™ CONVERSANT® System Version 7.0 MAP/100C Maintenance*, 585-313-109
  - — *INTUITY™ CONVERSANT® System Version 7.0 MAP/100P Maintenance*, 585-313-110
- *INTUITY™ CONVERSANT® System Reference*, 585-313-205
- *Application Design Guidelines*, 585-310-670 (available translated from the English into French, Spanish, Brazilian Portuguese, Dutch, Japanese, and German)
- Veritas documentation set, 585-350-906
- Novell UnixWare documentation set, 585-305-908

Additional Suggested Documentation

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

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

See the inside front cover for information on how to order INTUITY CONVERSANT documentation.

Trademarks and Service Marks

Lucent Technologies has made every effort to supply trademark information about company names, products, and services mentioned in the INTUITY CONVERSANT documentation library. Trademarks indicated below were derived from various sources.

- 5ESS, AUDIX, CONVERSANT, DEFINITY, and Voice Power are registered trademarks and INTUITY and FlexWord are trademarks of Lucent Technologies.
- 3M is a trademark of Minnesota Mining and Manufacturing.
- CLEO is a trademark of Interface Systems, Inc.
- Phillips is a registered trademark of the Phillips Screw Co.
- Microsoft, MS, MS-DOS, Internet Explorer, and Excel are registered trademarks and Windows is a trademark of Microsoft Corporation.
- Truevoice is a registered trademark of AT&T.
- UnixWare is a registered trademark of The Santa Cruz Operation, Inc.
- UNIX is a registered trademark of UNIX System Laboratories, Inc.
- Novell 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 and VTAM are registered trademarks of International Business Machines Corporation.
- LINKix is a trademark of CLEO Communications.
- Hayes and Smartmodem are trademarks of Hayes Microcomputer Products, Inc.
- Ethernet is a trademark of Xerox Corporation.
- VERITAS is a trademark of the Veritas Software Corporation.
- Netscape Navigator is a trademark of Netscape Communications Corporation.

Electronic Updates to This Book

The ACCESS Electronic News online bulletin board is available to provide you with additional information about the INTUITY CONVERSANT product, including updates and supplements to the information in this book. This free service is available 24 hours a day, 7 days a week. To register and receive a special offer on ACCESS Plus software, call 1-800-242-6005 and ask for Department 186.

How To Make Comments About This Book

A comment card is included at the back of this book. While we have tried to make this document fit your needs, we are interested in your suggestions for improving it and urge you to complete and return a comment card to us. If the comment card has been removed from this book, send your comments to the address below. Be sure to include the name and document number of this book.

Lucent Technologies
Product Publications
Room 22-2H15
11900 North Pecos Street
Denver, CO 80234-2703 US

Fax: (303)538-1741

Disclaimer

Intellectual property related to this product and registered to AT&T Corporation has been transferred to Lucent Technologies Incorporated. Any reference within this text to American Telephone and Telegraph Corporation or AT&T should be interpreted as references to Lucent Technologies Incorporated. The exception is cross references to books published prior to December 31, 1996, which retain their original AT&T titles.

Lucent Technologies — formed as a result of AT&T's planned restructuring — designs, builds, and delivers a wide range of public and private networks, communications systems and software, consumer and business telephone systems, and microelectronic components. The world-renowned Bell Laboratories is the research and development arm for the company.

Overview of Speech

1

Overview

This chapter provides

- Information about the speech filesystem
- Information about speech terminology
- An overview of the speech-related features available for speech development

Purpose

This chapter is designed to ensure that you

- Understand the software features available for speech and ways in which they can be used
- Recognize speech hardware
- Are familiar with setting up filesystems

Understanding the Speech Filesystem

All speech to be played as part of an application resides as UnixWare files in a mounted UnixWare filesystem. By default, speech filesystems reside in **/voice1**, which is linked to **/home2**. These two filesystems can be used interchangeably. With the INTUITY™ CONVERSANT® system, you can define where in **/voice1** or **/home2** speech files are stored.

Each speech phrase requires a minimum of 8 Kbytes of space. Depending on the encoding rate, an 8-Kbyte block holds different amounts of speech. See Table 1-1 to estimate how much speech is in each speech phrase block.

Table 1-1. Encoding Rates and Speech Phrase Blocks

Coding Rate	Seconds of Speech Per 8-Kbyte Block
SBC16	4.0
SBC24	2.6
ADPCM32	2.0
ADPCM64	1.0
CELP16	4.0

See Appendix B, “Speech File Formats” for detailed information about coding rates. For information about the encoding methods and capacities, see “Speech Storage Capacities” and “Speech Administration Capacities” in Chapter 4, “Features,” of *INTUITY™ CONVERSANT® System Version 7.0 System Description*, 585-313-204.

Default Speech Directory

The default speech directory is designated as **/voice1/vfs/talkfiles**, which is a UnixWare directory for storing speech. It is organized into 8-Kbyte blocks, which allows for quick and efficient retrieval of speech files.

Specifying a Speech Directory

Use the following procedure to change the directory in which speech files are stored from the default (**/voice1/vfs/talkfiles**):

1. Stop the voice system.

   For the procedure to do so, see “Common System Procedures” in *INTUITY™ CONVERSANT® System Reference*, 585-313-205.

2. Access the **/vs/data/irAPI.rc** file.
3. Add the following entry, where *<directory>* is the full path of the new directory where you want to store speech files:

   SPEECHDIR=*<directory>*

   The SPEECHDIR variable specifies the new directory.
4. Restart the voice system.

NOTE:
The speech administration tools (for example, **list**, **add**, **copy**, **erase**, and **vdf**) are available for use only with speech files stored in the speech directory defined by the SPEECHDIR variable in the **irAPI.rc** file.

Saving and Restoring Speech Files

Speech files are backed up when a **mkimage** system backup is performed. The **spsav** (save) and **spres** (restore) commands are also available for backing up speech. See *INTUITY™ CONVERSANT® System Version 7.0 Administration*, 585-313-501, for information about these commands.

CAUTION:
*Use the **spsav** command to copy all speech after you make any changes. This allows you to restore the speech disk(s) during a recovery process with the **spres** command rather than from a **mkimage** tape, which takes longer and may not restore all speech.*

Adding a Second Speech Disk

If you require speech-intensive applications, or if your system has 72 or more channels (telephone network connections), it is recommended that you add a second disk specifically for storing speech.

For instructions to do so, see ”Adding a Hard Disk Drive” in “Replacing a Hard Disk Drive” in the INTUITY CONVERSANT maintenance book for your platform. The second disk in the system is used to store speech filesystems that reside in **/home3**.

Recovering from a Corrupted Speech Disk

For information on recovering from a corrupted speech disk, see “Replacing a Hard Disk Drive” in the INTUITY CONVERSANT maintenance book for your platform.

Changing the Size of the Speech Filesystem

With the Veritas advanced filesystem, you can change the size of the speech filesystem without having to reload the UnixWare system.

For detailed information about increasing and decreasing the size of the filesystem, see “Basic Filesystem Operations” in Chapter 5, “OA&M Menu Interface Operation,” of *Veritas Volume Manager System Administrator’s Guide* in the Veritas documentation set, 585-350-906.

Creating and Removing Filesystems

With the Veritas advanced filesystem, you can create and remove INTUITY CONVERSANT speech filesystems.

For detailed information about creating and removing filesystems, see “Basic Filesystem Operations” in Chapter 5, “OA&M Menu Interface Operation,” of *Veritas Volume Manager System Administrator’s Guide* in the Veritas documentation set, 585-350-906.

Displaying Filesystem Attributes

With the Veritas advanced filesystem, you can display the contents of your filesystems.

For detailed information about displaying the contents of filesystems, see the section “Basic Filesystem Operations, Underlying Speech Concepts,” in Chapter 5, “OA&M Menu Interface Operation,” of *Veritas Volume Manager System Administrator’s Guide* in the Veritas documentation set, 585-350-906.

Defining Phrases

A phrase is a unit of speech, such as letter, number, word, sentence, or paragraph, that is spoken to a caller. Examples of phrases include a welcome message, a bank balance, or the name of a month. Every phrase in an application is identified by a phrase tag or phrase number. A script speaks a phrase to callers by referencing either the phrase tag or the phrase number in the application. See ‘‘Defining Phrase Tags’’ and ‘‘Defining Phrase Numbers’’ below for more information.

Defining Phrase Tags

A phrase tag is a string of up to 50 characters that identifies the contents of a phrase used by an application script. In other words, a phrase tag identifies a specific phrase. When you define a message to be played during a transaction, you specify a given phrase by its phrase tag (as opposed to its content).

The two types of phrase tags are as follows:

- Enhanced Basic Speech
  - Monetary units (such as dollars and cents)
  - Time
  - Weekdays
  - Months
  - Numbers
  - Characters

  The Enhanced Basic Speech package includes prerecorded speech formats corresponding to the above types of phrases. For a list of these formats, see Appendix A, "Enhanced Basic Speech Formats".

  Script Builder and Voice@Work use predefined Enhanced Basic Speech phrase tags for spoken output, such as digits and letters in various inflections. For more information for Script Builder, see Chapter 9, "Speech Administration," in *INTUITY*™ *CONVERSANT*® *Application Development with Script Builder*, 585-310-760. For more information for Voice@Work, see the chapter on variables in *Voice@Work*, 585-313-207.

  NOTE:
  Predefined Enhanced Basic Speech phrase tags begin with a colon (:). Therefore, do not use a colon as the first character in any custom phrase tag.

- Custom speech

  Custom phrase tags are designed specifically for the application you are developing and are usually more than one word in length. Examples:
  - "Your account balance is"
  - "Please enter your account number"
  - "The current interest rate is"

Defining Phrase Numbers

A phrase number is a number that identifies the contents of a phrase used by an application script. A script speaks a phrase to callers by referencing either the phrase tag or the phrase number. A phrase number is assigned to a phrase when you add the phrase to an application (for example, when you add a phrase through a Prompt and Collect action in a Script Builder application or through a Prompt and Collect node in a Voice@Work application).

Defining Talkfiles

A talkfile is a list of phrases usually associated with an application script. Talkfiles are stored under the directory **/speech/talk**. All of these files have a **.pl** extension. The first line in the file shows the talkfile number. The rest of the file displays the phrases (as they were entered in the application) preceded by their phrase numbers. The actual phrases are located in the speech filesystem.

Each talkfile can contain as many as 65,535 phrases. The INTUITY CONVERSANT system can have up to 16,384 talkfiles, although the system is limited by the size of the hard disk. The talkfile number and phrase tag or phrase number together uniquely identify a phrase.

Defining Speech Files

A speech file is a file containing an encoded speech phrase. Speech files can be stored anywhere, although the default speech filesystem is located in the **/voice1/vfs/talkfiles** directory.

Defining the Speech Filesystem

A speech filesystem is a mounted UnixWare filesystem where speech resides and is defined in the **irAPI.rc** file. Only one speech filesystem can be active at a given time. See the various topics under "Understanding the Speech Filesystem" above for more information about INTUITY CONVERSANT filesystems.

Overview of Speech Development Features

Nearly all applications involve playing speech to a caller. The following speech development tools and features are available for the creation, editing, recognition, and inclusion of speech in an application.

- Application development tools
  - Script Builder
  - Voice@Work
- Features
  - Text-to-Speech
  - Graphical Speech Editor (GSE)
  - WholeWord Speech Recognition
  - FlexWord™ Speech Recognition
  - Dial Pulse Recognition
  - Form Filler Plus

Script Builder and Voice@Work Application Development Tools

The Script Builder and Voice@Work application development tools both enable you to design applications that specify every detail of the interaction between the INTUITY CONVERSANT system and its callers. For example, the greeting heard by the caller when connecting with the service, the menu of options offered, the way callers are prompted for credit card numbers and other pertinent information, how long to wait for caller responses, and the relevant databases that need to be accessed are all parts of an application that you can define and implement with either of these tools.

Once an application is designed, you can use these tools to test, generate, transfer, and install it. Speech phrases can be recorded and edited with these applications.

- Script Builder applications are developed through an interface accessed through the INTUITY CONVERSANT main menu,
- Voice@Work applications are developed on a Windows-based PC and then installed on the INTUITY CONVERSANT system.

Reference

- For information about producing speech with Script Builder or Voice@Work, see Chapter 2, ‘‘Developing Speech’’.
- For instructions to install the Script Builder feature software, see “Installing the Script Builder Package” in “Installing the Optional Feature Software” in the INTUITY CONVERSANT maintenance book for your platform. For detailed information on using Script Builder, see *INTUITY™ CONVERSANT® Version 7.0 Application Development with Script Builder*, 585-313-206.
- For Voice@Work hardware requirements and instructions to install the feature software, see *Voice@Work*, 585-313-207.

Text-to-Speech

The Text-to-Speech (TTS) feature allows you to include speech in an application by providing text as input and allowing the speech and signal processing (SSP) circuit card to produce synthesized speech. Currently TTS is available for US English only. The default gender for the speaking voice is male, although a female voice can be specified.

Text that TTS speaks can either be typed at the keyboard or obtained from a file or database. In a Script Builder or Voice@Work application, you can include the text in either an Announce or a Prompt and Collect action step (Script Builder) or node (Voice@Work), or you can use the **tts_file** external function, which directs the system to speck text from an ASCII file. TTS is specified in the

The conversion of text to speech involves many steps, in which TTS does the following:

- Filters the text to identify sentence and phrase boundaries
- Identifies words as a parts of speech (noun, verb, preposition, etc.)
- Expands conventional abbreviations appropriate to the context (for example, “Dr.” can be expanded to “doctor” or “drive,” depending on the context)
- Identifies proper names and addresses for special handling
- Adjusts for inappropriate punctuation (for example, “Dr.” with or without a period is interpreted in the same way)
- Translates non-alphabetic characters (for example, $5 is spoken as “five dollars”)
- Analyzes the text to determine pronunciation, emphasis, timing, and pitch

The synthesized voice can be made more natural sounding in several ways:

- Punctuation marks such as commas, colons, and periods can be used to insert natural pauses in the synthesized voice as well as changes in voice intonation.
- Advanced TTS features can be used to alter the rate of speech and to add intervals of silence that provide emphasis and help increase listener comprehension (see Appendix E, ‘‘Advanced Text-to-Speech Features’’).

Accuracy of the TTS conversion can be increased through the use of advanced TTS features to mark text as belonging to a specific class, such as address, proper name, telephone number, or fraction. Class detectors can then resolve the ambiguity that some text presents (Dr., for example, would be spoken “doctor” in a proper name field, but “drive” in an address field).

TTS can be used to produce text for prompts in an application. Use of TTS during the development of prototype applications eliminates the need to record custom phrases until the application is perfected. But TTS is most often used to speak text that changes frequently, for example:

- Fields from large databases of customer data, such as names, addresses, and order information
- Dynamic text, such as electronic mail

Requirements

The TTS feature requires:

- Text-to-Speech feature software package
- SSP circuit card with TTS functionality assigned

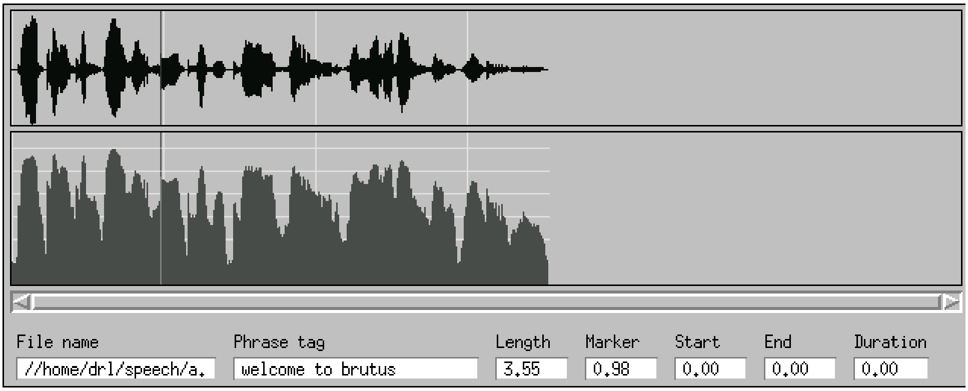
Reference

- To use TTS in ScriptBuilder applications, see "Using Text-to-Speech" in Chapter 8, "Using Optional Features," of *INTUITY™ CONVERSANT® System Version 7.0 Application Development with Script Builder*, 585-313-206.
- To use TTS in Voice@Work applications, see information on inserting controls with the prompt editor in the chapter on prompts in *Voice@Work,* 585-313-207.
- To use TTS with TAS applications, see Appendix B, "Summary of TAS Script Instructions," in *INTUITY™ CONVERSANT® System Version 7.0 Application Development with Advanced Methods*, 585-313-203.
- To install the TT S feature software, see "Installing the Text-to-Speech Package" in "Installing the Optional Feature Software" in the INTUITY CONVERSANT maintenance book for your platform.
- For information on SSP circuit card functions, see "SSP Circuit Card" below.

Graphical Speech Editor

The Graphical Speech Editor (GSE) provides a graphical interface for creating and editing recorded speech files. The interface includes menus that you can access to perform different functions associated with retrieving, creating, editing, and saving recorded speech. Typically, the editor is used to remove silence from the beginnings and endings of recorded speech phrases.

Figure 1-1 is an example of a speech file in the Graphical Speech Editor. Two speech graphs, envelope and energy, represent the digitized audio signal. This visual representation of the speech file simplifies the editing process.



File name	Phrase tag	Length	Marker	Start	End	Duration
//home/drl/speech/a.	welcome to brutus	3.55	0.98	0.00	0.00	0.00

Figure 1-1. Graphical Speech Editor Layout

Requirements

The GSE feature requires:

- Graphical Speech Editor software package
- Tip/Ring card to receive audio input for recording and to output audio speech
- Mouse configured for an appropriate serial port (normally COM2)
- Video controller circuit card configured for the correct number of colors
- Stopping of the voice system

NOTE:
The system cannot take calls when the voice system is stopped.

Reference

- To use the GSE to edit recorded speech, see Chapter 3, ‘‘Editing Speech’’
- For information on system capacities and requirements for the Graphical Speech Editor, see “Graphical Speech Editor” in Chapter 5, “Feature Packages,*”* in *INTUITY*™ *CONVERSANT® System Version 7.0 System Description*, 585-313-204.
- To install the GSE feature software, see “Installing the Graphical Speech Editor Package” in “Installing Optional Feature System Software” in the INTUITY CONVERSANT maintenance book for your platform.
- To administer the mouse, see “Initializing the Mouse” in “Installing Base System Software” in the INTUITY CONVERSANT maintenance book for your platform.
- To configure the video controller circuit card, see “Setting Up the Monitor” in “Installing Base System Software” in the INTUITY CONVERSANT maintenance book for your platform.

Speech Recognition

Speech recognition is an INTUITY CONVERSANT system feature that allows the system to recognize and respond to spoken voice responses from the caller. The system offers both WholeWord and FlexWord speech recognition.

NOTE:
Previous system releases offered speech technologies on a per-system basis. Releases starting with versions 6.0 offer speech technologies on a per-channel basis. The SSP, NGTR, and IVC6 circuit cards include basic playback and coding functions. WholeWord speech recognition and FlexWord speech recognition can be added on a per-channel basis.

WholeWord Speech Recognition

WholeWord speech recognition recognizes entire words, not phonemes or parts of words. However, WholeWord speech recognition is not restricted to recognizing single words. For example, “calling card” can be recognized as a whole word.

WholeWord speech recognition is available in the following languages:

- Australian English
- Brazilian Portuguese
- Canadian French
- Castilian Spanish
- Dutch
- French
- German
- Italian
- Japanese
- Latin-American Spanish
- UK English
- US English

Standard Versus Custom Vocabulary

Lucent Technologies provides a standard WholeWord speech recognition vocabulary that includes:

- The digits 0–9 (and their commonly used synonyms)
- The words “yes” and “no”

The standard WholeWord recognition vocabulary also provides connected-digit recognition.

The standard speech vocabulary is best suited for applications that require callers to respond with “yes” or “no” (such as a survey application) or for number-intensive applications (such as a banking application that requires callers to enter account numbers). You can purchase a custom vocabulary if the standard WholeWord speech recognition vocabulary does not meet your needs.

Requirements

WholeWord speech recognition requires:

- WholeWord speech recognition base package and one or two language package(s)
- SSP circuit card with WholeWord functionality assigned

Reference

- To install the WholeWord feature software, see “Installing the WholeWord Recognition Packages” in “Installing Optional Feature System Software” in the INTUITY CONVERSANT maintenance book for your platform.
- For information on:
  - Using the FlexWord toolkit to create words, wordlists, and vocabularies, see Chapter 5, “Recognizing FlexWord™ Speech Input”
  - SSP circuit card functions, see “SSP Circuit Card’’ below.
  - Using the WholeWord feature in Script Builder applications, see “Using WholeWord Speech Recognition” in Chapter 8, “Using Optional Features,” in *INTUITY™ CONVERSANT® System Version 7.0 Application Development with Script Builder*, 585-313-206
  - Using the WholeWord feature in Voice@Work applications, see the chapter on the application resources tool in *Voice@Work*, 585-313-207

FlexWord Speech Recognition

FlexWord speech recognition recognizes phonemes or parts of words. It is available for use in the following languages:

- Brazilian Portuguese
- French
- German
- Japanese
- Latin-American Spanish
- US English

Because FlexWord recognition provides a cost-effective way of designing large, customized vocabularies and menu options, it is ideal for word-intensive or phrase-intensive applications. For example, a name dialer, an application that allows employees to speak the name of another employee instead of dialing a telephone extension, could be designed using FlexWord speech recognition. The names can be built from existing phoneme models, sparing the expense of custom whole-word data collection processes.

FlexWord Toolkit

The optional FlexWord toolkit allows you to:

- Create FlexWord wordlists and vocabularies
- Add and delete words and wordlists in an existing vocabulary
- Change the phonetic structure of words in an existing vocabulary
- Hear audible playback of the phonemic representation of a word (US English only)

Requirements

FlexWord speech recognition requires:

- FlexWord speech recognition base package and one language package
- SSP circuit card with FlexWord functionality assigned
- Mouse configured for an appropriate serial port— normally COM2 (for the FlexWord toolkit)
- Video controller circuit card configured for the correct number of colors (for editing of wordlist entries in the FlexWord toolkit)
- Tip/Ring circuit card (for playback of wordlist entries in the FlexWord toolkit)

NOTE:
To play back wordlist entries (for US English only) TTS must also be assigned to the SSP circuit card.

Reference

- To install the FlexWord feature software, see “Installing the FlexWord Speech Recognition Package” in “Installing Optional Feature System Software” in the INTUITY CONVERSANT maintenance book for your platform.
- To administer the mouse, see “Initializing the Mouse” in “Installing Base System Software” in the INTUITY CONVERSANT maintenance book for your platform.
- To configure the video controller circuit card, see “Setting Up the Monitor” in “Installing Base System Software” in the INTUITY CONVERSANT maintenance book for your platform.
- For information on SSP circuit card functions, see ‘‘SSP Circuit Card’’ below.
- To use the FlexWord toolkit to create words, wordlists, and vocabularies, see Chapter 5, ‘‘Recognizing FlexWord™ Speech Input’’.

- To use the FlexWord feature in Script Builder applications, see “Using FlexWord Speech Recognition” in Chapter 8, “Using Optional Features,” in *INTUITY™ CONVERSANT® System Version 7.0 Application Development with Script Builder*, 585-313-206.
- To use the FlexWord feature in Voice@Work applications, see the chapter on FlexWord speech recognition in *Voice@Work*, 585-313-207.

“Dial Pulse Recognition

Dial pulse recognition (DPR) allows users with rotary telephones or push-button telephones that generate dial pulses to respond to the INTUITY CONVERSANT system. DPR converts the “pops” and “clicks” on the line to dial pulses. DPR supports digits 0-9 on analog and digital interfaces. For more information on DPR, see Chapter 6, ‘‘Recognizing Dial Pulse Input’’.

Requirements

DPR requires:

- Dial Pulse Recognition software package
- SSP and/or Tip/Ring circuit card

Reference

- To install the DPR feature software, see “Installing the Dial Pulse Recognition Package” in “Installing the Optional Feature Software” in the INTUITY CONVERSANT maintenance book for your platform.
- For information on SSP circuit card functions, see ‘‘SSP Circuit Card’’ below.

Form Filler Plus

The Form Filler Plus feature enables application scripts to record callers’ responses to prompts for later transcription and review. As many as ten responses can be recorded per call session. Caller responses are then stored in the Form Filler Plus database, where they can be retrieved at a later time using the Form Retriever transcription script.

Application

Application voice forms, which prompt for and record caller input for Form Filler Plus, are available through a high-level Script Builder application template provided with the Form Filler Plus package. By copying and modifying this template, you can develop a customized Form Filler Plus application. Alternatively, you can use the **FF_Code** and **FF_Store** action steps provided with the Form Filler Plus package to develop a customized Form Filler Plus application. In Voice@Work, Form Filler Plus is available as an external function.

Whichever of these methods you use, the Form Filler Plus package enables you to select the coding rate that best suits their application needs and access the advanced error-handling capabilities available with this feature, including the ability to store partial messages in the event of error or caller hang up.

Operation

The Form Retriever transcription script provides an easy-to-use one-key operation. Authorized Form Filler Plus transcribers and reviewers can retrieve caller responses by calling the Form Retriever transcription script by using the **Execute** action access code.

The **Execute** action starts a new script on a channel, replacing the script that performed the **Execute** action. The Form Retriever prompts the user for a mode, password, and Script ID. (Transcribe and review mode passwords can be set when the Form Filler Plus package is installed, whereas the Script ID is determined by the script developer). Caller responses are retrieved by Script ID. The transcriber or reviewer can either choose to hear calls for a particular script by entering a Script ID or an asterisk (*) to retrieve the script with the oldest record. Once a script is selected, call records are played from oldest to newest and each response recorded in a call record is played from first to last.

Transcription

Transcribers (using transcribe mode) can play and replay the recorded phrases in a new call record and either delete the record or mark it for review. Once the record is deleted or marked, the next-oldest new call record for the Script ID is played.

Reviewers (using review mode) can play and replay a call record that has previously been marked for review by a transcriber. Reviewers can then delete the record or skip it for later review. Once the record is deleted or skipped, the next-oldest marked call record for the Script ID is played.

Transcribers or reviewers can use an “undo” command to undo the previous delete, mark, or skip command and return to the previous call record. Although there is no limit to the number of transcribers that can use the application at any one time, only one individual can work on a particular call record at a time. The Form Retriever bypasses records that are being played by other users. Transcribers or reviewers can also adjust the pause interval between each phrase that is played from a call record with “slow down” and “speed up” commands.

The number of records deleted or marked for review is announced at the termination of the transcription or review session, enabling Form Filler Plus users to note their activity.

Requirements

The Form Filler Application software package is required.

Reference

To install the Form Filler Plus feature software, see "Installing the Form Filler Application" in "Installing the Optional Feature Software" in the INTUITY CONVERSANT maintenance book for your platform.

To use Form Filler Plus in a:

- Script Builder application, see "Using Form Filler Plus" in Chapter 8, "Using Optional Features with Script Builder," of *INTUITY*™ *CONVERSANT® System Version 7.0 Application Development with Script Builder*, 585-313-206
- Voice@Work application, see the chapter on external functions in *Voice@Work*, 585-313-207

SSP Circuit Card

The speech and signal processor circuit card (SSP) is a high-performance signal processor circuit card capable of simultaneous support for various speech technologies. A single SSP circuit card can run Text-to-Speech, WholeWord speech recognition, FlexWord speech recognition, voice code and play back, and full call classification analysis (CCA) simultaneously.

NOTE:
This card replaces the older companion (CMP) circuit card set, consisting of one SP (signal processor) circuit card plus two CMP circuit cards. The ability of the SSP circuit card to do the work previously done by three circuit cards allows for more available space in the system.

Capacity

Table 1-2 lists the SSP speech channel capacities.

NOTE:
Channel counts assume that the entire SSP circuit card is dedicated to the specified feature. For SSP circuit card channel capacities for features other than those listed above, see "Speech and Signal Processor Card" in Chapter 2, "Hardware," in *INTUITY*™ *CONVERSANT® System Version 7.0 System Description*, 585-313-204.

Table 1-2. SSP Circuit Card Channel Capacities

Feature	Maximum Number of Simultaneous Transactions for SSP Circuit Card
TTS	60
FlexWord speech recognition	15
WholeWord speech recognition without barge-in	15
WholeWord speech recognition with barge-in	10
DPR	120

Reference

- To install an SSP circuit card, see “Installing or Replacing Circuit Cards” in the INTUITY CONVERSANT maintenance book for your platform.
- To assign SSP functionality to an SSP circuit card, see “Assign SP/SSP Functions” in Chapter 3, “Voice System Administration” in *INTUITY™ CONVERSANT® System Version 7.0 Administration*, 585-313-501.

Developing Speech

2

Overview

This chapter describes the following aspects developing speech for a voice application:

- Determining the transaction
- Planning the voice script
- Writing the voice script
- Recording the speech
- Encoding the speech

Purpose

This chapter provides background information necessary to complete the procedures for creating speech on the INTUITY™ CONVERSANT® system, from determining the transaction and planning the script to recording and encoding the speech.

Overview of Creating Speech

Speech processing begins with the creation of encoded and digitized speech files for disk storage. The content of each speech file is a single speech phrase that is spoken at some point in an application dialog.

A speech phrase can consist of a full sentence, a single word, a specified period of silence, music, or even a tone (for example, a “beep”) that is specific to an application. You determine the speech phrase content based on the application requirements.

During a call, the individual speech phrases specified in the application are downloaded by the system from a hard disk drive to either a Tip/Ring circuit card or a speech and signal processor (SSP) circuit card. The circuit card actually plays the speech.

Processing the recorded speech so that the system can access it involves the following activities:

1. Determining the transaction
2. Planning the voice script
3. Writing the voice script
4. Selecting a speech development method
5. Recording speech
6. Digitizing speech phrases
7. Installing speech

Determining the Transaction

The application provides the automated version of the communication between the caller and the agent. The transaction is one component of the application that involves the actual exchanges between the caller and the agent. The transaction is also referred to as the *call flow*.

Before you can begin speech development, you must determine the transaction for the application. It is also a good idea to develop an outline of the application that you want to follow, as well as a general idea of what speech phrases and prompts are necessary. For example, you must decide what type of service you are going to provide, as well as the language and the gender in which the speech will be recorded.

Reference

For information on:

- Planning a voice response application, see *Application Design Guidelines*, 585-310-670 in the language translation of your choice
- Developing an application using Script Builder, see *INTUITY™ CONVERSANT® System Version 7.0 Application Development with Script Builder*, 585-313-206
- Developing an application using Voice@Work, see the chapter on creating an application in *Voice@Work*, 585-313-207
- Developing an application using INTUITY Response Application Programming Interface (IRAPI), the Transaction State Machine (TSM) process script level language, or C language, see *INTUITY™ CONVERSANT® System Version 7.0 Application Development with Advanced Methods,* 585-313-203

Planning the Voice Script

The voice script includes the exact phrases to be recorded, based on the transaction you have determined. The following are suggestions to consider while writing the voice script:

- Track the contents of the voice script by using phrase numbers. Number each phrase in the written voice script.
- Write out every word you expect to be spoken. Edit the voice script to change any poorly written or repetitive phrases. The voice script should be as clear as possible so that a speakers can use it to record phrases.
- Ensure that changes are written into the voice script if changes are made during recording.
- Make all commands short and easy to understand. Users tend to remember only the ends of phrases, so place the needed caller action at the end of a phrase, for example, “For account information, press one.”
- Make prompts clear, but courteous. Remember to welcome users to your company and the system. Thank them at the end.
- Use vocabulary that is understandable and not beyond the scope of your users. For example, do not use computer or programming terminology unless it is familiar to all your users.
- Use the following types of phrases in your voice script:
  - — Long phrases that stand alone, for example,

    > “Welcome to the INTUITY CONVERSANT order entry system.”

    Long phrases are easier to speak for a recording because they stand alone.

— Short phrases that you plan to concatenate, for example,

"Your balance is "
"Press 1"

Typically short phrases include phrases that will be used over and over again.

- Anticipate the environment in which the phrases will be used — that is, whether the phrase will be used at the beginning, in the middle, or at the end of a sentence. The following example shows each use of the word "enter."

  "*Enter* the pound key." (phrase at the beginning of the sentence)

  "You need to *enter* the pound key." (phrase in the middle of the sentence)

  "Please press *enter*." (phrase at the end of the sentence)

  You may plan to use the word "enter" as one phrase, but you need three recordings of this phrase — one phrase with rising inflection, one with medial inflection, and one with falling inflection. Recording words with the proper emphasis is discussed below in "Writing the Voice Script".

- Avoid a long string of adjectives. The following is an example of a poorly designed instruction:

  "Check the 5-digit class schedule number, listed to the left of the specific course, in the course offering schedule book."

- Review the voice script to see if the prompts and responses make sense.

Writing the Voice Script

In writing the voice script, when using a professional speaker, prepare a document that produces the best recordings possible. Mark the target phrases in a way that is easy for the speaker to recognize. Placing quotation marks around the important phrases is helpful. This is called *framing*.

Using Framing in Voice Scripts

Human speech is a continuous, uninterrupted signal. It should not be assumed that you can remove a word from one phrase and place that same word in another phrase that is being recorded for a different use. Individual words that you plan to concatenate must be carefully recorded with the proper inflections and sounds framing them.

To achieve a better recording of short words and phrases, use quotation marks to frame those words you want to emphasize. For example, to achieve accurate recordings of the word "enter," use quotation marks in your voice script as follows so that the speaker concentrates on the word "enter:"

"Enter" the pound sign.

Please press "enter."

The following is an example of a well-prepared voice script that uses framing. The information in quotation marks is the information that the professional speaker should focus on, while the remaining information is the framework.

"Welcome to our telephone information service."

"To learn more about our investment opportunities, press the star key."

"This amount represents" the total balance.

"Please enter" two oh one.

You have "a balance of" two hundred dollars.

You can deposit "up to" five hundred dollars.

Analyzing Speech Inflections

Three types of inflection exist with speech phrases:

- Rising inflection

  Rising inflection is usually used in questions and at the beginning of some words. For example, when you ask, "Can I help you?", the word "you" is spoken with rising inflection.

- Medial inflection

  Medial inflection is usually used in the middle of a word or statement. For example, when you speak the number "302" (as "three oh two"), the "0" is spoken with medial inflection.

- Falling inflection

  Falling inflection is usually used at the end of a word or statement. For example, when you speak "2.0", the "0" is spoken with falling inflection.

NOTE:
Enhanced Basic Speech formats are available with rising, medial, and falling inflections. See Appendix A, "Enhanced Basic Speech Formats".

Placing Frame Words

Place words or phrases before and after the word or phrase that you need recorded, if possible. These phrases should be familiar phrases that guide the speaker into speaking the word or phrase with the correct inflection. For example, if you want an accurate recording of the word "and" with medial inflection, you could record the word "and" in both of the following frames:

Installing "and" verifying

Cutting "and" pasting

You can remove the words that frame “and” later since they are not needed. These frame words are important, though, because the frame words enable a speaker to speak the word “and” in the context necessary to ensure it is concatenated properly when used in a phrase.

NOTE:
The word “and” is part of the Lucent Technologies Enhanced Basic Speech package.

Selecting Speech Sounds for Framing

Words that end with the < *r* > or < *l* > sounds do not make good framing words because those sounds carry over to the next word. In this example,

December “eighth”

“December” is not a good frame word because it ends in an < *r* > sound, which affects the vowel quality of “eighth.” A better frame word is “August,” as follows:

August “eighth”

Including Voiceless Speech Sounds

By contrast, placing a voiceless stop before and after your target word will help achieve an accurate recording. Voiceless stops are sounds like < *p* >, < *t* >, and < *k* >. When a voiceless stop is spoken, the stream of air is blocked and the vocal cords do not vibrate, resulting in a momentary silence. In the example above, the final < *t* > of “August” provides a silence that makes it easy to isolate “eighth.”

Other voiceless sounds useful to end or begin a frame or space are < *f* > and < *s* >.

Selecting a Speech Development Method

As an application developer, you have several options from which to choose for including speech in your application. The options that require you to record speech are:

- Hiring a professional speaker
- Purchasing a custom speech package from Lucent Technologies
- Producing self-recorded custom speech with either the Script Builder or the Voice@Work development tool
- Using the Graphical Speech Editor (GSE)

The options that do not require you to record speech are:

- Using the Enhanced Basic Speech Package
- Importing speech from another application
- Sharing speech already recorded in another application
- Using Text-to-Speech (TTS)

Hiring a Professional Speaker

Hiring a professional speaker, such as an actor or an announcer, gives you recorded speech of high quality. An additional advantage of using a professional speaker is that you may be able to obtain more control and faster response when adding new speech phrases. Consider the following guidelines when choosing a professional speaker:

- Have all phrases prepared for the speaker to read in advance of the recording session. See "Planning the Voice Script" and "Writing the Voice Script" above for guidelines.
- Audition several speakers of both sexes. Record and digitize their voices to evaluate the encoded quality. You may want to listen to several male and female voices to compare the digitized quality.
- Ensure that the speaker is able to maintain the following:
  - — Constant speaking rhythm and general intonation throughout the recording session (this ensures that phrases spoken early in the session result in normal-sounding speech when they are concatenated with phrases spoken later in the session)
  - — Constant acceptable level of volume
  - — Clear pronunciation
  - — Constant orientation and distance from the microphone
- Ensure that alphabetic and numeric characters that are to be recorded with rising, medial, and falling inflections are spoken with appropriate inflection.
- Use the same speaker for all speech associated with a specific application.

If you hire a professional speaker, you can edit the speech phrases for the application script. See Chapter 3, ''Editing Speech'', for more information about editing speech.

See "''Recording Speech'' below for information on setting up a recording session with a professional speaker.

Producing Self-Recorded Custom Speech

Script Builder and Voice@Work both allow you to record speech yourself and store it. You may want to begin with the Enhanced Basic Speech phrases mentioned previously. Again, the standard set includes letters and digits in different speaking inflections and many commonly used phrases, such as the words used to speak dates, times, dollar amounts, etc. You can then use the recording capabilities provided with Script Builder or Voice@Work to record your own speech for phrases unique to your application.

- The Script Builder voice editor supports a speech coding rate of ADPCM32. Recorded speech is stored on the hard drive on the INTUITY CONVERSANT system.
- Speech recorded with Voice@Work default speech editor is saved in **.wav** files. However, when these speech files are installed on the INTUITY CONVERSANT system through Voice@Work, they are converted to ADPCM coding.

NOTE:
The quality of speech recorded through Script Builder and Voice@Work is not as high as the quality produced from professionally recorded speech.

Reference

For detailed information on recording speech with:

- Script Builder, see “The Speech Administration Window” in Chapter 9, “Speech Administration,” of *INTUITY™ CONVERSANT® System 7.0 Application Development with Script Builder*, 585-313-206
- Voice@Work, see the chapter on phrases in *Voice@Work*, 585-313-207

Purchasing the Enhanced Basic Speech Package

You can purchase the professionally recorded Enhanced Basic Speech package from Lucent Technologies. Enhanced Basic Speech provides the following:

- The most commonly used words and phrases, including the letters of the alphabet, pronounced and recorded in rising, falling, and medial inflections
- Essential numbers (“zero” through “twenty,” “thirty,” “forty,” “fifty,” “sixty,” “seventy,” “eighty,” “ninety,” “hundred,” “thousand,” and “million”)
- Days of the week
- Months of the year
- Ordinal numbers 1 through 31 (that is, “1st” through “31st”)
- The words “dollars” and “cents”

The Enhanced Basic Speech package speaks information using a variety of built-in speech formats. For example, if you want the system to speak a number using a money format, you might use number phrases followed by the phrase "dollars and," then the number of cents and the phrase "cents." For a complete listing of the available Enhanced Basic Speech phrases, see Appendix A, ''Enhanced Basic Speech Formats''.

Purchasing a Lucent Technologies Custom Speech Package

You can purchase a professionally recorded custom speech package from Lucent Technologies. You write out the script and Lucent Technologies records and digitizes the speech. Custom speech packages are available with both male and female voices. Custom speech contains phrases designed specifically for the application you are developing. For example, "Thank you for calling Lucent Technologies," is a custom speech phrase.

An advantage of purchasing a custom speech package from Lucent Technologies is that the speakers who record the custom speech phrases are often the same speakers who record the enhanced basic speech phrases. Therefore, a continuity can exist among scripts that use both custom and enhanced basic speech.

NOTE:
Contact your Lucent Technologies service representative if you are interested in purchasing a custom speech package.

Using Text-to-Speech

Using the Text-to-Speech (TTS) package is an option that eliminates the need for recording speech. You enter the phrases to be spoken, and TTS synthesizes the speech. The default speaking voice is male.

NOTE:
TTS is available in US English *only*.

TTS converts text to speech in the following manner:

1. The text is filtered to identify the sentence and phrase boundaries, expand conventional abbreviations, and translate nonalphabetic characters (for example, $5 is translated to "five dollars").
2. Each word is labeled according to part of speech it is (noun, verb, preposition, etc).
3. The text is analyzed to determine pronunciation and emphasis.
4. The text is further analyzed to determine timing and pitch, which is then associated with the pronunciation analysis.
5. The analyzed text is synthesized into speech.

TTS constructs speech by concatenating units of speech. When constructing speech from these units, the TTS feature adjusts parameters, such as pitch and duration, to make the outcome sound natural. Text filtering is critical because it:

- Expands abbreviations appropriate to the context (for example, "Dr." may be expanded to "doctor" or "drive," depending on the context)
- Adjusts for inappropriate punctuation (for example, "Dr" with or without a period is interpreted in the same way)
- Identifies names and addresses for special handling (standard post office abbreviations apply)

TTS functionality is supported through the Script Builder and Voice@Work application development interfaces as well as through TSM script instruction. The talkoff function and other system features for voice response work with TTS as they work with other speech files.

TTS, Enhanced Basic Speech, and prerecorded phrases can be used in the same application.

With some TTS applications, you may need to further customize the use of synthesized speech — for example, by adding silence delays, changing the speaking rate, or marking text as members of a more specific text category. See Appendix E, "Advanced Text-to-Speech Features'', for examples of these methods of customizing.

Reference

For information on using TTS in a:

- Script Builder application, see "Using Text-to-Speech" in Chapter 8, "Using Optional Features," in *INTUITY*™ *CONVERSANT® System Version 7.0 Application Development with Script Builder*, 585-313-206.
- Voice@Work application, see information on inserting controls with the prompt editor in the chapter on prompts in *Voice@Work,* 585-313-20.
- In a TAS application, see Appendix B, "Summary of TAS Script Instructions," in *INTUITY*™ *CONVERSANT® System Version 7.0 Application Development with Advanced Methods*, 585-313-203.

Using the Graphical Speech Editor

The Graphical Speech Editor (GSE) enables you to record and edit speech files for applications on the INTUITY CONVERSANT system. The following GSE options allow you to create and edit speech files designed to be run on system applications:

- Cut and paste
- Copy and paste

- Record
- Output volume

You can also use the GSE to edit speech files recorded through Script Builder or, once the files have been installed on the INTUITY CONVERSANT system, through Voice@Work. New phrases can be added to, changed, or deleted from these prerecorded files. See Chapter 3, ‘‘Editing Speech’’ for more information on editing speech with the various available tools.

Sharing Speech

Sharing speech allows two or more applications to share common speech phrases, only one copy of which exists on your hard disk. If you have more than one application on a system, it will probably be more convenient for you to use the shared speech feature. Sharing speech provides a performance advantage in that shared speech phrases need to be:

- Administered and recorded only once
- Stored only once, allowing you to conserve disk space

Reference

For information sharing speech in a:

- Script Builder application, see Chapter 9, “Speech Administration,” in *INTUITY™ CONVERSANT® System Version 7.0 Application Development with Script Builder*, 585-313-206
- Voice@Work application, see the chapter on sharing application resources in *Voice@Work*, 585-313-207

Importing Speech

You can import speech phrases from other applications and edit them.

- In a Script Builder application, you can import the phrase and edit it with the GSE or Script Builder.

NOTE:
With importing, two copies of the speech exist on the hard disk on the INTUITY CONVERSANT system.

For example, suppose the following phrase is already recorded for another Script Builder application:

“Thank you, please call Bank America again.”

You can import the phrase and edit it using GSE to say, "Thank you, please call again," and include it in your current Script Builder application. If you just want the "thank you" part of the phrase, you can use Script Builder to edit it. Script Builder can edit the beginning and ending of a speech phrase, whereas the GSE can edit any part of a speech phrase.

- For Voice@Work applications, working at the Voice@Work PC, you can use the Phrase Editor to import into Voice@Work the **.wav** file corresponding to the phrase you want. Then you can edit the phrase with the audio editing application installed on your PC.

NOTE:
Depending on the encoding formats that your editing application can accommodate, you may be able to copy files from a Script Builder application on the INTUITY CONVERSANT system onto the Voice@Work PC and from there edit them and import them as **.wav** files into a Voice@Work application.

Reference

For information on:

- Importing speech into one Script Builder application from another, see Chapter 9, "Speech Administration," in *INTUITY™ CONVERSANT® System Version 7.0 Application Development with Script Builder*, 585-313-206
- Importing speech into one Voice@Work application from another, see the chapter on phrases in *Voice@Work*, 585-313-207
- Editing speech, see Chapter 3, ''Editing Speech'', in this book.

Recording Speech

The following methods are available to obtain recorded speech for an application:

- Using a professional speaker
- Purchasing a custom speech package from Lucent Technologies
- Producing self-recorded custom speech with Script Builder or Voice@Work

Using a Professional Speaker

Numerous technical considerations are involved in planning a recording session with a professional speaker and recording the speech on magnetic tape.

Environmental Conditions

A studio specifically designed for recording sessions is necessary when using a professional speaker. It should be noise-free and environmentally-controlled. Arrange for the recording environment to be quiet and acoustically “neutral.” The room should also have soft walls (drapes, carpet, etc.) and be carpeted.

Equipment Specifications

The recommended equipment includes a reel-to-reel tape recorder or high-quality cassette player and amplifier. The reel-to-reel tape should be recorded at 7½ inches/second (19 centimeters/second). Post-processing such as filtering is not required. A video cassette recorder (VCR) with a digital audio processor also produces a high-quality recording.

The recording apparatus and medium should provide the following:

- Dynamic range of at least 50 dB
- Bandwidth from 100 to 8000 Hz
- Flat frequency response in bandwidth
- Low noise insertion

Recording the Spoken Phrases on Magnetic Tape

After the recording session has been planned and the voice script has been created, the professional speaker records the speech on magnetic tape. See “Planning the Voice Script” and “Writing the Voice Script” above for tips and information on planning and writing your voice script prior to the recording stages.

Recording Natural Sounding Speech

The speaker uses the script to record entire sentences on tape so that the speech sounds natural. For example, for a temperature service, the following sentence can be recorded, although only parts of this sentence will be used:

> “The current temperature is” sixty-seven “degrees Fahrenheit.”

When this sentence is encoded, the phrase “The current temperature is” can be encoded as one phrase and “degrees Fahrenheit” can be encoded as a second phrase. The speech phrase “sixty-seven” should be removed because “sixty-seven” is a combination of two phrases that are recorded separately and concatenated later (numbers and alphabetic characters are recorded as separate phrases). See “Recording Alphabetic and Numeric Characters with Inflections” below for tips to record alphabetic and numeric characters.

A stock service might use a sentence similar to the following:

> “The Dow was” “up” “two” “at the close of trading.”

This entire sentence is recorded, but the sentence can be encoded as three separate speech files that are concatenated later. Each of the separately-encoded phrases is shown in quotation marks (" ") above.

Recording Sets of Related Words

When recording sets of related words, such as the days of the week, ordinal numbers, or the months of the year, use a frame sentence in a typical context. A frame sentence for the days of the week might be as follows:

The movie for "[the name of day]" is ____.

NOTE:
Remember that the Enhanced Basic Speech package includes days of the week, ordinal numbers, and months of the year. Therefore, it is not necessary to record these words if you purchase the Enhanced Basic Speech package. See Appendix A, ''Enhanced Basic Speech Formats'', for a complete list of all formats included in the Enhanced Basic Speech package.

During speech editing, the frame words before and after the day of the week are deleted and only the phrase that is inserted in place of "[the name of day]" is saved as a phrase.

The speaker, the studio manager, and a coordinator are usually present at the recording session. The customer for whom the speech is recorded may also be present. During the recording session, these individuals can provide feedback about the necessary inflections for words and phrases and the overall quality of the speech.

Recording Alphabetic and Numeric Characters with Inflections

Record alphabetic and numeric characters with frame words that separate instances of initial, medial, and rising inflections for each letter and number.

Monitor the speaker during this phase of the recording session to ensure that proper inflection is used and that volume and rhythm are constant.

NOTE:
Remember that the Lucent Technologies Enhanced Basic Speech package includes letters and numbers. See Appendix A, ''Enhanced Basic Speech Formats'', for a complete list of all phrases included.

Purchasing a Custom Speech Package from Lucent Technologies

Lucent Technologies can provide you with professionally recorded custom speech through its speech recording service. Lucent Technologies needs the following items to record custom speech phrases:

- A list of phrase tags that need to be recorded
  - In Script Builder, you can use the List function to print all the phrase tags in an application. Unrecorded phrases are marked with an asterisk. See “Displaying Phrase Tags” in Chapter 9, “Speech Administration,” of *INTUITY™ CONVERSANT® System Version 7.0 Application Development with Script Builder*, 585-313-206.
  - In Voice@Work, you can print the contents of the **.pl** (phrase list) file to get a list of phrase tags. See the chapter on the code generation tool in *Voice@Work*, 585-313-207 for information on **.pl** files. The status of the phrases (recorded or unrecorded) is not indicated.
- A list of the complete text of all the phrases that need to be recorded

  NOTE:
  Phrases must be written the exact way they are to be spoken.

  - In Voice@Work, you can use the Print Reports function to print the text of all the phrases in an application. See the chapter on the file menu in Voice@Work 585-313-207 for information on selecting phrases to print.
  - In Script Builder, you cannot print the phrase text, only the phrase tags.
- A clear indication of the phrase tags that corresponds to each phrase
- Specifications of what type of speech encoding should be used

  NOTE:
  The choice of formats depends in part on the amount of speech necessary for the applications on the INTUITY CONVERSANT system. Some coding rates requires more storage space than others. See Table 1-1 in Chapter 1, ‘‘Overview of Speech’’ for a comparison of the storage requirements for various coding rates.
- Indication of whether each phrase is to be recorded in a male or female voice

Contact the Speech Coordinator for the INTUITY CONVERSANT system at the following number for additional information on the custom speech recording service: 614-860-2260

Producing Self-Recorded Custom Speech

You can record speech yourself using the Script Builder application development tool, the GSE, or the Voice@Work application development tool.

Reference

For information on recording speech using:

- Script Builder on the INTUITY CONVERSANT system, see Chapter 9, “Speech Administration,” in *INTUITY™ CONVERSANT® System Version 7.0 Application Development with Script Builder*, 585-313-206
- The GSE on the INTUITY CONVERSANT system, see Chapter 3, “Editing Speech”.
- The audio application installed at the Voice@Work PC, see the chapter on phrases in *Voice@Work*, 585-313-207

Digitizing Speech Phrases

Recorded speech phrases are input from a reel-to-reel tape recorder, from a person over a telephone line, or from an amplifier and microphone. Once the speech phrases are recorded, the speech must be digitized. Speech is digitized by encoding it in an acceptable format. Once the speech phrases are digitized, they are stored as digital data.

The following are ways to digitize speech:

- Use the Graphical Speech Editor.

  If you record speech using the GSE, it is digitized automatically. See “Producing Self-Recorded Custom Speech” above for information about recording speech with the GSE.

- Use the Script Builder application development tool.

  When you record speech using Script Builder, it is digitized automatically in ADPCM32 format. See Chapter 9, “Speech Administration,” in *INTUITY™ CONVERSANT® System Version 7.0 Application Development with Script Builder,* 585-313-206 for more information about recording speech with Script Builder.

- Use the Voice@Work application development tool.

  During the code generation process, speech saved in **.wav** files by the Voice@Work default audio application is digitized in ADPCM32 format. See the chapter on the code generation tool in *Voice@Work,* 585-313-207 for more information.

- Use Lucent Technologies.

  If you want Lucent Technologies to digitize the speech, contact your Lucent Technologies representative for information.

Installing Speech

After the speech has been digitized, it must installed onto the INTUITY CONVERSANT system.

Installing Speech in a Script Builder or TSM Application

To install a Script Builder or TSM application on the INTUITY CONVERSANT system, see “Verifying and Installing the Application” in Chapter 10 “Application Administration,” of *INTUITY™ CONVERSANT® System Version 7.0 Application Development with Script Builder*, 585-313-206. See also “Restoring Speech” in Chapter 9, “Speech Administration,” in the same book.

Installing Speech in a Voice@Work Application

Voice@Work applications must be transferred from the Voice@Work PC to the Intuity CONVERSANT system before they can be installed. To transfer a Voice@Work application, see the chapter on the application transfer tool and to install a Voice@Work application, see the chapter on the SCI tool in *Voice@Work,* 585-313-207.

Installing Speech in an IRAPI Application

If your application was created using C language, use the following procedure to install speech:

1. Insert a diskette containing the speech files into the diskette drive.
2. Go to the appropriate speech filesystem.

   For example, if you are using the default speech filesystem, enter **cd /voice1/vfs/talkfiles**
3. Making sure to use a talkfile number not already in use, enter **mkdir *<talkfile #>***

   For example, if you are using talkfile100, enter **mkdir 100**
4. Copy the speech phrases to the directory specified above.

   When the system prompt is displayed, the speech is loaded onto the hard disk and can be accessed by applications.

NOTE:
If you are adding speech to an existing application, be aware that the system overwrites any existing speech files that have the same name as a file being added.

With an IRAPI application, you can put speech anywhere on your system. For information on accessing speech files in an IRAPI application, see Chapter 6, “IRAPI Programming,” in *INTUITY™ CONVERSANT® System Version 7.0 Application Development with Advanced Methods*, 585-313-203.

Editing Speech

3

Overview

This chapter describes how to use the following tools to edit speech files:

- The Graphical Speech Editor (GSE)
- Script Builder
- Voice@Work

Most frequently, these editors are used to remove unwanted silence from the beginnings and endings of recorded speech phrases. However, the GSE and the audio application provided with the Voice@Work feature enable you to cut and paste segments of speech within the body of a phrase.

Purpose

The purpose of this chapter is to ensure that you understand the capabilities of the Graphical Speech Editor (GSE), Script Builder, and Voice@Work for editing speech files.

Using The Graphical Speech Editor for Editing Speech

The GSE user interface includes windows used to retrieve, edit, and save speech files.

Overview

The GSE is an X-Windows based graphical user interface (GUI) that you access and operate with a mouse and a keyboard. The GSE can play and record speech, and can accept audio speech signals from a microphone, a cassette tape from audio line input, or a telephone, supported through one of the system Tip/Ring circuit cards. These speech signals are digitally stored under appropriate file names for use by INTUITY™ CONVERSANT® applications. The contents of these files can be played back and heard via a telephone.

The GSE allows you to cut, copy, and paste speech segments within a speech file or across multiple speech files and to remove unwanted silence from the beginnings and endings of phrases. It also allows you to change the volume of individual speech segments.

The GSE displays a specific segment of speech that appears as a graphical waveform on the system monitor. You mark an area by moving the mouse over a segment of speech to highlight the speech (waveform) that is needed. You can then cut or copy this highlighted region to a buffer, and subsequently paste it to some other location in the waveform. The waveform displayed on the monitor is the most current result of all editing operations performed on a particular speech file.

Supported Speech Formats

The GSE supports the editing and manipulation of speech recorded in the following formats only:

- ADPCM16
- ADPCM32

The GSE provides the ability to convert user-selected ADPCM formats. The GSE itself requires the PCM64 format. Formats are converted when speech files are opened in the GSE.

The GSE buffer can handle up to 4 minutes of speech, whether the speech consists of one phrase or a series of phrases.

While the GSE can be used to edit speech files from both the Script Builder and Voice@Work applications, typically, you will use the Voice@Work audio application on your PC to edit speech files for Voice@Work.

GSE Software and Hardware

- For a list of the required software and hardware, see "Graphical Speech Editor" in Chapter 1, "Overview of Speech".
- For the procedure to set the video controller circuit card for the monitor, see "Setting the Monitor" in "Installing the Base System Software" in the maintenance book for your platform.
- For the procedure to install the GSE software, see "Installing the Graphical Speech Editor Package" in "Installing Optional Feature System Software" in the INTUITY CONVERSANT maintenance book for your platform.
- For the procedure to initialize the mouse, see "Initializing the Mouse," in "Installing Base System Software" in the Intuity CONVERSANT maintenance book for your platform. Normally, the mouse is configured for the COM2 serial port. For information about the serial mouse port, see "Connecting to Asynchronous Devices" in Chapter 3, "Making Cable Connections and Powering Up the System" of *INTUITY™ CONVERSANT® System Version 7.0 New System Installation*, 585-313-106.

Starting the GSE

Use the following procedure to start the GSE:

1. Log in as **root**.
2. Enter **gse *[-l<chan#>] [-p<playchan#>] [-r<recchan#>]***

   where:

   - *-l* signifies the listen channel (input and output)*.*
   - *-p* signifies the play channel (output only).
   - *-r* signifies the record channel (input only).

   The designated channels must be configured for both input and output to allow both recording and playing speech phrases.

   - The ***-l*** channel is for telephone lines and incorporates both audio input and output.
   - Alternatively, the ***-p*** and ***-r*** channels can be used together to establish audio input/output, where ***-r*** (input) is designated for the microphone, and ***-p*** (output) for the speaker.

   GSE channel numbers are *not* the same as system channel numbers. For the procedure to calculate the channel numbers for the l, p, and r channels, see Appendix C, "Calculating the OS Index". For more information about the **gse** command, see *INTUITY™ CONVERSANT® System Version 7.0 Administration*, 585-313-501.

NOTE:
If the voice system is running, a prompt is displayed asking if it is okay to stop the voice system. The INTUITY CONVERSANT system cannot take calls when the voice system is stopped.

The system displays the GSE window (Figure 3-1).

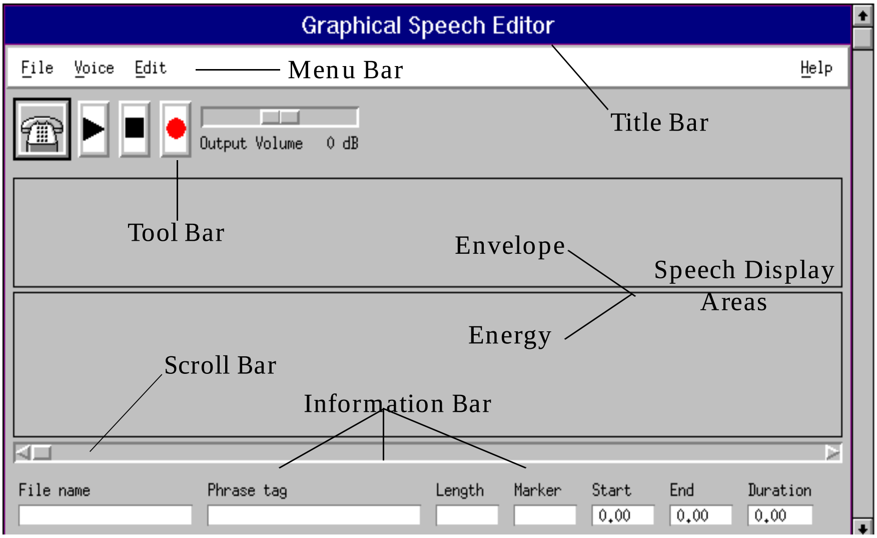


Figure 3-1. GSE Window

Graphical Speech Editor User Interface

The GSE user interface contains the following elements (see Figure 3-1):

- Title bar
- Menu bar
- Tool bar
- Speech display— envelope and energy
- Scroll bar
- Information bar

Title Bar

The title bar (see Figure 3-1) expands across the top of the GSE window to identify the GSE feature.

Menu Bar

The menu bar (see Figure 3-1) enables you to select the File, Voice, Edit, or Help menu.

File Menu

The File menu provides the following options (Table 3-1).

Table 3-1. File Menu Options

Option	**Function**
Open	Accesses an existing speech file. (You must first create and save new files before you can open them.)
Save	Saves your changes after you have edited a speech file.
Save As	Enables you to name and save new speech files.
Exit (or Ctrl+Q)	Closes the GSE and returns you to the system prompt.

Voice Menu

The Voice menu provides the following options (Table 3-2). You can also select all Voice menu items from their corresponding icons on the tool bar.

Table 3-2. Voice Menu Options

Option	**Function**
Hook	Displays a submenu that enables you to select `on-hook` or `off-hook` status. (The telephone receiver (handset) can be either `on-hook` or `off-hook`. To play or record speech, the receiver must be `off-hook`.)
Play	After a speech file has been opened and the telephone receiver is off-hook, plays the message contained in the speech file.
Stop	Ends a recording session or stops a message playback.
Record	Starts a recording session. Typically, you would select `Record` after the telephone receiver is off-hook.

Edit Menu

The Edit menu provides the following options (Table 3-3). Use either the mouse or the corresponding key sequences found under the edit menu to perform any of these editing functions.

Table 3-3. Edit Menu Options

Option	Function
Cut	Removes a selected segment of speech from the speech file so you can either delete it entirely or paste it at a different location.
Copy	Copies a segment of speech into a buffer so you can paste it at different locations in the speech file. (You must select the segment before you can use the Copy function.)
Paste	Pastes at the present location of the marker segments of speech that have either been copied into the buffer or cut. (Use the left and right mouse buttons together to move the marker across the speech display.)
Volume	Sets the volume at the level the speech file is to be heard by the caller.
Undo	Undoes the last change made by any editing feature provided by the GSE.

Help Menu

The Help menu is reserved for future use.

Tool Bar

The tool bar (see Figure 3-1) contains five icons used to perform the basic editing functions of the GSE. Each icon represents a function that you can also perform via the appropriate selection on the menu bar. Table 3-4 describes the functions of the toolbar icons.

Table 3-4. Tool Bar Icons for Editing

Icon	Function
Telephone ☎	Reflects the on-hook (handset on telephone set) or off-hook (handset lifted up) state of the telephone handset. To change the state, click the telephone icon. **NOTE:** The telephone handset *must* be off-hook prior to recording speech or listening to a playback via the Play function.
Play ▶	Plays back recorded speech.
Stop ■	Stops the recording or playing of speech.
Record ●	Starts recording speech.
Output volume Output Volume 0 dB	Adjusts the volume at which a speech file is played and displays the increased or decreased output level, measured in decibels, below the icon. To use this icon, drag the volume button in either direction with the mouse. **NOTE:** Changing the volume in this manner affects *only* speech being played during the present GSE session. To permanently change the volume, you must use the Edit menu.

Speech Display Area

The speech display area represents a graphical image of speech. Approximately 7 seconds of speech are displayed at a time from a speech file that can be up to 4 minutes long. Two windows, envelope and energy, concurrently display separate graphical representations of the same digitized speech file (see Figure 3-1 and Figure 3-2). The envelope and energy representations of the audio signal are essentially the same. Their difference lies in their respective interpretative values to you.

- The envelope is a linear scaled representation of voltage on a line. It shows the sound wave amplitude at different intervals of time. The bubble-like structures in the envelope window (Figure 3-2) are graphical representations of an audio signal oscillating between the positive and negative extremes.
- The energy is the output level of sound behind every phonetic utterance of speech. This energy is vertically measured in increments of 10 dB and is graphically plotted on the display. The energy window displays the amount of energy in the audio signal.

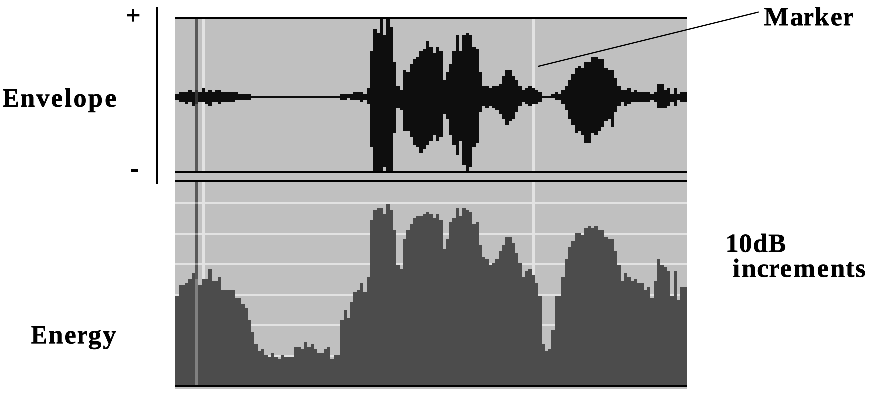


Figure 3-2. Graphical Speech Editor Speech Display

Scroll Bar

The scroll bar (Figure 3-1) enables you to pace through the contents of the speech file by clicking on the end arrows in either direction. A rectangular column within the scroll bar identifies the location of the displayed segment in relation to the total contents of the speech file. The width of this column reflects the proportion of the speech file being displayed.

You can also drag this rectangular column to the approximate location of the speech file you want to access.

Information Bar

The information bar (Figure 3-1) is located at the bottom of the window. It provides information about GSE operations, such as the speech file name, phrase tag, and length of speech. The information bar includes the windows described in Table 3-5.

Table 3-5. Information Bar Windows

Window	Description
File Name	Identifies the name of the speech file.
Phrase Tag	Used for applications developed on Script Builder. The phrase tag is for identification purposes only. See Chapter 1, ''Overview of Speech'', for information about phrase tags.
Length	Identifies the entire length of the speech file in seconds.
Marker	Identifies the position of the marker (vertical cursor) while speech is measured in seconds from the start of the file.
Start	Identifies the starting point of a segment of speech that has been selected for editing, measured in seconds from the start of the file.
End	Identifies the end boundary of a segment of speech that has been selected for editing, measured in seconds from the start of the file.
Duration	Identifies the duration of the selected segment, or the difference between the start and end limits of the segment, measured in seconds.

Editing a Speech File with the GSE

Editing a speech file involves the following GSE functions:

- ''Using the Mouse''
- ''Opening a File''
- ''Setting the Marker''
- ''Selecting a Segment of Speech''
- ''Listening to Speech''
- ''Cutting and Pasting Speech''
- ''Copying and Pasting Speech''
- ''Adjusting the Volume''
- ''Undoing Changes''
- ''Saving Changes''
- ''Exiting''

Using the Mouse

Use the left mouse button to:

- Select menu items and segments of speech
- Modify the length of a selected segment
- Move horizontally in either direction with the scroll bar
- Move vertically up or down with the scroll bar (opening files)

Use the left and right mouse buttons together to:

- Move the marker
- Set a position on the scroll bar

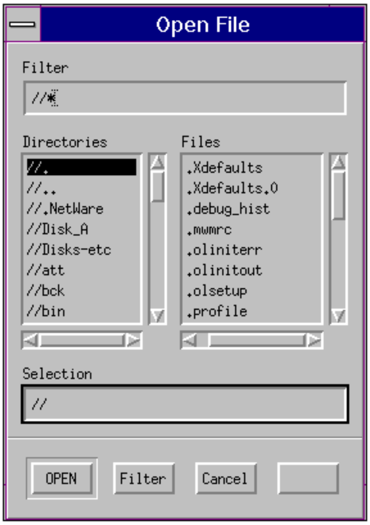
NOTE:
Pressing the two buttons together simulates functionality of the middle mouse button.

Opening a File

Before you can edit an existing speech file, you must open it (bring it into the GSE). Use the following procedure to open a speech file:

1. From the GSE window (Figure 3-1) select **File** from the menu bar.
2. Select **Open**.

   The system displays the Open File window (Figure 3-3).



Filter: //*

Selection: //

Figure 3-3. Open File Window

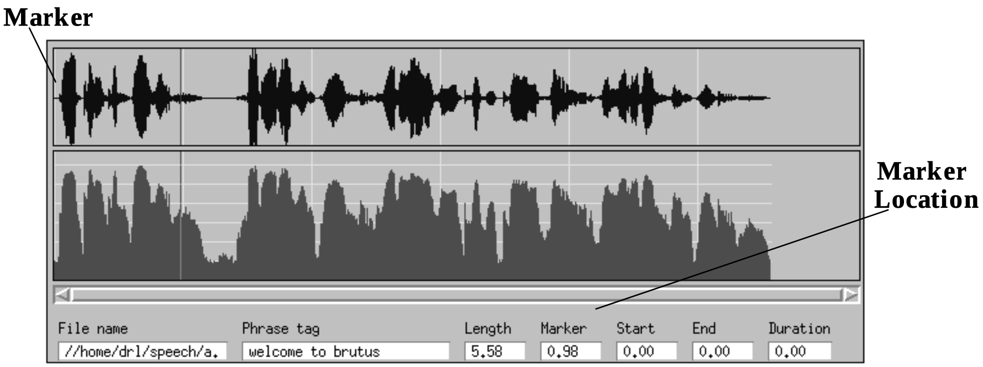
This window provides scrolling tools used to identify files within a specific directory. In addition, the Filter button expands the identification of a path, or pattern, from which a particular speech file can be retrieved.

3. Click Open to open the file or click Cancel to return to the speech editing window.

Setting the Marker

The marker is a light blue vertical line that identifies a particular segment of speech on the speech display. This segment is reflected in the marker section of the information bar at the bottom of the window (Figure 3-4).

To move the marker, press the left and right mouse buttons together. After moving the cursor to the desired location, release the mouse button to position the marker at the current location.



File name	Phrase tag	Length	Marker	Start	End	Duration
//home/drl/speech/a.	welcome to brutus	5.58	0.98	0.00	0.00	0.00

Figure 3-4. Speech Marker

Selecting a Segment of Speech

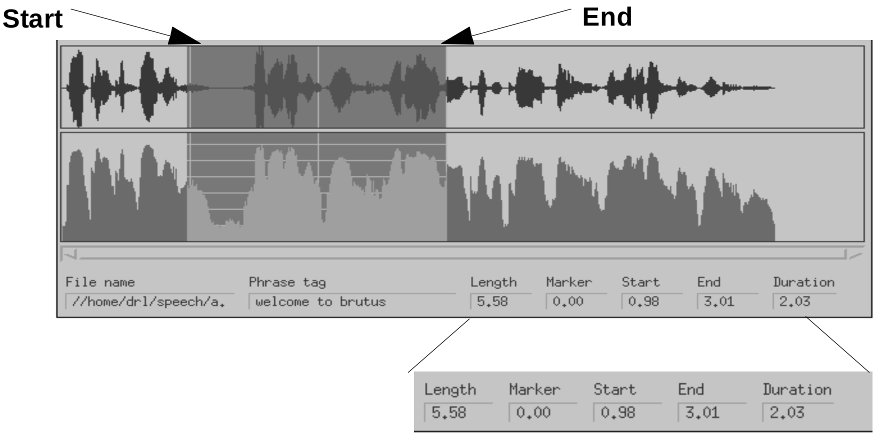
To select a segment of speech, use the following procedure from the speech display area:

1. Press the left mouse button to identify the start position, drag the marker to the desired location, and release the marker to mark the end position.

   The selected segment is highlighted within the identified starting and ending boundaries (Figure 3-5). On the information bar at the bottom of the window the Start and End delimiters of the selected segment are presented in seconds of time (Figure 3-5). The Duration window identifies the total time that the selected segment of speech stretches across.

   NOTE:
   After you have selected a segment of speech, you can modify its length without reselecting it.

2. Press the left mouse button at either end point and drag it to the desired location to modify the length.
3. Use the left mouse button as described above to lengthen the selected segment one window at a time.



File name	Phrase tag	Length	Marker	Start	End	Duration
//home/drl/speech/a.	welcome to brutus	5.58	0.00	0.98	3.01	2.03

Length	Marker	Start	End	Duration
5.58	0.00	0.98	3.01	2.03

Figure 3-5. Selecting Segments of Speech

Listening to Speech

Use the following procedure to listen to a segment of speech:

1. Open a speech file. See "Opening a File" above for the procedure.
2. Make sure the telephone receiver is off-hook.
3. Select a segment of speech. See "Selecting a Segment of Speech" above for the procedure.
4. Select **Play** from the Voice menu to hear the speech file.
5. Click Open.

Cutting and Pasting Speech

The Cut function removes the segment of speech from the file and simultaneously copies it into a buffer. You can then paste the speech into a different part of the file with the Paste function.

NOTE:
You must use the Cut function before the Paste function.

Use the following procedure to cut and paste a segment of speech:

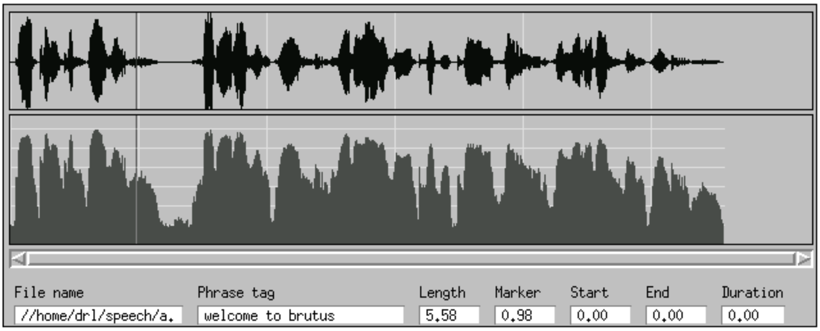
1. From the GSE window (Figure 3-1), select the segment of speech to be removed from its present location. See"Selecting a Segment of Speech" above for the procedure.

   The system displays the specific phrase that you intend to edit.

2. Select **Edit** from the menu bar.

3. Select **Cut**, or press Ctrl T.

   The selected segment of speech is removed from the file. In Figure 3-6, the segment of speech selected in Figure 3-5 is now cut.



File name	Phrase tag	Length	Marker	Start	End	Duration
//home/drl/speech/a.	welcome to brutus	5.58	0.98	0.00	0.00	0.00

Figure 3-6. Cutting Segments of Speech

4. Move the marker to the destination area by using the left and right mouse buttons together.

5. Select **Edit** from the menu bar.

6. Select **Paste**, or press Ctrl P.

   The contents of the buffer are pasted at the identified location of the speech file.

NOTE:
This process can be repeated as long as the speech file does not exceed 4 minutes.

Copying and Pasting Speech

Use the Copy function to copy selected segments of speech to a different location of the speech file. This function copies the selected segment into the buffer without cutting it from its original position. You can then paste the speech into a different part of the file with the Paste function.

NOTE:
You must use the Copy function before the Paste function.

Use the following procedure to copy a segment of speech:

1. From the GSE window (Figure 3-1), select the segment of speech to be copied.
2. Select **Copy** or press Ctrl C.

   The system copies the selected speech segment speech into the buffer. You can now paste the contents of the buffer into different parts of the speech file.
3. Move the marker to the destination area by using the left and right mouse buttons together.
4. Select **Edit** from the menu bar.
5. Select **Paste** from the menu bar or press Ctrl P.

   The contents of the buffer are pasted at the identified location of the speech file.

NOTE:
This process can be repeated as long as the speech file does not exceed 4 minutes.

Adjusting the Volume

The Volume function adjusts the volume at which a caller hears a speech file. Use the following procedure to adjust the volume:

1. From the GSE window (Figure 3-1), select the segment of speech.
2. Select **Edit** from the menu bar.
3. Select **Volume** or press Ctrl V.
4. Use the left mouse button to drag the volume button in either direction.

   The increased or decreased output level, measured in decibels, is reflected below the icon.
5. Click OK to save the volume setting.
6. Click Cancel to return to the speech editing window.

Undoing Changes

You can undo the last change you made to a speech file. Use the following procedure to undo the last change:

1. From the GSE window (Figure 3-1), select **Edit** on the menu bar.
2. Select **Undo** or press Ctrl U.

Saving Changes

Use the following procedure to name and save a new speech file:

1. From the GSE window (Figure 3-1), select **File** on the menu bar.
2. Select **Save As**.

   The system displays the Save As window (Figure 3-7), which provides scrolling tools that help you name files within a specific directory. The Filter button enables you to identify a path or pattern to which a particular speech file is to be saved.

3. Click OK to complete the save, or click Cancel to return to the speech editing window without saving the changes.

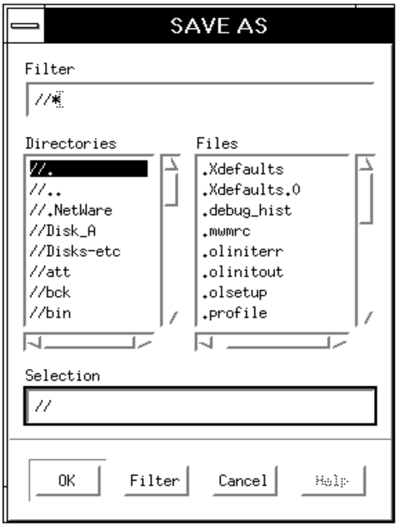


Figure 3-7. Save As Window

Exiting

Use the following procedure to exit the GSE and windows:

1. From the GSE window (Figure 3-1), select **Exit** from the File menu or press Ctrl Q.

   The GSE closes and you are returned to the system prompt.

2. Click and hold down the left mouse button and press Ctrl to exit windows.

NOTE:
If you try to exit before saving changes to a speech file, the system displays a window with a warning message.

Converting Speech Files with the GSE

The INTUITY CONVERSANT system provides utilities to:

- Convert speech stored in the speech filesystem for use with the GSE
- Convert GSE output files and store their contents in the speech filesystem

There are two methods of retrieving speech files between the GSE and the system. One pertains to existing speech pools that support applications. The other pertains to custom phrases. These phrases are stored in system talkfiles.

- The **gse_copypl** command copies phrases from an identified speech pool into UnixWare files that are in the PCM64 format required by the GSE.
- Conversely, the **gse_addpl** command adds (restores) phrases to a specific speech pool from correspondingly named UnixWare files in the GSE format.
- For single phrases, the **gse_copy** command, copies the phrase (identified by a talkfile number and a phrase number) into a UnixWare file.
- Conversely, the **gse_add** command adds (restores) the phrase to the talkfile (identified by the same talkfile number and phrase number).

NOTE:
The **gse_copypl** command uses the **gse_copy** routine to extract individual phrases. The **gse_addpl** command uses the **gse_add** routine to restore individual phrases. For more information all the commands above, see *INTUITY™ CONVERSANT® System Version 7.0 Administration*, 585-313-501.

Retrieving Files from Speech Pools

NOTE:
The voice system must be running to retrieve speech files from speech pools and prepare them for editing.

Use the following procedure to retrieve files from speech pools:

1. Log in as **root**.
2. Ensure that the voice system is running.

   For information on stopping and starting the voice system, see “Common System Procedures in INTUITY™ CONVERSANT® System Reference, 585-313-205.
3. Enter **gse_copypl *<speech pool> <output dir> [<file1>... <fileN>]***

   where:

   - *<speech pool>* is the name of the speech pool from which the speech is to be retrieved.

- *<output dir>* is the name of the directory where the output files are to be put.
- *<file1> <file2>... <fileN>* are optional file names identifying the particular phrase names to be extracted.

The **gse_copypl** command reads the phrase listfile belonging to the speech pool (**/speech/talk/<speech pool>.pl**) to determine the talkfile number, phrase numbers, and output file names for the phrases to be extracted.

- To determine which speech pools a Script Builder application uses, examine the Shared Speech pools parameter. See “Defining Shared Speech” in Chapter 6, “Defining Parameters,” in *INTUITY™ CONVERSANT® System Version 7.0 Application Development with Script Builder*, 585-313-206, for more information.
- To determine which speech pools a Voice@Work application uses, examine the custom speech parameter. See information about globals under the Phrases Manager in the chapter on phrases and information about application resources in the chapter on the application resources tool in *Voice@Work*, 585-313-207.

Output files are placed in the directory <***output dir***>. If no output file names are specified on the command line, all phrases in the speech pool are extracted. If file names are given, only the phrases with those file names that are specified in the phrase listfile are extracted.

Example

In this example, a phrase listfile named **talk3.pl** is stored in the **/speech/talk** directory. The name of the speech pool represented is **talk3** (Table 3-6).

The talkfile number being used by the speech pool (103) is on the first line of the file. Each subsequent line of the file contains a file name, phrase number, and phrase tag (in that order) for each phrase in the speech pool.

Table 3-6. Phrase Listfile (talk3) Format

File Name	Phrase Number	Phrase Tag
103	—	Phrase list for application: talk3
f1000	1000	This is the first Phrase Tag
f1001	1001	This is the second Phrase Tag
f1002	1002	This is the third Phrase Tag

You must first create a directory for talk3.files by entering:

```
cd /speech/talk
mkdir talk3.files
```

Entering **gse_copypl talk3 /speech/talk/talk3.files** extracts phrases 1000, 1001, and 1002 from talkfile 103 and puts them in files f1000, f1001, and f1002, respectively, in the directory **/speech/talk/talk3.files**. These files are then be ready for editing with the GSE.

Entering **gse_copypl talk3 /speech/talk/talk3.files f1001 f1002** extracts only phrases 1001 and 1002 from the talkfile.

Restoring Files to Speech Pools

Use this procedure to o restore speech files to their original format and add them to the appropriate speech pool:

1. Log in as **root**.
2. Enter
   gse_addpl *<speech pool> <input dir> <codetype> [<file1>... <fileN>]*

   where:

   - *<speech pool>* is the name of the speech pool to which the speech is to be added
   - *<input dir>* is the directory location of the GSE edited files
   - *<codetype>* is either “pcm64,” “adpcm16,” or “adpcm32”;
   - *<file1> <file2>... <fileN>* are optional file names identifying the phrase names to be added.

   The **gse_addpl** command reads the phrase listfile in the speech pool (**/speech/talk/<speech pool>.pl**) to determine the talkfile number, phrase numbers, and file names of the phrases to be added.

   - To determine which speech pools a Script Builder application uses, examine the Shared Speech pools parameter. See “Defining Shared Speech” in Chapter 6, “Defining Parameters,” in *INTUITY™ CONVERSANT® System Version 7.0 Application Development with Script Builder*, 585-313-206, for more information.
   - To determine which speech pools a Voice@Work application uses, examine the custom speech parameter. See information about globals under the Phrases Manager in the chapter on phrases and information about application resources in the chapter on the application resources tool in *Voice@Work*, 585-313-207.

   If no file names are specified on the command line, all phrases in the speech pool for which files are found in ***<input dir>*** are added. If file names are given, only the phrases with the particular file names that are specified in the phrase listfile are added.

NOTE:
Script Builder uses a negative phrase number to designate an unrecorded phrase in the phrase listfile. If **gse_addpl** is used to add a previously unrecorded phrase, the phrase number is changed to its positive value to indicate that the phrase exists. Applications should then be verified and installed with the specific speech pool through Script Builder. See "Defining Shared Speech" in Chapter 6, "Defining Parameters," in *INTUITY™ CONVERSANT® System Version 7.0 Application Development with Script Builder*, 585-313-206, for more information about the shared speech parameter in Script Builder.

Example

Consider a phrase listfile named **talk3.pl** stored in the **/speech/talk** directory. The name of the speech pool it represents is **talk3**. The talkfile number being used by the speech pool (103) is on the first line of the file. Each subsequent line of the file contains a file name, phrase number, and phrase tag (in that order) for each phrase in the speech pool.

Therefore, entering **gse_addpl talk3 /speech/talk/talk3.files adpcm32** adds phrases 1000, 1001, and 1002 to talkfile 103 in ADPCM32 format from files f1000, f1001, and f1002, respectively, in the directory **/speech/talk/talk3.files**. These files are not removed by the **gse_addpl** command.

Entering **gse_addpl talk3 /speech/talk/talk3.files pcm64 f1001 f1002** adds phrases 1001 and 1002 only from the talkfile in PCM64 format.

Retrieving Files from the Speech Filesystem

Use this procedure to retrieve speech files from the speech filesystem and prepare the files for editing with GSE:

1. Log in as **root**
2. Ensure that the voice system is running.

   The voice system must be running to extract speech files for editing. For information on stopping and starting the voice system, see "Common System Procedures in INTUITY™ CONVERSANT® System Reference, 585-313-205.
3. Enter **gse_copy *<talkfile#> <phrase#> <outputfile> ["<phrasetag>"]***

   where:

   - *<talkfile#>* is the talkfile number
   - *<phrase#>* is the phrase number
   - *<outputfile>* is the output file
   - *<phrasetag>* is an optional 50-character string that is placed into the GSE voice header of the output file. The GSE displays the tag value when the file is being edited.

NOTE:
You can use the **gse_copy** command in unique situations where you need to edit a phrase not belonging to a speech pool. In this event, you must know the talkfile number and phrase number of the phrase to be edited. If you do not know these numbers, you may be able to determine them as follows:

- If no information is available, enter **list talkfile** for a list of all phrases stored in the speech filesystem sorted by talkfile number. You can determine the talkfile number and phrase number by searching for the phrase tag (the name of the phrase).
- If you know the talkfile number, enter **list talkfile *<tnum>*** for a list of all phrases in the given talkfile, where *<tnum>* is the talkfile number.
- If you know the phrase number, but not the talkfile number, enter **list phrase *<phnum>*** for a list of all talkfiles in which the given phrase number is used, where *<phnum>* is the phrase number.

NOTE:
You must keep track of which extracted files go with what talkfile and phrase in order to put the speech back in the same place after editing. It is suggested that *<talkfile>* be the same as the phrase number and that the directory containing *<talkfile>* be named the same as the talkfile number.

Example

Entering **gse_copy 103 1000 /usr/speech/103/1000** extracts phrase 1000 from talkfile 103 and puts it in the **/usr/speech/103/1000** file for editing by the GSE.

Restoring Files to the Speech Filesystem

Use this procedure to restore speech files to their original speech file:

1. Log in as **root**.
2. Enter **gse_add *<talkfile#> <phrase#> <codetype> <inputfile>***

   where:

   - *<talkfile#>* is the talkfile number
   - *<phrase#>* is the phrase number
   - *<codetype>* is either pcm64, adpcm16, or adpcm3”
   - *<inputfile>* is the input file

NOTE:
You can use the **gse_add** command in situations where you need to add a phrase not belonging to a speech pool. In this event, you must know the talkfile and phrase numbers of the phrases to be added. If you do not know these numbers, you can determine them by using the **list** command as described in “Retrieving Files from the Speech Filesystem” above. Once

the talkfile number and phrase numbers are known, use the **gse_add** command to add the phrase to the talkfile.

Example

Entering **gse_add 103 1000 adpcm32 /usr/speech/103/1000** adds phrase 1000 to talkfile 103 in ADPCM32 format from the file **/usr/speech/103/1000**.

Using Script Builder for Editing Speech

You can edit speech files with Script Builder by trimming off pieces of the recording from the beginning or end, and playing the edited recording. Continue playing, trimming and replaying until the phrase is edited as you want. Restoring already trimmed speech is possible during any given editing session if too much is trimmed. The editing capability in Script Builder does not allow you to cut and paste segments of speech from the body of a phrase.

For detailed information about editing speech files with Script Builder, see "Editing Speech" in Chapter 9, "Speech Administration," of *INTUITY*™ *CONVERSANT*® *System Version 7.0 Application Development with Script Builder*, 585-313-206.

Using Voice@Work for Editing Speech

You can edit speech files with Voice@Work by using the default audio application provided with the Voice@Work feature or with a customer-provided audio application installed on your Voice@Work PC. The default audio application can edit **.wav** files.

To edit speech files with the Voice@Work audio application, see information on the Phrase Editor in the chapter on phrases in *Voice@Work*, 585-313-207.

Recognizing WholeWord Speech Input

4

Overview

This chapter describes the capabilities of WholeWord speech recognition and the factors that influence the accuracy of recognition. Topics covered include:

- Languages supported
- Bilingual capabilities
- Standard vocabulary
- Speech recognition types
- Barge-in
- Positive and negative influences on speech recognition accuracy
- The process by which WholeWord speech recognition works

Purpose

This chapter provides background information necessary to use the WholeWord speech recognition package for optimal accuracy.

Introduction to WholeWord Speech Recognition

WholeWord speech recognition is available as an optional installable package on the INTUITY™ CONVERSANT® system. WholeWord speech recognition allows you to write applications that prompt for and understand spoken input from callers. This feature supports a standard vocabulary that includes:

- The numbers “zero” through “nine” and their commonly used synonyms
- The words “yes” and “no”

WholeWord Hardware and Software

- For a list of the required software and hardware, see “WholeWord Speech Recognition” in Chapter 1, “Overview of Speech”.
- For the procedure to install the WholeWord software, see “Installing the WholeWord Recognition Package” in ”Installing Optional Feature System Software” in the INTUITY CONVERSANT maintenance book for your platform.

Languages Supported

WholeWord speech recognition provides standard vocabulary support for the following languages.

- Australian English
- Brazilian Portuguese
- Canadian French
- Castilian Spanish
- Dutch
- French
- German
- Italian
- Japanese
- Latin-American Spanish
- UK English
- US English

For additional information about global support, contact your Lucent Technologies representative.

Bilingual Speech Recognition Capabilities

The functionality of the SSP circuit card allows you to load and operate any two WholeWord speech recognition language packages on a single system simultaneously. The bilingual speech recognition capability is available within the same call, but not within the same prompt. For example, an initial prompt can be used to determine which language the caller prefers to use. However, a custom grammar has the ability to combine bilingual recognition capabilities in the same prompt.

Standard Vocabulary

Table 4-1 lists the standard vocabulary for languages supported in WholeWord speech recognition.

Table 4-1. Standard Vocabulary for WholeWord Speech Recognition

Caller Input Field	US English, UK English, and Australian English	Canadian French and French	Latin-American Spanish, and Castilian Spanish	Japanese	German	Dutch	Brazilian Portuguese	Italian
0	zero, oh	zéro	cero	zero	null	nul	zero	zero
1	one	un, une	uno	ichi	eins	een	um	uno
2	two	deux	dos	ni	zwei, zwo	twee	dois	duo
3	three	trois	tres	san	drei	drie	três	tre
4	four	quatre	cuatro	shi, yon	vier	vier	quatro	quattro
5	five	cinq	cinco	go	fünf	vijf	cinco	cinque
6	six	six	seis	roku	sechs	zes	meia, seis	sei
7	seven	sept	siete	shichi, nana	sieben	zeven	sete	sette
8	eight	huit	ocho	hachi	acht	acht	oito	otto
9	nine	neuf	nueve	ku, kyu	neun	negen	nove	novo
yes	yes	oui	sí	hai	ja	ja	sim	sì
no	no	non	no	iie	nein	nee	não	no

Standard WholeWord Speech Recognition Types

Recognition types allow you to indicate which keyword or words are possible selections for each prompt. All of the recognition types and words used by an application constitute that application's vocabulary. A complete recognition type includes:

- Mathematical models of the words
- A grammar
- Specifications of maximum and minimum numbers of digits
- A recognition data interface process (DIP) — optional (discussed below)

The standard WholeWord speech recognition package provides several commonly used recognition types. The recognition type is selected:

- In Script Builder applications in the Recog*:* field on Page 2 of the Define Prompt & Collect screen (see "Using Dial Pulse Recognition or Speech Recognition" in Chapter 8, "Using Optional Features," of *INTUITY™ CONVERSANT® System Version 7.0 Application Development with Script Builder*, 585-313-206, for more information).
- In Voice@Work applications, through the Response tab of the Prompt and Collect, Announce, Menu, Automenu and Voice Capture nodes (see the chapters on working with nodes in *Voice@Work*, 585-313-207, for more information).

Models and Speaker In dependence

Each word of the system's vocabulary is represented by one or more mathematical models that contain the speech signal characteristics of the word. The speech recognition process compares a person's voice to the set of predeveloped speech models. Each model is constructed from thousands of samples of the spoken word. Regional accents and dialects associated with a particular language are incorporated into each model, as well as a mix of male and female speakers so that the system recognizes callers of either gender speaking in any of those dialects with any of those accents. The recognition type determines which models the incoming speech is compared to.

Grammars

A grammar, which is selected by a recognition type, is a set of rules that specifies allowable vocabulary words and vocabulary word combinations at any one point in the script (for example, "four," "five," "six," "no"). While collecting spoken input, the speech recognition algorithm uses models and grammars to generate a list of candidates that most closely resemble this spoken input. The algorithm returns the most likely match to the script or, if no match fits, rejects the input. When input is rejected, the algorithm returns a "?" or an empty string to the script. This return message is the same for all of the languages supported. All grammars provided in the speech recognition package share the same set of models for words that are common to their package's vocabulary.

Keyword Recognition Types

Keyword recognition types are used for isolated word recognition, that is, responses of a single word or digit. Table 4-2 shows the standard WholeWord speech recognition types, which are used when prompting the caller to speak a response of one word or digit. The US_1_3 recognition type, for example, means that callers are expected to say a single word, either "one," "two," *or* "three." A prompt that might use the 1_3 digit grammar type could be structured as follows: "For checking account balance, say 'one'. For savings account balance, say 'two'. For interest rates, say 'three'."

Table 4-2. Standard WholeWord Speech Recognition Types

Language	"Yes" or "No"	One digit 1_3	One digit 1_3 or "no"	One digit 1_5	One digit 1_5 or "no"	One digit 0_9 or "oh"
Australian English	AU_YN	AU_1_3	AU_1_3N	AU_1_5	AU_1_5N	AU_DIG
Brazilian Portuguese	BP_YN	BP_1_3	BP_1_3N	BP_1_5	BP_1_5N	BP_DIG
Canadian French	CF_YN	CF_1_3	CF_1_3N	CF_1_5	CF_1_5N	CF_DIG
Castilian Spanish	CS_YN	CS_1_3	CS_1_3N	CS_1_5	CS_1_5N	CS_DIG
Dutch	NL_YN	NL_1_3	NL_1_3N	NL_1_5	NL_1_5N	NL_DIG
French	FR_YN	FR_1_3	FR_1_3N	FR_1_5	FR_1_5N	FR_DIG
German	GR_YN	GR_1_3	GR_1_3N	GR_1_5	GR_1_5N	GR_DIG
Italian	IT_YN	IT_1_3	IT_1_3N	IT_1_5	IT_1_5N	IT_DIG
Japanese	JN_YN	JN_1_3	JN_1_3N	JN_1_5	JN_1_5N	JN_DIG
Latin-American Spanish	MS_YN	MS_1_3	MS_1_3N	MS_1_5	MS_1_5N	MS_DIG
UK English	UK_YN	UK_1_3	UK_1_3N	UK_1_5	UK_1_5N	UK_DIG
US English	US_YN	US_1_3	US_1_3N	US_1_5	US_1_5N	US_DIG

Connected-Digit Recognition Types

Connected-digit recognition types are used for connected-word recognition, that is, responses of more than one word or more than one digit string. Table 4-3 lists the connected-word recognition types. The spoken string can be fixed in length from 1 to 10 digits, or it can be of a variable length. Grammars are provided for the 1- to10-digit fixed-length and the 1- to 24-digit variable-length strings.

NOTE:
Languages other than US English do not allow strings of greater than 10 digits. With US English, use of a custom grammar is recommended.

Table 4-3. WholeWord Speech Connected-Digit Recognition Types

Language	Fixed Length 1-10 Digits or Variable Length 0-24 digits (US English only)
Australian English	AU_DIG
Brazilian Portuguese	BP_DIG
Canadian French	CF_DIG
Castilian Spanish	CS_DIG
Dutch	NL_DIG
French	FR_DIG
German	GR_DIG
Italian	IT_DIG
Japanese	JN_DIG
Latin-American Spanish	MS_DIG
UK English	UK_DIG
US English	US_DIG

For better accuracy, you must specify the desired string length when selecting the recognition type for a fixed-length string. An area code (614), for example, consists of three words; therefore, the minimum and maximum values are “3.” If a variable-length string (US English *only)* of one to four words is required, “1” is the minimum value and “4” is the maximum value.

The minimum and maximum number of digits in the string are specified:

- In Script Builder applications in the `Min Number of Digits:` and `Max Number of Digits:` fields on Page 2 of the Define Prompt & Collect screen (see Chapter 7, "Defining the Transaction," in *INTUITY™ CONVERSANT® System Version 7.0 Application Development with Script Builder*, 585-313-206, for more information).
- In Voice@Work applications, through the `Minimum Length` and `Maximum Length` fields in the Response tab of the Prompt and Collect node (see the chapter on node descriptions in Voice@Work, 585-313-207, for more information).

Connected-Digit Recognition

Connected digits are strings of naturally spoken digits, which may or may not include a pause. All packages recognize spoken digit strings. Connected-digit recognition packages allow you to collect input by specifying a fixed-length (1–10 digits) or, for US English *only,* a variable-length (1–24 digits). Number entry is an essential component if you want callers to input numbers with more than one digit, (for example, "1, 3, 5" or "2, 4, 6, 8"). Recognition is better for fixed-length strings than for variable-length strings.

Fixed-Length versus Variable-Length Connected Digits

The capability to recognize fixed-length strings of 1–10 digits is provided with WholeWord speech recognition packages (see Table 4-3). A fixed-length string is a string of digits that is always made up of the same number of digits. For example, a US social security number is a fixed-length number, since it always consists of nine digits. By contrast, a variable-length string indicates a string of digits that varies. For example, a street address is a variable-length string.

NOTE:
The variable-length string is available for US English *only*.

Accuracy is best when you use fixed-length strings. Whenever possible, try to specify the length of the string to increase recognition accuracy. For strings longer than 10 digits, use of a custom grammar is recommended (see "Custom Grammars" below). Using the provided variable-length grammar as an alternative results in decreased accuracy. If your application must accept a variable-length digit string, you may be able to increase recognition accuracy by using a two-step entry process. First prompt the caller for the number of digits, and then prompt for the digits. For example:

"How many digits are in the next code?"
"Please say the three-digit code now."

The script can then select a fixed-length recognition type for the appropriate number of digits.

Recognition DIPs

A recognition data interface process (DIP) is used to help increase an application’s accuracy. For example, certain digit strings, such as credit card numbers, have check digits built into them. Verifying a check digit is done most efficiently through a DIP. However, a DIP is not necessary for many recognition types. If there is no further information to consider for a string other than what is in the grammar, a DIP is not needed. Specifically, none of the standard WholeWord speech recognition types, such as “yes” and “no,” needs a DIP.

For some recognition types, there may be tradeoffs that determine how much of the structure of the input is built into the grammar and how much should be left for the DIP. For more information about custom DIPs, see “Writing the DIP” in Chapter 4, “Data Interface Process,” in *INTUITY*™ *CONVERSANT*® *System Version 7.0 Application Development with Advanced Methods*, 585-313-203. Also see ‘‘Custom DIPs’’ below. For more information on post-processing, see Appendix F, ‘‘Recognition Post-Processing’’.

Word Spotting

WholeWord speech recognition supports word spotting. Word spotting is the ability of the system to ignore extraneous speech during speech recognition. For example, if a caller says, “I want number five, please,” the system recognizes the word “five” as a valid response and ignores the rest of the input. In other words, callers do not have to speak the key word, which in this case, is “five” in isolation. The caller can speak other words, and the system can distinguish the key word from the extraneous words.

This means that the caller’s input is not limited to the words in the system vocabulary. For example, in the phrase “Uh, yes please,” the vocabulary word “yes” is recognized if the system is using the standard vocabulary.

However, to maintain good recognition accuracy it is advisable to structure the application to prompt the caller for the required information only. Applications should encourage the caller to speak only what is required.

Phrase Screening

Phrase screening is the ability of the system to determine whether or not a candidate key word is a close enough match to be declared a valid key word. For example, if the prompt states, “Please respond with ‘yes’ or ‘no’,” and the caller speaks the word “what,” the system informs the caller that the response is invalid. It could then replay the prompt, “Please respond with ‘yes’ or ‘no’.”

However, to maintain good recognition accuracy it is advisable to structure the application to prompt the caller for the required information only. Applications should encourage the caller to speak only what is required.

Recognition Confirmation

Recognition confirmation is the ability of the system to repeat the caller's spoken response back to the customer for confirmation. For example, a script prompts a caller, "Please say your area code," and the caller says, "Six, one, four." Using recognition confirmation, the system repeats what it recognizes the caller's response to be and then prompts, "If this is correct, say 'yes', if not say 'no'." The caller then confirms whether the system recognized the spoken input properly.

For more information regarding the recognition confirmation:

- In Script Builder applications, see "Specifying the Confirm Action" in Chapter 7, "Defining the Transaction," in *INTUITY™ CONVERSANT® System Version 7.0 Application Development with Script Builder,* 585-313-206
- In Voice@Work applications, see information on the `Confirmation Input Mode` field in the Confirmation tab of the Prompt and Collect node in the chapter on node descriptions in *Voice@Work,* 585-313-207.

Barge-In

Barge-in, also called "recognize during prompt," is the ability of the system to allow callers to interrupt or barge in during voice playback by speaking a vocabulary word. Speech recognition accepts either speech or touchtone input in response to a prompt. Barge-in operates for speech much like the talkoff option does for touch-tone input, where a caller can interrupt the prompt by pressing a touch tone. See "Dual Tone Multifrequency (DTMF) Support" below for more information on interrupts.

Experienced callers appreciate being able to shorten the transaction time by not being required to listen completely to each prompt. You can enable or disable barge-in for any of the prompts in your application. For more information about the talkoff option for touchtone input, see *Application Design Guidelines,* 585-310-670.

NOTE:
The system is able to detect touch tones immediately; however, it does take a few seconds longer to detect valid speech, as opposed to a cough, sneeze, etc. You should expect a slight delay when using barge-in.

For isolated word recognition, the prompt does not stop until the system recognizes a valid vocabulary word. Once the prompt completes playback, the initial timeout field eventually ends the recognition if no valid input is received.

For packages that support connected-digit recognition, the playback of the prompt stops between the recognition of the first and last word of the input, when the system decides that valid input has started.

Enable or Disable

At the beginning of an application, after the answer instruction, you can place an **SR_Prompt** instruction to enable barge-in. If a barge-in resource is available, it is reserved for this call. Otherwise the return code to **SR_Prompt** is negative, and barge-in will not be available. For information about using **SR_Prompt** in:

- Script Builder applications, see “Defining the SR_Prompt External Action” in Chapter 8, “Using Optional Features,” of *INTUITY*™ *CONVERSANT*® *System Version 7.0 Application Development with Script Builder*, 585-313-206
- *Voice@Work* applications, see the chapter on standard external functions in Voice@Work, 585-313-207.

You can also enable or disable barge-in a Prompt and Collect action (assuming barge-in has been globally enabled with a **SR_Prompt** instruction) so a prompt can be interrupted by voice or touch tones. Enabling barge-in for prompts in:

- Script Builder applications is done by setting the `Speak with Interrupt` field set “yes” on page 1 of the Define Prompt & Collect screen (see Chapter 8, “Using Optional Features,” in *INTUITY*™ *CONVERSANT*® *System Version 7.0 Application Development with Script Builder*, 585-313-206, for more information)
- Voice@Work applications is done by setting the `Allow Interrupt` field to “true” in the Response tab of the Prompt and Collect node, the Automenu node, or the Announce node (see the chapter on node descriptions in *Voice@Work*, 585-313-207, for more information)

If your application does allow barge-in, be sure your prompts are worded so that callers know they cannot speak until the prompt is finished. For example, look at the wording and the <pause> length of the prompts in “Discouraging Barge-In” below. There are no large gaps of silence to tempt callers to speak barge in.

Discouraging Barge-In

The following example includes short pauses that tell the caller that the system is waiting for a response. The caller answers at the time of the short pause.

> “For sales say, ‘one’.’”<short pause>
> “For service, say, ‘two’.”<short pause>
> “To speak with a representative, say ‘three’.”

The following example does not include pauses. The caller waits until the prompt is completed before answering.

> “You may order up to five copies. Please say how may copies you want.”

NOTE:
If there is silence at the end of a prompt, the caller may speak but the system may not be prepared to listen. Make sure that there is no silence at the end of your recorded phrases when barge-in is disabled.

Encouraging Barge-In

The following prompts are designed to encourage callers to barge in when barge-in is enabled. Note that the pauses are longer than in the examples in "Discouraging Barge-In" above, so callers are encouraged to respond after hearing the desired option and the word "now."

"For sales say, 'one' now." <pause of 1-1.5 seconds>
"For service, say, 'two' now." <pause of 1-1.5 seconds>
"For a representative, say 'three' now." <pause of 1-1.5 seconds>

System Response for Barge-In

For an application using speech recognition with barge-in, callers who choose to talk over the prompt message will not experience barge-in until most of the digit string is spoken. The distinction in response is as follows:

- When a fixed-length digit recognition type is used on the SSP circuit card, the prompt can be interrupted after *N*-2 digits are spoken plus a system delay of 0.75 second, where *N* is the total number of digits to be collected. For example, for the US_9DIG recognition type, the prompt can be interrupted after seven digits.
- When short prompts are used, callers may not detect delays in the barge-in point because the prompt will frequently complete before the caller has spoken much input.

Dual Tone Multifrequency (DTMF) Support

Even with WholeWord speech recognition installed, callers still have the option of responding to prompts with touch tones rather than speaking. Speech recognition accepts either speech or touchtone input in response to a prompt. Talkoff is the ability of the system to allow callers to interrupt during voice playback by pressing a DTMF touch tone. Talkoff operates for touch tones much like barge-in does for speech, where a caller can interrupt a prompt by speaking.

Often the response time for talkoff is faster than for barge-in. This is because touch tones were designed to be recognized by machines and are rarely confused with other sound or voice input. However, WholeWord speech recognition must perform the more complex task of separating the desired input from other sound or voice input that may simultaneously be present. The system does not terminate a prompt until it processes enough input to ensure with relatively high confidence that the input is valid. For a prompt requiring single-digit input, talkoff and barge-in may appear equally fast since the input is completed with a single touch tone or a single spoken digit. Prompts for multidigit input reveal the difference between the response times for talkoff and barge-in. See ''Barge-In'' above for information comparing and contrasting talkoff and barge-in.

Custom Grammars

NOTE:
Custom grammars are not currently supported for Voice@Work applications.

Recognition types are the choices associated with the `Recog:` field on the Script Builder Define Prompt & Collect screen. The system uses the recognition type, along with other fields on the Define Prompt & Collect screen, to select a speech recognition grammar to be used for that prompt. Certain digit strings and custom vocabulary words may require custom recognition types and associated custom grammars for acceptable recognition accuracy. Examples include a credit card number or a merchant number. These digit strings have specific limitations on the position of certain digits within the string. For example, a telephone area code in North America requires a number 2 through 9 for the first digit, 0 through 9 for the second digit, and 0 through 9 for the third digit.

Custom recognition types help limit the recognition possibilities for the SSP (speech and signal processing) circuit card, which results in better accuracy. For the SSP circuit card, there is no restriction on the number of grammars that can fit on one card on any system.

For more information, contact your Lucent Technologies representative.

Custom Vocabulary

Lucent Technologies can create a custom vocabulary to supply your application with speech recognition software for words not included in the standard package. For example, the vocabulary might include the words “checking” and “savings.” With a custom vocabulary, the application could prompt the following:

> “For your checking account balance, say ‘checking’ or to hear your savings account balance, say ‘savings’.”

A custom vocabulary requires application analysis by Lucent Technologies, speech data collection, model creation, and custom grammar work. You may want to consider FlexWord™ speech recognition as a more practical alternative. See Chapter 5, “Recognizing FlexWord™ Speech Input”.

For more information, contact your Lucent Technologies representative.

Custom DIPs

You can improve recognition accuracy by using an application with a DIP. DIPs, which are typically written in the C programming language, interact with your script to help access external information. Once a request is received from a transaction state machine (TSM) script, for example, the DIP processes the message and returns the results to the corresponding script. DIPs usually work based on knowledge that is unavailable to the SSP circuit card.

WholeWord Speech Recognition Accuracy

WholeWord speech recognition accuracy depends not only on the recognition algorithms, but also on the models, grammars, DIPs, prompt structure, calling environment, user behavior, and the recognized data itself. Each of these factors can impact recognition accuracy positively or negatively. Also, measures of accuracy must be based across the entire calling population. Therefore, any attempt to measure accuracy must include a statistically representative sample of the calling population.

Positive Influences on WholeWord Speech Recognition Accuracy

The items described below have a positive impact on WholeWord speech recognition accuracy.

Isolated Word Recognition

Isolated word recognition is very high. The smaller the number of choices in an isolated word recognition type, the better the accuracy. For example, US_1_3 is more accurate than US_1_5, which in turn is more accurate than using US_DIG with length 1.

Fixed-Length Digit String

For connected-digit recognition, a fixed-length recognition type provides better accuracy than a variable-length recognition type. If possible, avoid the use of variable-length strings in WholeWord speech recognition applications.

Validation of Data

Try to verify the recognized result against a database or a host field. This helps improve the overall accuracy of an application, especially when a longer string is input. For more information on improving accuracy of an application, see "Return Top 4 (RT4)" in Appendix F, "Recognition Post-Processing".

Reprompt

If the keyword is not spoken, and the system does not misinterpret extraneous words for a keyword, the system can reprompt the caller. If the accuracy measurement is based on either a WholeWord or FlexWord speech recognition application with a confirmation and reprompt step, the accuracy increases.

Prompt Structure

The prompt structure can greatly affect accuracy by promoting a clearly articulated response, helping the caller to barge in at the appropriate time or to wait until the prompt is complete before talking (when barge-in is disabled), and providing consistent instructions on what the caller should say to get the desired result.

- Menu prompts

  For best results, build menu prompts with the following structure:

  > <*desired result*> <*action required*>

  Examples:

  > "To hear your checking account balance, say 1."
  > "To hear your savings account balance, say 2."

  By speaking the action required at the end of the prompt, the caller does not have to remember what is required of him or her through the description of the desired result. In addition, if you want to encourage your callers to barge in when they hear their desired result, you can add a small pause after the action-required phrase.

- Yes and no prompts

  Structure yes and no prompts as yes and no questions. For example:

  > "Would you like to hear your order again?"

  If the caller does not respond to the prompt, the follow-up prompt could be as follows:

  > "Would you like to hear your order again? Please say 'yes' or 'no'."

  This wording is more natural than the following:

  > "To hear your order again, say 'yes'. Otherwise, say 'no'."

To encourage the use of barge-in, add a small pause (about 1.5 seconds) following the action required phrase. For example:

> “Would you like to hear your order again? <pause>
> Please say ‘yes’ or ‘no’.”

Calling Experience and Informative Prompts

In an application where the calling population is closed and callers are experienced or trained to use the application, recognition accuracy improves.

Lengthy prompts that provide detailed instructions on how to respond may improve accuracy, but are generally unacceptable unless the application has infrequent users. Users who interact with system prompts infrequently (for example, once or twice a year) are more willing to listen to a lengthy prompt than those who do so frequently.

Custom Grammars and DIPS

Custom grammars improve the recognizer’s ability to “score” the candidate by selectively limiting the recognition possibilities. The recognizer assigns a score to each input based on closeness of match to the models for the selected grammar. Custom DIPS help further process the recognition result with information unavailable to the recognizer.

Negative Influences on WholeWord Speech Recognition Accuracy

The items described below have a negative impact on WholeWord speech recognition accuracy.

Environment

A very noisy environment, such as an airport or train station, can cause recognition accuracy problems. In certain cases, speech data can be collected to build custom word models based on the noisy environment to improve recognition accuracy.

Extraneous Words within Responses

The system can sometimes misinterpret extra words spoken alongside the keyword if they have the same characteristics as the key word.

Information Type

Attempting to recognize data not normally spoken in the form of the digits “0” through “9” adversely affect accuracy. For example, dollar amounts and days of the month are not usually spoken in digit form “0” through “9”. To speak the date December 15, the caller would have to say “1-2-1-5.” Training callers to speak information in this format can increase application accuracy. However, if callers speak natural numbers, such as “fifteen,” speech recognition will not work.

Strong Regional or National Accents or Dialects

Although WholeWord speech recognition is based on thousands of speech samples per word, the system can still misinterpret strong accents or dialects.

Connected-Digit String Length

Connected-digit string recognition can be thought of as a sequence of single-digit recognitions performed as one operation. For example, assume that the per-digit accuracy is X% and that a digit string of one digit will be correct X% of the time. Taking into consideration that this is a probabilistic, exponential model, when longer digit strings are used the overall expected accuracy will be X^n%. Therefore, a 2-digit string will have an overall expected accuracy of X^2% and a 10-digit string will have an overall accuracy of X^{10}%. As a result, string accuracies are affected by the length of the string. Shorter string lengths are more accurate than longer string lengths. In addition, individual digit accuracies, as well as overall string accuracies, vary according to the language and noise conditions of different national networks.

Connected-digit string accuracy can be maximized in various ways:

- Accuracy is always better for shorter strings than longer strings.
- Fixed-length strings are more accurate than variable length since the recognizer knows to look for “X” number of digits.
- With custom programming, it is possible to further improve accuracy of an application by having the recognizer return a list of possible strings. When these can be validated against external information such as comparing potential account number strings against a database of valid account numbers, the correct string can frequently be chosen.
- The recognizer can also be given a custom digit string grammar that can guide the recognizer when the digit string must conform to specific digit sequence rules. To obtain custom grammars, contact your Lucent Technologies representative.

For a WholeWord speech recognition string of digits, the per-digit accuracy is comparable to isolated word recognition. However, the accuracy of the whole string is lower than the per-digit accuracy, and steadily decreases as more digits are added.

Other Limitations

The capability of the system and WholeWord speech recognition is application dependent. If the system is under-engineered for a particular application, it may not perform satisfactorily. Several application-related factors can affect the number of channels available for speech recognition.

Specific application-related factors that affect the number of supported WholeWord speech recognition channels include:

- The percentage of time spent recognizing speech input
- The percentage of callers who use touch-tone entries, which require far fewer hardware and software resources
- The number of simultaneous speech recognition calls expected
- The use of the barge-in with WholeWord speech recognition, which increases the hardware and software resources required to process each transaction

How WholeWord Speech Recognition Works

Processing involved during WholeWord speech recognition on the INTUITY CONVERSANT system includes a caller response, the action of the recognizer, and processing.

Caller Response

During a Prompt and Collect action, the system prompts the caller for a response. If the Prompt and Collect action allows for voice input, the system locates and reserves a free recognizer resource. If the caller responds using touch tones, resources of the SSP circuit card are not required for this particular prompt and the resource is freed. If the caller responds using voice input, the input is sampled at 8000 samples per second, digitized, and then transferred to the SSP circuit card via a time division multiplexed (TDM) bus. The TDM bus provides a communication link between the SSP circuit card and the telephone interface cards (T1 or Tip/Ring) and also allows SSP circuit card resources to be shared across all incoming voice channels. This makes available more channels of recognition, since there is no need to dedicate certain incoming channels to only WholeWord speech recognition when accepting both touch tone and voice input. Thus channels can share resources.

The Recognizer

The SSP circuit card is loaded with the WholeWord speech recognition software, also called the “recognizer.” The recognizer compares the incoming speech sample to the word models indicated by the recognition type chosen in the

Prompt and Collect action. The recognizer computes a score for each of the models. The score represents the likelihood that the incoming speech matches a word model indicated by the recognition type. The recognizer returns the most likely candidate to the variable specified:

- For Script Builder applications in the `Caller Input Field`. on page 2 of the Define Prompt & Collect screen (see Chapter 8, "Using Optional Features," in *INTUITY™ CONVERSANT® System Version 7.0 Application Development with Script Builder*, 585-313-206, for more information)
- For Voice@Work applications in the `Save Response In` field in the Prompt and Collect, Automenu, and Menu nodes (see the chapter on node descriptions in *Voice@Work*, 585-313-207, for more information)

For the following example the variable is $CI_VALUE for a Script Builder application and `[CollectedDigits]` for a Voice@Work application). If none of the candidates sufficiently match, the input is rejected and a "?" or an empty string value is returned to the script. For example, for a prompt of "1, 2, or 3" the application developer selects the US_1_3 recognition type, which is used for a response consisting of "1", "2", or "3." The caller's voice response to the prompt is compared to a model of the numbers "1", "2", and "3," and a score is computed for each number. Whichever score is highest determines the recognized result. A value of "1", "2, or "3" is returned in $CI_VALUE (or `[CollectedDigits]`). If no value scores high enough, the recognizer returns a "?"as the value. The application then decides what to do next, based on the value of the field. For strings of digits, the recognizer first determines up to four of the top scoring strings and their resulting scores. The recognition DIP then returns to the script the top scoring string whose length falls between minimum and maximum number of digits specified in the application. The string is then stored in $CI_VALUE (or `[Collected Digits]`).

Computational Processing

The SSP circuit card has six of the latest-generation digital signal processors. A single SSP circuit card has the ability to run Text-to-Speech (TTS), speech recognition, dial pulse recognition (DPR), voice code and playback, and full call classification analysis (CCA) simultaneously. In addition to these features, the SSP allows for available space in the system while consuming little power.

The SSP circuit card uses only 19 watts per circuit card. For more information on the SSP channel capacities, see "Capacity" in Chapter 1, "Overview of Speech".

Further Processing

You can improve recognition accuracy by performing post-processing on the recognition candidates using the **recog_dip**. See Appendix 4, "Recognizing WholeWord Speech Input", for further information on **recog_dip** and other recognition post-processing functions.

Recognizing FlexWord™ Speech Input

5

Overview

This chapter provides:

- Background information about the capabilities of FlexWord™ speech recognition
- Guidelines for designing FlexWord wordlists for use in applications
- Procedures for creating, modifying, installing, activating, and deactivating FlexWord wordlists
- Information on positive and negative influences on the accuracy of FlexWord speech recognition

Purpose

This chapter provides the information necessary to create FlexWord wordlists and use them in applications.

Introduction

The FlexWord speech recognition package can be used to recognize specific words spoken by callers from a vocabulary that you define to suit your application.

- A word is any phrase that can be recognized with FlexWord speech recognition. A word must be associated with a wordlist.
- A wordlist is a set of words that can be recognized with FlexWord speech recognition by a Prompt and Collect action in a Script Builder or Voice@Work application.
- A vocabulary is a set of wordlists associated with a particular FlexWord speech recognition application.

Languages Supported

FlexWord speech recognition supports the following languages:

- Brazilian Portuguese
- French
- German
- Japanese
- Latin-American Spanish
- US English

For INTUITY CONVERSANT system Version 7.0, the speech and signal processor (SSP) circuit card is required for all languages.

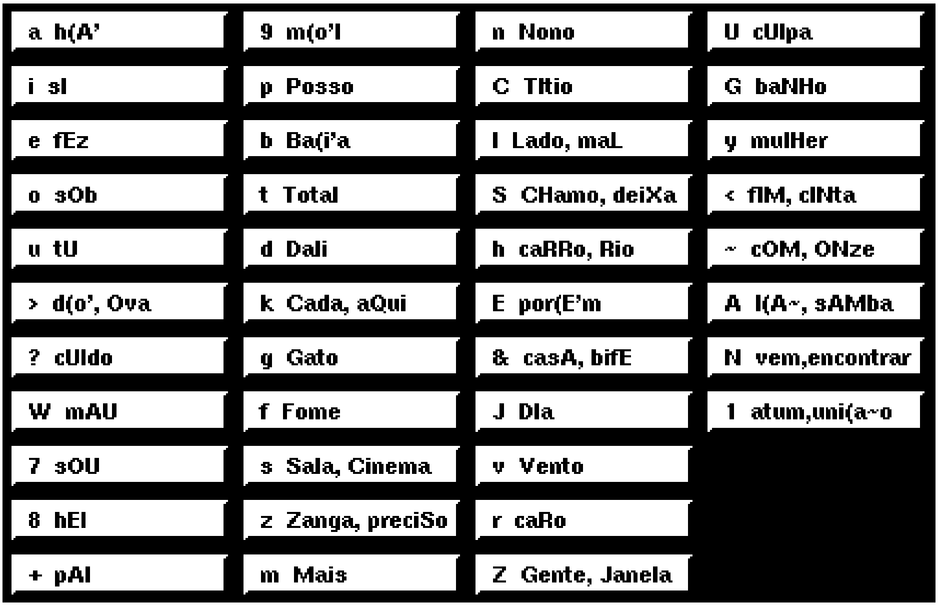
NOTE:
FlexWord speech recognition can run one language per system only.

Phonemes

FlexWord speech recognition uses sub-word technology. Sub-word technology relies on phonemic recognition for analyzing and recognizing words.

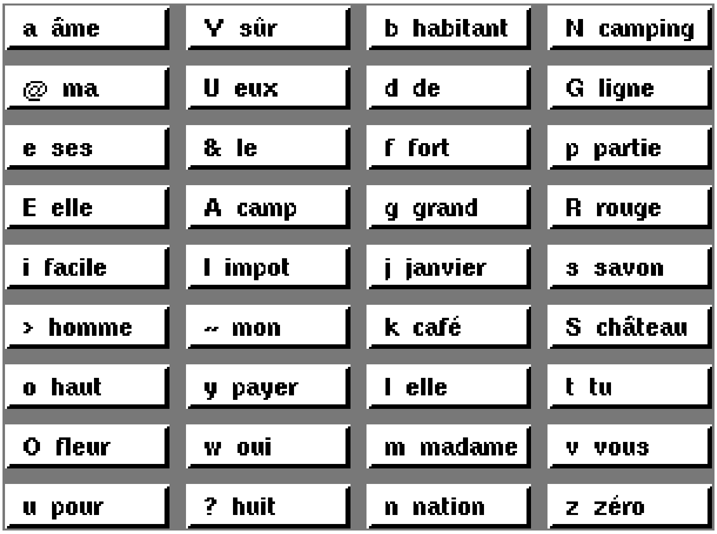
Phonemes are units of sound that form recognizable words when strung together in a particular order. The English language contains 40 phonemes that represent all basic sounds used in the language. The word “sales,” for example, consists of four phonemes: s-A-l-z. Other languages use different phonemes.

Illustrated in Figure 5-1 through Figure 5-6 are FlexWord speech recognition phonemes for the supported languages. A word accompanies each phoneme to illustrate its sound.



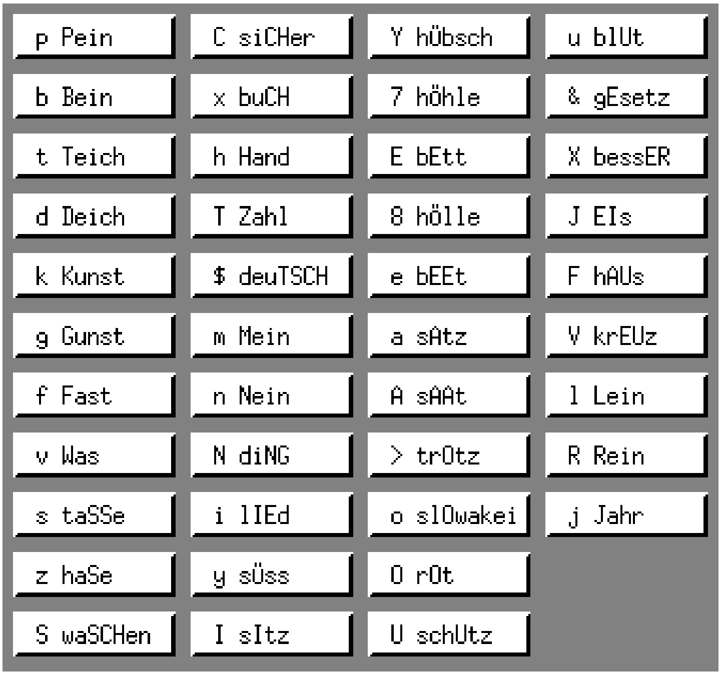
a h(A'	9 m(o'l	n Nono	U cUlpa
i sI	p Posso	C TItio	G baNHo
e fEz	b Ba(i'a	l Lado, maL	y mulHer
o sOb	t Total	S CHamo, deiXa	< fIM, cINta
u tU	d Dali	h caRRo, Rio	~ cOM, ONze
> d(o', Ova	k Cada, aQui	E por(E'm	A l(A~, sAMba
? cUIdo	g Gato	& casA, bifE	N vem,encontrar
W mAU	f Fome	J DIa	1 atum,uni(a~o
7 sOU	s Sala, Cinema	v Vento	
8 hEI	z Zanga, preciSo	r caRo	
+ pAI	m Mais	Z Gente, Janela	

Figure 5-1. Brazilian Portuguese Phonemes



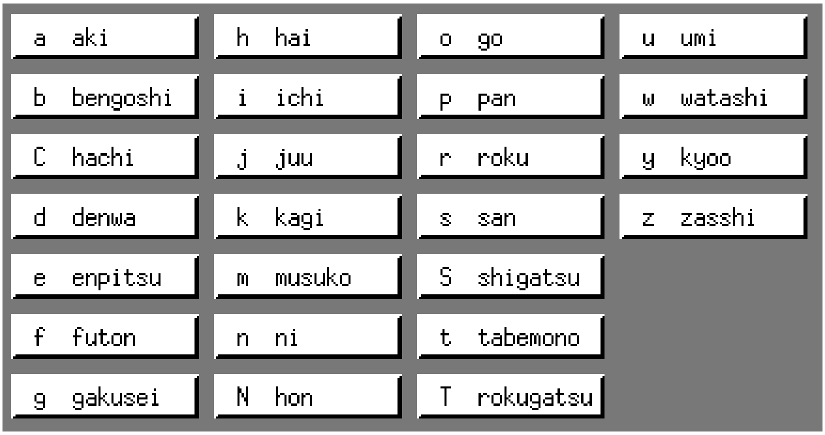
a âme	Y sûr	b habitant	N camping
@ ma	U eux	d de	G ligne
e ses	& le	f fort	p partie
E elle	A camp	g grand	R rouge
i facile	I impot	j janvier	s savon
> homme	~ mon	k café	S château
o haut	y payer	l elle	t tu
O fleur	w oui	m madame	v vous
u pour	? huit	n nation	z zéro

Figure 5-2. French Phonemes



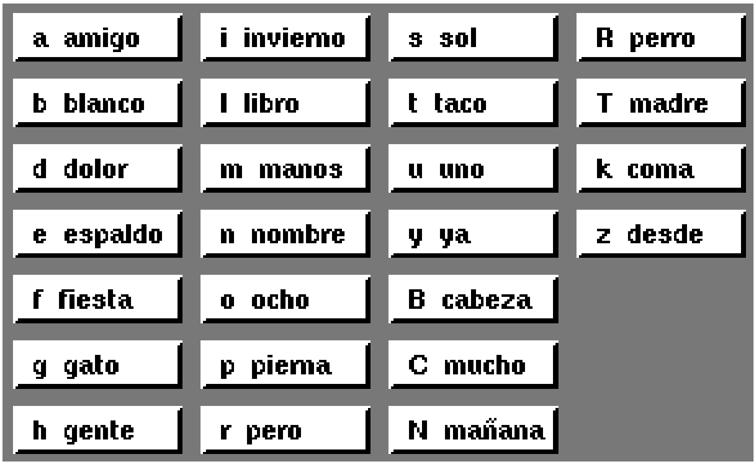
p Pein	C siCHer	Y hübsch	u blUt
b Bein	x buCH	7 höhle	& gEsetz
t Teich	h Hand	E bEtt	X bessER
d Deich	T Zahl	8 hölle	J EIs
k Kunst	$ deuTSCH	e bEEt	F hAUs
g Gunst	m Mein	a sAtz	V krEUz
f Fast	n Nein	A sAAt	l Lein
v Was	N diNG	> trOtz	R Rein
s taSSe	i lIEd	o slOwakei	j Jahr
z haSe	y sÜss	O rOt	
S waSCHen	I sItz	U schUtz	

Figure 5-3. German Phonemes



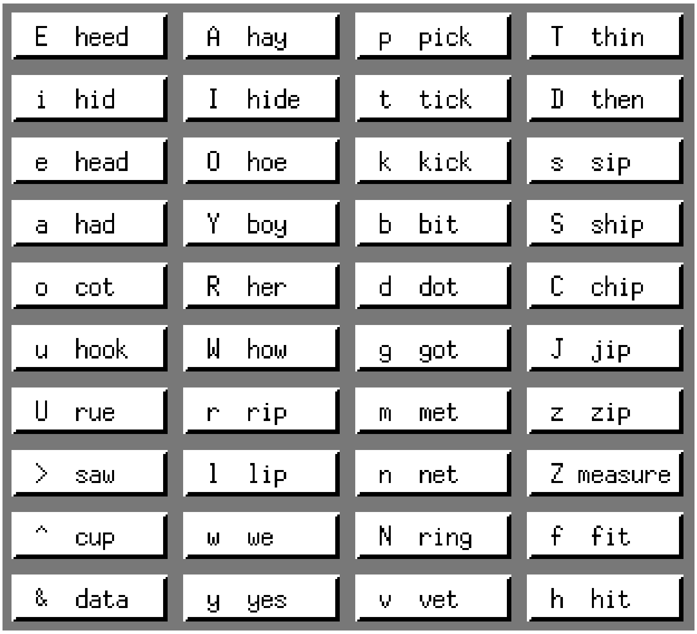
a aki	h hai	o go	u umi
b bengoshi	i ichi	p pan	w watashi
C hachi	j juu	r roku	y kyoo
d denwa	k kagi	s san	z zasshi
e enpitsu	m musuko	S shigatsu	
f futon	n ni	t tabemono	
g gakusei	N hon	T rokugatsu	

Figure 5-4. Japanese Phonemes



a amigo	i inviemo	s sol	R perro
b blanco	l libro	t taco	T madre
d dolor	m manos	u uno	k coma
e espaldo	n nombre	y ya	z desde
f fiesta	o ocho	B cabeza	
g gato	p piema	C mucho	
h gente	r pero	N mañana	

Figure 5-5. Latin-American Spanish Phonemes



E heed	A hay	p pick	T thin
i hid	I hide	t tick	D then
e head	O hoe	k kick	s sip
a had	Y boy	b bit	S ship
o cot	R her	d dot	C chip
u hook	W how	g got	J jip
U rue	r rip	m met	z zip
> saw	l lip	n net	Z measure
^ cup	w we	N ring	f fit
& data	y yes	v vet	h hit

Figure 5-6. US English Phonemes

FlexWord Capacity

Every entry in a wordlist is counted as a distinct and separate word. The maximum number of words that can be loaded onto the FlexWord speech recognizer is 2000. (FlexWord speech recognition supports up to 200 wordlists and 500 words per wordlist, as shown in Table 5-1.) For example, if the word “help” appears as an entry in 10 of your wordlists, “help” is counted as 10 separate words. Phrases such as “loan information” count as only one word. Each wordlist must be given a name tag consisting of 1–14 uppercase characters. For definitions of word, wordlist, or vocabulary terms, see the “Glossary” at back of this book.

Table 5-1. FlexWord Maximum Capacity

Wordlist entries (words or phrases)	2000
Wordlists	200
Words per wordlist	500
Characters per wordlist name tag	64
Phonetic transcription length	64

FlexWord Toolkit

FlexWord supports fast, low-cost delivery of speech recognition vocabularies on a custom basis through the use of the FlexWord toolkit. The FlexWord toolkit is an optional package that supports unique or dynamic applications such as name dialing, menu selection by words, command words, and entry of client or department names. This option gives you a point-and-click graphical environment for adding, deleting, or changing words on new or existing wordlists. You can use these wordlists in your application immediately after they are created. The FlexWord toolkit supports all available languages.

NOTE:
Starting with the release of Version 6.0, running the FlexWord toolkit no longer requires stopping the voice system.

Vocabularies

A FlexWord speech recognition vocabulary can include a maximum of 2000 words or phrases per system, allowing up to 200 wordlists. Each wordlist can include up to 500 words. (For an application that requires more than 2000 words, contact your Lucent Technologies representative.)

The following are a few possibilities for creating the 2000 words:

- 4 wordlists with 500 words each for a total of 2000 words
- 50 wordlists with 40 words each for a total of 2000 words

A word can be used in more than one wordlist. Each prompt in a FlexWord speech recognition script requires a wordlist, which is a list of valid words with which the caller could respond. For example, for a script to play a prompt that asks, "Would you like information on your checking account or savings account?", a wordlist including the entries "checking" and "savings" is required.

Keyword Spotting

The FlexWord speech recognition standard vocabulary package can recognize or spot a keyword within a spoken phrase. The system filters out extraneous speech or noises (such as a cough) during recognition. The caller's input is not limited to the words in the system vocabulary. For example, if one of the words in your wordlist is "checking," then the system can recognize the word "checking" when a caller says "checking account please". However, to maintain high recognition accuracy it is advisable to structure the application to prompt the caller for the required information only. Applications should encourage the caller to speak only what is required.

Phrase Screening

You can program the FlexWord speech recognition package to alert and reprompt the caller if the system does not recognize the spoken response. For example, if the prompt states, "Please respond with 'checking' or 'savings'," and the caller speaks the word "banking", the application can be designed to inform the caller that the response is invalid and to replay the prompt.

Barge-In

FlexWord speech recognition does not support barge-in. Callers must wait until the system is completely finished prompting for information before they can speak their response.

Further Processing

You can improve recognition accuracy using the **recog_dip** by performing post-processing on the recognition candidates. See Chapter F, "Recognition Post-Processing'', for further information on **recog_dip** and other recognition post-processing functions.

Designing a FlexWord Speech Recognition Application

Preparing a FlexWord speech recognition application entails defining wordlists and using the wordlists in scripts.

Defining Wordlists

The first step in designing a wordlist is to define the words you want the system to recognize and group these words into wordlists. As Figure 5-7 shows, each prompt should have its own wordlist. The first menu prompt looks to the "INFORMATION" wordlist to verify the caller's first request. The second menu prompt looks to the "LOAN" wordlist to verify the caller's second request. The wordlists in both Figure 5-7 and Figure 5-8 contain groups of possible words from which callers can choose. All of the wordlists for all of your applications constitute the vocabulary.

Using Wordlists in Scripts

Each application can use several wordlists. You must specify the name of the wordlist to be used for a prompt. FlexWord speech recognition looks to the wordlist that you designate.

- In Script Builder, the wordlist is specified in the `Recog`*:* field on Page 2 of the Define Prompt and Collect screen step (the screen shows the name tags of all available wordlists). See "Defining the Prompt & Collect Action Step" in Chapter 7, "Defining the Transaction," in *INTUITY™ CONVERSANT® System Version 7.0 Application Development with Script Builder*, 585-313-206, for details.

  NOTE:
  A wordlist name tags is derived from the a wordlist file name; thus, a wordlist with the name tag "LOAN" is listed as the file **LOAN** in the **/vs/asp/flexword** directory. This name tag appears as one of the choices in the `Recognition_Type` field of the Define Prompt & Collect screen.

- In Voice@Work, the wordlist is specified through the Confirm tab in the Node Inspector for the Prompt and Collect node. See the chapter on node descriptions in *Voice@Work*, 585-313-207, for details.

The directory **/att/asr/wordlists/active** on the INTUITY CONVERSANT system is the directory for all active wordlists. The content of each wordlist file includes the spelling and phonetic breakdown of all of the words on the specific wordlist. The words "account information," "mortgage information," and "loan information," for example, are included in the "LOAN" wordlist file in the **active** directory.

FlexWord speech recognition supports word spotting, that is, it has the ability to search past extraneous speech during a recognition and scan for specified words. For example, if callers are given the menu choice “account” and they specify “account please,” FlexWord speech recognition recognizes and accepts the word “account” and ignores “please.” After receiving spoken input, the FlexWord speech recognition algorithm scans the appropriate wordlist and generates a group of candidates that most closely resembles the caller’s response. The algorithm returns the most likely match to the script. Defining the wordlist appropriately is essential for the success of an application.

For more information on how to create effective wordlists, see “Positive Influences on FlexWord Speech Recognition Accuracy” below.

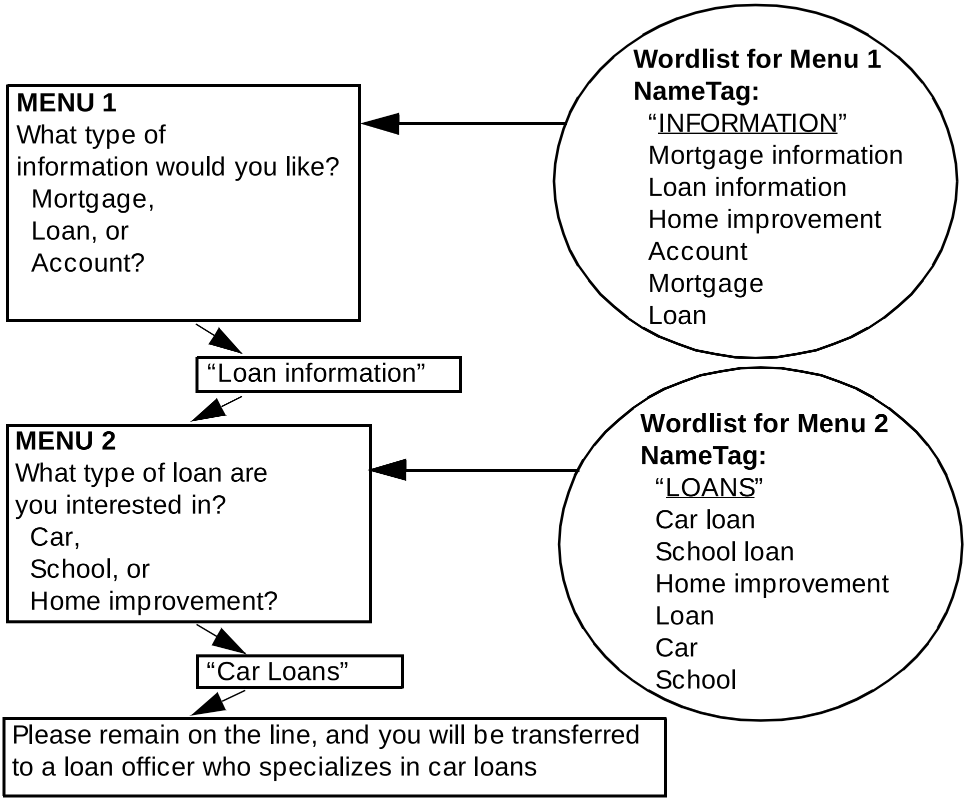


Figure 5-7. FlexWord Speech Application Example Using Menu Prompts

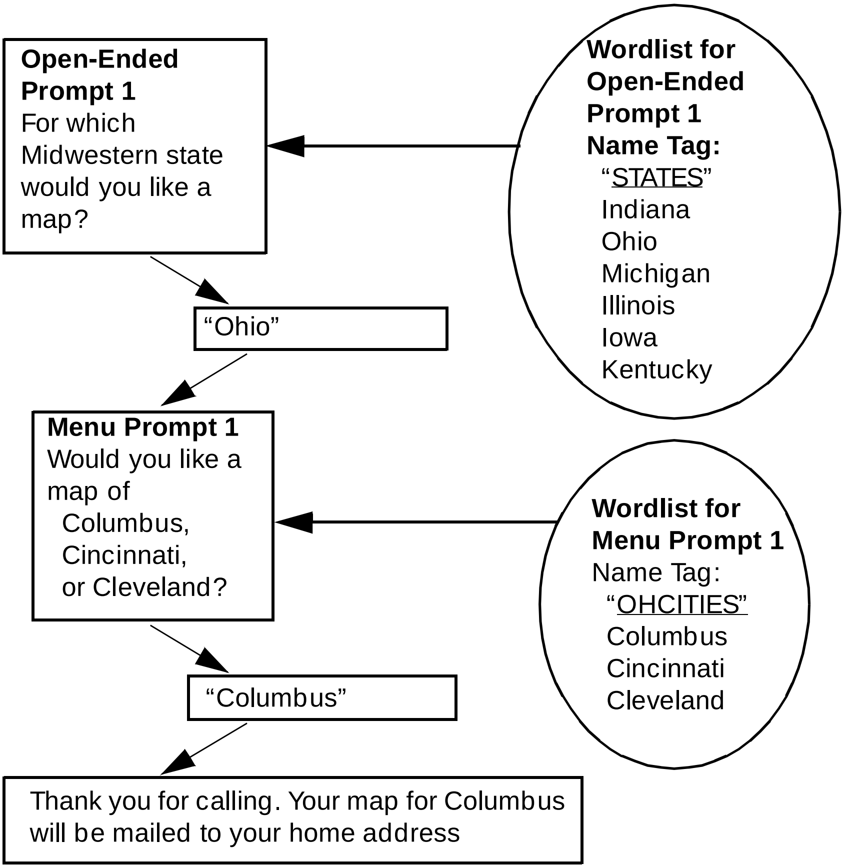

Open-Ended Prompt 1
For which Midwestern state would you like a map?
"Ohio"
Menu Prompt 1
Would you like a map of Columbus, Cincinnati, or Cleveland?
"Columbus"
Thank you for calling. Your map for Columbus will be mailed to your home address
Wordlist for Open-Ended Prompt 1
Name Tag:
"STATES"
Indiana
Ohio
Michigan
Illinois
Iowa
Kentucky
Wordlist for Menu Prompt 1
Name Tag:
"OHCITIES"
Columbus
Cincinnati
Cleveland

Figure 5-8. FlexWord Speech Application Example Using Open-Ended and Menu Prompts

Using the FlexWord Toolkit to Create Wordlists

The FlexWord toolkit allows you to construct words and phrases by stringing phonemes together. The toolkit includes a standard dictionary corresponding to the language chosen, as well as three commands for creating and modifying wordlists. The correct software and hardware are both necessary, and the hardware must be properly configured before you can create, edit, and delete wordlists.

NOTE:
The Version 6.0 toolkit and its resulting wordlists can only be used with an INTUITY CONVERSANT Version 6.0 or greater system.

FlexWord Software and Hardware

- For a list of the required software and hardware, see "FlexWord Speech Recognition'' in Chapter 1, "Overview of Speech''.
- To install the FlexWord feature software, see "Installing the FlexWord Speech Recognition Package" in "Installing Optional Feature System Software" in the INTUITY CONVERSANT maintenance book for your platform.
- To configure the video controller circuit card, see "Setting Up the Monitor" in "Installing Base System Software" in the INTUITY CONVERSANT maintenance book for your platform.
- To initialize the mouse, see "Initializing the Mouse," in "Installing Base System Software" in the INTUITY CONVERSANT maintenance book for your platform. Normally, the mouse is configured for the COM2 serial port. For information about the serial mouse port, see "Connecting to Asynchronous Devices" in Chapter 3, "Making Cable Connections and Powering Up the System" of *INTUITY™ CONVERSANT® System Version 7.0 New System Installation, 585-313-106.*

Configuring the Tip/Ring Circuit Card

The Tip/Ring circuit card plays the phonetic pronunciation of the words (used for US English only*)* in a wordlist.

NOTE:
The voice system must be running in order to play the words in a wordlist.

It is most convenient to configure your Tip/Ring circuit card with OS Index 0. However, it is not required. If the OS Index is not zero, you must specify the correct channel number with the **wl_edit** command. See "Replacing, Installing, or Upgrading Circuit Cards" in the INTUITY CONVERSANT maintenance book for your platform, for more information about the Tip/Ring circuit card. See "Creating a FlexWord Wordlist" below for usage of the **wl_edit** command.

Configuring the Speaker and Amplifier

If you use an external speaker and amplifier system for phonetic pronunciation from within the FlexWord editor (used for US English only), the AUDIO OUT jack on the faceplate of the Tip/Ring circuit card must be connected to the speaker and amplifier system. For more information about the Tip/Ring circuit card, see Chapter 3, "Making Cable Connections," in *INTUITY™ CONVERSANT® System Version 7.0 New System Installation*, 585-313-106.

Connecting the Telephone Line

If you use a telephone line for the phonetic playback within the FlexWord editor (used for US English only), the default connection for the telephone line is to Port 1 on the Tip/Ring circuit card (that is, the lowest channel, 0, on the Tip/Ring circuit card). The port is the first modular jack labeled 1, 2, 3 and provides access to channels 0, 1, and 2. See Chapter 3, "Making Cable Connections and Powering Up the System," in *INTUITY™ CONVERSANT® System Version 7.0 New System Installation*, 585-313-106, for an illustration of Port 1 on the Tip/Ring circuit card.

NOTE:
The AUDIO OUT jack on the Tip/Ring circuit card should be physically disconnected from any speaker and amplifier system.

To use a telephone connection with the FlexWord editor, after you initialize the editor, place a call to the telephone number associated with the channel you previously identified. The FlexWord editor takes the line off-hook and can then communicate to the user through the telephone handset.

Configuring the SSP Circuit Card

Once configured for TTS, the SSP circuit card generates the phonetic pronunciation from within the FlexWord editor. TTS is available for US English only.

NOTE:
The FlexWord function should only be assigned to the SSP circuit card when at least one wordlist is currently activated. If the FlexWord function is assigned when no wordlists are activated, when the SSP circuit card is restored, the card will go to a BROKEN state.

Configuring the SSP circuit card for TTS functionality involves several steps. First, you must determine which SSP circuit card should be assigned TTS functionality. Use the following procedure to make the determination.

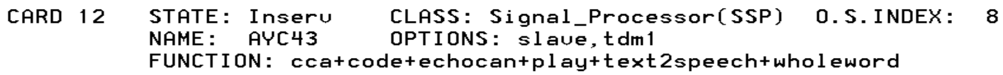
1. Enter **display card sp**

   The system displays output in the format shown in Figure 5-9.

2. Make a note of the card number and the number in the `O.S. Index` field.

3. Assign the TTS function to the chosen circuit card number.

   See “Assign SP/SSP Functions” in Chapter 3, “Voice System Administration” in I*NTUITY™ CONVERSANT® System Version 7.0 Administration*, 585-313-501.



```
CARD 12    STATE: Inserv      CLASS: Signal_Processor(SSP)   O.S.INDEX:   8
           NAME:  AYC43       OPTIONS: slave,tdm1
           FUNCTION: cca+code+echocan+play+text2speech+wholeword
```

Figure 5-9. Output from the Display Card Command After Assigning TTS Functionality

NOTE:
Before you begin using the FlexWord toolkit, verify that all the circuit cards needed are in service by repeating Step 1.

FlexWord Commands

The following commands are used in the procedures to create wordlists, modify wordlists by adding and deleting entries, make phonetic changes in a wordlist., copy wordlists to diskette, and activate wordlists. For information on these commands, see *INTUITY CONVERSANT System Version 7.0 Administration*, 585-313-501.

- **wl_init**
- **wl_edit**
- **wl_copy**
- **wl_install**
- **wl_gen**

NOTE:
You do not have to stop the voice system to run any of the commands associated with the FlexWord feature.

Using the Mouse

The left mouse button is used to select items from the menu bar on the FlexWord editor window as well as to move the marker so that you can type in the `Phonemes` and `String to Add` boxes.

NOTE:
You can also select items from menu bar by typing the letter indicated with an underscore. For example, entering **f** at the keyboard while the FlexWord editor window is displayed accesses the File menu.

Creating a FlexWord Wordlist

Use the following procedure to create a FlexWord wordlist:

1. Log in as **root**.
2. Select a working directory.

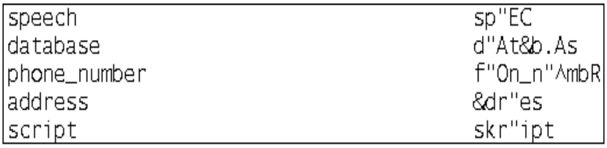
   You may want to create a directory with a name similar to the wordlist you will create.
3. Using vi (or other editor), create a file in the working directory that contains an entry for each word or phrase you want to recognize in an application.
   - The filename must be uppercase, alphanumeric, and 14 characters or less in length. (The filenames must be uppercase for the **wl_gen** command to execute.)
   - Each entry must be on a separate line.
   - White space is not allowed. Separate multiple words in an entry with an underscore (_). For example, in a list of cities, Grove_City would be written with the underscore as shown.
   - To type accented characters in non-English languages, see Table 5-2.
   - For guidelines on including out-of-vocabulary (OOV) entries, see “Including Out-of-Vocabulary Terms” below.
   - The maximum number of words in a wordlist must not exceed 2000 (see Table 5-1).
4. Enter **wl_init *filename* [*language*]**

   where:
   - *filename* is the name of the wordlist file.
   - *language* is the language for the wordlist (Brazilian, English, French, German, Japanese, or Spanish).

   The system adds a phonetic transcription for each entry in the wordlist according to the language you specify (English is the default), and then displays the following message:

```
File filename now contains initial phonetic breakdown
Enter cat <filename> to view the phonetic spelling of
each word/phrase in your file.
```

You can use vi (or other editor) to view the phonetic transcriptions the **wl_init** command has added. Figure 5-10 shows an example of the command output. (You can also view the phonetic transcription once you open the wordlist in the FlexWord editor by clicking on a wordlist entry and looking in the Phonemes window.)



```
speech                                      sp"EC
database                                    d"At&b.As
phone_number                                f"On_n"^mbR
address                                     &dr"es
script                                      skr"ipt
```

Figure 5-10. Example of a Wordlist File with Phonetic Spelling For US English

Modifying Entries in a FlexWord Wordlist

The FlexWord editor can be used to change wordlist entries to reflect alternative pronunciations, for example regional dialects. You may want to include two or more versions of the same entry to account for alternative pronunciations. You can also use it to correct pronunciations for items for nonstandard pronunciation. For example, a street named Lahser may commonly be pronounced as "Lasher," despite the spelling.

For US English, you can use TTS to hear the pronunciation of the wordlist entries as an aid to decide whether entries need to be modified. For other languages, you may be able to make the modifications based on the list of phonemes displayed in the FlexWord editor.

Use the following procedure to modify the phonetic transcription of entries in a FlexWord wordlist:

1. Log in as **root**.
2. Enter **wl_edit [-l *chan#*] [-s *ssp#*] [-D *dirname*] [-O] [-L *language]***

   where:

   - *chan#* is the number of the Tip/Ring channel to be used for speech playback. If you do not specify this parameter, playback defaults to channel 0.
   - *ssp#* specifies which SSP circuit card to use for speech playback.
   - *dirname* is the directory in which the FlexWord editor looks for word list files.
   - -O gives the maximum messaging output to the screen.
   - *language* specifies the language output that appears on the screen (Brazilian, English, French, German, Japanese, or Spanish).

NOTE:
Error messages are written to a small window at the bottom of the screen. Messages are also logged to the file **/usr/tmp/wledit.output.**

The system displays the FlexWord editor window (Figure 5-11). The editor displays phonemes for the language you selected.

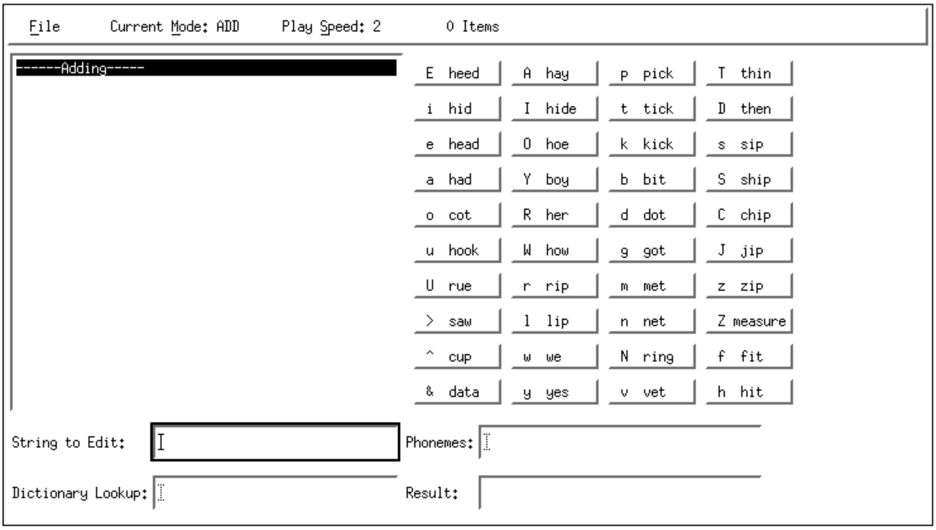


Figure 5-11. FlexWord Editor Window

3. Call into the Tip/Ring channel number you previously specified if you are set up to call into a telephone line (see "Connecting the Telephone Line" above).

NOTE:
It may take a few minutes for the channel to go on-hook. To verify whether the channel is on-hook, you can use the **sysmon** command. See *INTUITY CONVERSANT System Version 7.0 Administration*, 585-313-501 for more information.

4. Select **File** from the menu bar at the top of the FlexWord editor.

   The system displays the File menu.

5. Select **Open**.

   The system displays the Open File window (Figure 5-12).

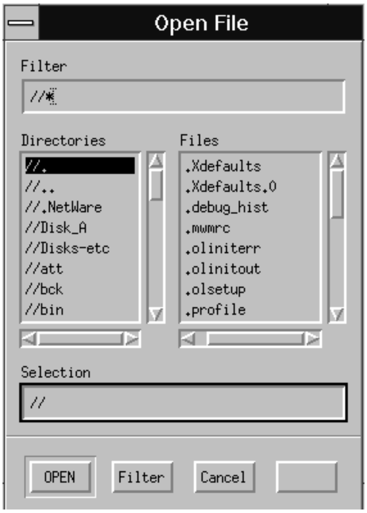


Figure 5-12. Open File Window in the FlexWord Editor

6. From the list of files, select the wordlist to be edited.

   NOTE:
   The directory that is displayed is the last directory accessed before the FlexWord editor was invoked. It may not be the directory containing your wordlist. You may need to select the correct directory.

7. Select the **Open** button.

   The FlexWord editor opens the wordlist file.

   If TTS is enabled on the SSP circuit card, TTS pronounces the first entry in the wordlist. If you have not selected US English as your language, TTS will attempt to pronounce the first entry based on English phonemes. Phonemes that are not contained in the set for US English may not be pronounced correctly.

   NOTE:
   Step 8 is for US English only.

8. Listen to the phonetic pronunciation of each entry by selecting it in the wordlist.

   You can adjust the play speed by selecting **Play Speed** from the menu bar at the top of the FlexWord editor and then selecting the rate (1 through 4) from the Play Speed menu.

9. Type any necessary changes to the phonetic transcription of a selected wordlist entry in the Phonemes box on the FlexWord editor.

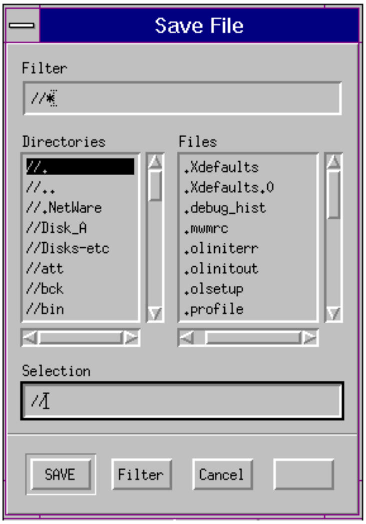
- Use the phonemes shown in Figure 5-1 through Figure 5-6, depending on your language.
- To add emphasis, use the following characters:
  - — (quote) " — Indicates primary emphasis and must go before a vowel sound.
  - — (period) . — Indicates secondary emphasis and must go before a vowel sound.

If TTS is enabled (US English only), you can listen to the pronunciation of each phoneme displayed in the FlexWord editor by clicking on it. Otherwise you can use the word displayed along with each phoneme to judge the sound of the phoneme.

10. Select **File** from the menu bar at the top of the FlexWord editor.
11. Select **Save**.

    The system displays the Save File window (Figure 5-13).
12. Select the correct file to save.
13. Select the **Save** button on the Save File window.

    The system saves the wordlist, including any changes you made to the phonetic transcriptions.
14. Select **Quit** from the File menu at the top of the FlexWord editor to exit the editor.



Filter: //*

Selection: //

Figure 5-13. Save File Window in the FlexWord Editor

Adding Entries to a Wordlist

Once a wordlist has been created, you can use the FlexWord editor to add entries to it. Use the following procedure to add entries to a wordlist:

1. Open the wordlist by completing Steps 1 through 7 of in "Modifying Entries in a FlexWord Wordlist" above.
2. Decide where you want the new entry to be placed in the wordlist.

   The new entry will be added below the currently selected entry.
3. Select **Current Mode** from the menu bar at the top of the FlexWord editor (Figure 5-11).
4. Select **Add**.

   The FlexWord editor displays a `String to Add` box.
5. In the `String to Add` box, type the entry to be added and press ENTER.

   The FlexWord editor adds the new entry and its phonetic transcription. If TTS is enabled on the SSP circuit card, TTS pronounces the entry. If you have not selected US English as your language, TTS will attempt to pronounce the entry based on English phonemes. Phonemes that are not contained in the set for US English may not be pronounced correctly.

   NOTE:
   TTS pronounces the entry according to what the dictionary tells TTS. If the entry is pronounced incorrectly, you can change the phoneme string so that TTS pronounces it correctly. However, upon subsequent lookups of the entry, TTS will still pronounce it according to the dictionary lookup. You must manually adjust the pronunciation by using the Phonemes box to make the change permanent.
6. Type any necessary changes to the phonetic transcription of the entry in the Phonemes box on the FlexWord editor. Follow the guidelines in Step 9 of "Modifying Entries in a FlexWord Wordlist" above.
7. Press ENTER to add the entry to the wordlist.
8. Repeat Steps 4 through 7 to add another entry to the wordist.

Deleting Entries from a Wordlist

Use this procedure to delete an entry from a wordlist.

1. Open the wordlist by completing Steps 1 through 7 of the procedure in "Modifying Entries in a FlexWord Wordlist" above.

   The system displays the Current Mode menu.
2. Select **Current Mode** from the menu bar at the top of the FlexWord editor (Figure 5-11).

3. Select **Edit**.
4. Select the word or wordlist you want to delete.
5. Select **Delete** from the Current Mode menu.

   The system removes the selected entry from the wordlist.

   NOTE:
   You must select **Edit** before selecting **Delete**. If you try to delete an entry while in the Add mode, the system displays the message, `Cannot DELETE while in ADD mode!`

6. Repeat Steps 2 through 5 to delete another entry.

Special FlexWord Recognition Accents

Table 5-2 lists the special or accented characters used in certain languages and the input strings used that represent these characters in FlexWord wordlists.

For example, to input a word containing the French accented character “á”, type **(a’**, consisting of a left parentheses character, lowercase letter “a”, and single quotation character. To input a word containing the French accented character “à”, type **(a‘**, consisting of a left parentheses character, lowercase letter “a”, and grave character.

NOTE:
On most keyboards, the single quotation character is on the same key as the double quotation character. The grave character is typically on the same key as the tilde (~) character.

Table 5-2. Global Recognition Accents

Brazilian Portuguese		French		German		Japanese		Latin-American Spanish	
â	(a^	â	(a^	ä	ae	û	uu	á	‘a
à	(à	à	(a`	ö	oe	n’	n’	é	‘e
á	(a’	á	(a’	ü	ue	Observe these guidelines for Japanese: ■ Repeat any phoneme with a circumflex (^) over it, for example: aa ■ Do *not* use apostrophes or bars over vowels, for example: í		í	‘i
ã	(a~	c’	(c’	ß	ss			ó	‘o
ç	(c,	ç	(c,					ú	‘u
é	(e’	d’	(d’					ñ	~n
ê	(e^	ë	(e:						
í	(i’	é	(e’						
ó	(o’	è	(e`						
ô	(o^	ê	(e^						
õ	(o~	ï	(i^						
ú	(u’	l’	(l’						
û	(u^	ô	(o^						
ü	(u:	ù	(u`						
		û	(u^						
		ü	(u:						

FlexWord Recognition Vocabulary Administration

Vocabulary administration involves installing wordlists and moving them between the **active** and **inactive** directories. Also, the FlexWord function may need to be assigned to or unassigned from the SSP circuit card.

Copying Wordlists

Use this procedure to copy a wordlists to diskette in preparation for installing them on a target machine loaded with the FlexWord speech recognition software.

NOTE:
If the FlexWord software is installed on the same system where the wordlists reside, do not use this procedure. Instead copy the wordlists to the **inactive** directory (**/att/asr/wordlists/inactive**) with a standard UnixWare command.

1. Go to the directory where the wordlists are located.
2. Insert a diskette into the diskette drive.
3. Enter **wl_copy** ***filename filename filename ...***

   NOTE:
   Filenames must be relative, not absolute pathnames, since they will be used to load the wordlists onto a FlexWord recognition system.

   The system copies the files or directories specified to diskette. If any of the names are directory names, the contents of the directories are also copied to diskette.

Installing and Activating Wordlists from Diskette

Use the following procedure to install wordlists on the INTUITY CONVERSANT system from diskette and activate them.

1. Enter **wl_install**
2. Insert the diskette containing the wordlists (made in the procedure above "Copying Wordlists") into the diskette drive when prompted.

   The system copies the wordlists to the **inactive** directory (**/att/asr/wordlists/inactive)** and prompts as to whether any of the wordlists should be activated and the **wl_gen** command run.
3. Designate, if appropriate, the wordlists that should be activated.

   The system copies the designated wordlists to the **/att/asr/wordlists/active** directory and creates the data files needed for FlexWord recognition.

   NOTE:
   The **active** directory should contain only wordlists. The system does a format check when the data files are generated, and if files other than wordlists are in that directory, the system generates an error message.

Activating Wordlists Resident on the System

Activating a wordlist means creating all of the data files necessary to perform FlexWord speech recognition. Use the following procedure to activate a wordlist when the wordlist file already resides on the target VOICE system.

NOTE:
Do not use this procedure if you have already activated the wordlists through the **wl_install** command. See "Installing and Activating Wordlists from Diskette" above.

1. If there are no wordlists currently activated, make sure the FlexWord function is *not* assigned to the SSP circuit card.
2. Make sure the wordlist you want to activate is currently in the **inactive** directory (**/att/asr/wordlists/inactive**)
3. From the **inactive** directory, enter **ln *filename* /att/asr/wordlists/active**

   where *filename* is the name of the wordlist you want to activate.

   The system creates a hard link between the file in the **inactive** directory and the **active** directory.

   CAUTION:
   *Do not run ScriptBuilder at the same time as the **wl_gen** command in the step below.*

4. Enter **wl_gen [*language*]**

   where *language* is Brazilian, English, French, German, Japanese, or Spanish.

   The **wl_gen** command verifies the format of all active wordlists and creates the data files needed for FlexWord speech recognition. If the command finds more than 38 phonemes, including the underscore (_), within any entry in a wordlist, you receive an error message. Once **wl_gen** finds an error in a wordlist, it quits looking at that wordlist, and the wordlist containing the error is not used when generating the FlexWord recognition data files. Therefore, if you receive an error message, you may have to run **wl_gen** several times to locate each error.

5. Enter **diagnose card** to diagnose the FlexWord speech recognition SSP circuit card after activating your wordlists.

   See *INTUITY™ CONVERSANT® System Version 7.0 Administration,* 585-313-501, for more information about the **diagnose card** command.

6. Verify and install the application associated with the active wordlist before you call up the application.

   - For a Script Builder application, see "Verifying and Installing the Application," in Chapter 10, "Application Administration," in *INTUITY™ CONVERSANT® System 7.0 Application Development with Script Builder,* 585-313-206.
   - For a Voice@Work application, see the section on the SCI tool in Voice@Work, 585-313-207.

7. Assign the FlexWord function to the SSP circuit card.

   For the procedure, see “Assign SP/SSP Functions” in Chapter 3, “Voice System Administration” in *INTUITY™ CONVERSANT® System Version 7.0 Administration*, 585-313-501.

Deactivating Wordlists

Deactivating a wordlist means removing all data files associated with the wordlist. Use the following procedure to deactivate a wordlist.

1. If deactivating this wordlist will leave no wordlists activated on the system, make sure FlexWord function is *not* assigned to the SSP circuit card.

   For more information, see “Assign SP/SSP Functions” in Chapter 3, “Voice System Administration” in *INTUITY™ CONVERSANT® System Version 7.0 Administration*, 585-313-501.

2. Make sure you are in the **/att/asr/wordlists/active** directory.
3. Remove the wordlist from the **/att/asr/wordlists/active** directory.

   For example, enter **rm *filename***

   where *filename* is the name of the wordlist that you want to deactivate.

   Removing the wordlist destroys the link created by the **ln** command when the wordlist was activated.

4. Enter **wl_gen [*language*]**

   where *language* is Brazilian, English, French, German, Japanese, or Spanish.

   The system removes the data files associated with the wordlist you deactivated. If there are no more active wordlists, the system displays a message to that effect.

Removing Wordlists

Removing a wordlist involves deleting both the data files and wordlists from the system. Use the following procedure to remove a wordlist.

1. Deactivate the wordlist.

   See “Deactivating Wordlists” above for the procedure.

2. Go to the **/att/asr/wordlists/inactive** directory.
3. Remove the wordlist from **/att/asr/wordlist/inactive**.

   For example, enter **rm *filename***

   where *filename* is the name of the wordlist to be deleted.

FlexWord Speech Recognition Accuracy

FlexWord speech recognition accuracy depends not only on the recognition algorithms, but also on prompts, calling populations, the words to be recognized, and application designs. Given the variance in human speech among speaker and the statistical properties of recognition algorithms, the speech recognizer will make occasional errors. But the errors can be minimized and recognition can be enhanced with careful planning.

Positive Influences on FlexWord Speech Recognition Accuracy

Factors that positively affect accuracy rates include:

- Effective construction of wordlists and prompts
- Inclusion of an out-of-vocabulary (OOV) list
- Experienced calling populations
- Prompts with good structure and design

See also Chapter 7, "Putting It Together'', for information about getting the most out of a FlexWord speech recognition application.

Effective Wordlists

Choose words for your wordlist that have different sounds and are of medium length to help increase recognition accuracy. For example, using the last name and the first name on wordlists of a name dialer application (application that allows employees to speak a colleague's name rather than dial an extension number) increases recognition accuracy. In other types of applications, syllabic and vowel similarities may impair recognition. For example, "women's wear" and "men's clothing" are more effective wordlist entries than "women's clothing" and "men's clothing." The latter pair sound too much alike. Both have "men" embedded in the word and both share the word "clothing."

Note the words your customers actually use when they ask for a service. If you are automating an existing transaction that has previously taken place between a customer and an agent, use the agent as a resource and try to mimic the words of the customer requests.

Including Out-of-Vocabulary Terms

Note words your customers may use that are outside of your vocabulary wordlist. If you place these out-of-vocabulary (OOV) words in the wordlist preceded by two dashes, the recognizer classifies them as being OOV. As a result of the nature of the words in the OOV list, spoken words that are in neither list can often map to words in the OOV list. These spoken words will then be properly classified and handled by the FlexWord speech recognition application, even when they are not accurately recognized. A sample OOV wordlist is delivered with each FlexWord speech recognition language:

/vs/asp/flexword/<*language*>.oov

This list contains some of the words most frequently used by callers interacting with a voice response system. Append this wordlist and any custom OOV terms to each of your regular vocabulary wordlists.

Experienced Calling Population

Recognition accuracy improves for applications in which the calling populations are closed and callers are experienced or trained to interact with the application.

Prompt with Structure and Good Design

Prompts offered in calm, clear voices greatly affect recognition accuracy, as does the structure of the prompts. Prompts should guide the caller to say desired words or phrases.

For best results, menu prompts should be built with the structure of <*desired result*> <*action required*> and dissuade callers from attempting to speak before the prompt is finished, for example:

“To hear your checking account balance, say ‘checking’.”
“To hear your savings account balance, say ‘savings’.”

By placing the required action at the end of the prompt, the caller is better able to remember it. Do not list the action required before the desired result. For example, the following prompt, encourages the caller to forget the specific action required, since the last thing heard is the desired result.

“Say ‘one’ for a description of the upcoming gallery events.”,

Confirmation and Reprompt Steps

Overall recognition accuracy can be increased if the application includes confirmation and reprompt steps. It is always a good idea to verify the recognized result before continuing with the application, for example:

“You said ‘spring registration’. Is this correct? Please say ‘yes’ or ‘no’.”

With this prompt structure, callers can make sure their words match the recognized response. Figure 5-14 provides an illustration of a confirmation reprompt.

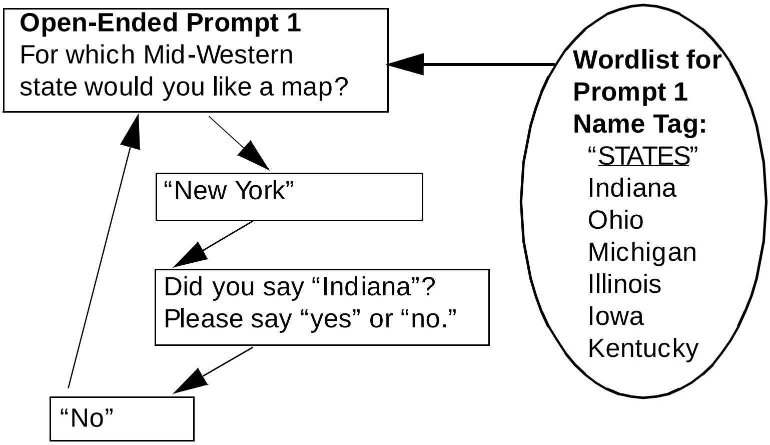


Figure 5-14. Example of a Reprompt in a FlexWord Application

Informative Prompts

Lengthy prompts providing detailed response instructions may improve recognition accuracy. However, experienced users often find lengthy prompts, since they must to listen to the complete prompt before responding. One solution may be to provide more informative prompts for first-time callers only. However, for applications with infrequent users, lengthy prompts may be more acceptable and help improve the accuracy.

Negative Influences on FlexWord Speech Recognition Accuracy

Low recognition accuracy rates are usually due to:

- Inexperienced callers
- Environmental factors
- Ineffective wordlists

Inexperienced Callers

Inexperienced callers may speak extraneous phrases or speak before a prompt is finished.

NOTE:
FlexWord speech recognition *does not* support barge-in capabilities. Thus, prompts should include some sort of time reference so that callers know when to respond.

The following examples illustrate effective and ineffective prompts. The effective prompt shows how recognition accuracy can be increased by including a time reference. This prompt encourages the caller to wait until the prompt is finished before responding. The ineffective prompt does not have the time reference "now." Without the "now" at the end of the sentence, the caller is more likely to attempt to speak before FlexWord recognition is ready.

Effective:	"Please say the name of the agent with whom you wish to speak, now."
Ineffective:	"Please say the name of the agent with whom you wish to speak."

NOTE:
For an application using FlexWord speech recognition, you must disable barge-in (speak with interrupt) for the Prompt and Collect action (Script Builder) or node (Voice@Work).

Environment

Noisy environments, such as an airport or a train station, or an unclear telephone connection may contribute to recognition accuracy problems.

Ineffective Wordlists

Ineffective wordlists cause lower recognition accuracy. In general, the larger the wordlist, the lower the expected recognition accuracy. Wordlists that contain short words and rhyming words also decrease accuracy:

- Wordlist size

  As the size of wordlists increase, accuracy decreases. The best accuracy results can be achieved by structuring an application to make use of several smaller wordlists rather than one large wordlist.

- Rhyming words

  Wordlists that include words with similar rhyming vowel sounds can cause a decrease in recognition accuracy.

- Short words

  One-syllable words that have the same vowel sounds are difficult to recognize. For example, "on" and "off" both share the short "o" sound. (Long "o" vowel sounds appear in words like "oh" and "no"— words in which you can hear the letter "o.") Short words should be used in moderation.

Recognizing Dial Pulse Input

6

Overview

Dial Pulse Recognition (DPR) how it allows a user with a rotary telephone or a push-button telephone that generates only dial pulses to respond to the INTUITY™ CONVERSANT® system. Since rotary telephones or push-button telephones that generate dial pulses are used in many countries. The DPR feature is available for a variety of countries to accommodate the dial pulse users.

NOTE:
DPR is supported for Script Builder applications. It is not currently supported for Voice@Work applications.

Purpose

The purpose of this chapter is to explain the following topics for DPR:

- Components and functionality
- Simultaneous dial pulse capabilities
- Standard dial pulse recognition types
- The accuracy of DPR
- Grammars
- Recognition confirmation
- Barge-in
- Troubleshooting guidelines

Description

DPR allows a user with a rotary telephone or a push-button telephone that generates dial pulses to respond to the INTUITY CONVERSANT system. DPR converts the “pops” and “clicks” on the line to dial pulses.

DPR supports digits “0” through “9” on analog and digital interfaces. DPR is available on the SSP (speech and signal processor) circuit card and on the IVC6 and NGTR Tip/Ring circuit cards. If you are using a digital line interface circuit card, you must use the SSP circuit card for DPR.

Simultaneous Dial Pulse Capabilities

DPR can work simultaneously with either DTMF (dual-tone multi-frequency), WholeWord speech recognition, or FlexWord™ speech recognition. WholeWord and FlexWord recognizers can independently work with the dial pulse recognizer. Combining DPR with a speech recognizer gives the caller the choice of either keying in the number with dial pulse or touchtone, or of using spoken input.

Dial Pulse Recognition Types

Recognition types are the choices associated with the `Recog:` field on the Script Builder Define Prompt & Collect screen. The system uses the recognition type along with other fields on the Define Prompt & Collect screen to select a recognition grammar to be used for that prompt.

Certain digit strings may require custom recognition types as a technique used to obtain acceptable recognition accuracy. Examples include a credit card number, a merchant number, and a 10-digit telephone number. These digit strings have specific limitations on the position of certain digits within the string (for example, a credit card whose first number must be 3 or 5, and whose second number must be 1 or 7).

Table 6-1 shows the standard dial pulse recognition types.

Table 6-1. Standard Dial Pulse Recognition Types

Digit Lengths and Digits Allowed	Recognition Type
Single digit 1 - 3	DP1_3
Single digit 1 - 5	DP1_5
Single digit 2, 5, or 8	DP258
Single digit 1, 4, 7, or 0	DP1470
Any digits 1 - 9 and 0	DP1_10
Any digits 2 - 9 and 0	DP2_10
Any digits 3 - 9 and 0	DP3_10

Grammars

A grammar is a set of rules that specifies allowable vocabulary words, vocabulary word combinations, or pulses at any one point. Grammars are built into a recognition type and increase recognition accuracy or limit expected input. After collecting dial pulse input, the DPR algorithm uses grammars to generate a list of candidates that most closely resemble this input. The algorithm returns the most likely match to the script or, if no match fits, rejects the input.

Grammars for DPR are modeled after grammars for speech recognition. DPR currently uses grammars only to do input restriction such as “ignore digit one” and pulse-to-digit mapping. Grammars provided for DPR in Script Builder applications are described in “Using Dial Pulse Recognition” in Chapter 8, “Using Optional Features,” of *INTUITY™ CONVERSANT® System Version 7.0 Application Development with Script Builder,* 585-313-206.

Recognition Confirmation

Recognition confirmation is the ability of the system to repeat the caller’s dial pulse response back to the system for confirmation. For example, an application prompts a caller to “Please enter your area code,” and the caller dials [6] [1] [4] or speaks “614.” The Prompt and Collect action can be structured using the DP258 recognition type, thereby restricting the expected input to only 2, 5, or 8. Using recognition confirmation, the system repeats what it recognized the caller’s response to be and then requests, “If this is correct, enter 2, if incorrect enter 5.” The caller then confirms whether the system recognized the input properly.

When an application requires a long string of numbers such as an account number from a caller, these can be requested and confirmed one digit at a time. Although time consuming, this process establishes higher accuracy and decreases the likelihood of the system accepting a faulty account number.

Barge-In

Barge-in (also called recognize during prompt) is the ability of the system to allow callers to interrupt during prompts by entering DPR digits.

NOTE:
Currently barge-in is not supported for use with DPR.

How Dial Pulse Recognition Works

Dial-pulse digits that a user enters are collected by the algorithm on the SSP circuit card or Tip/Ring circuit card and processed according to the selected recognition type. A maximum of 60 channels are available per SSP circuit card. When a Tip/Ring circuit card is used, all lines have DPR available.

If the DPR software package is installed, no administration is required for the Tip/Ring cards. For DPR to run on the SSP circuit card, it must be assigned through the cvis_menu screens. For information on assigning SSP functionality, see Chapter 3, See “Assign SSP Functions” in Chapter 3, “Voice System Administration” in I*NTUITY™ CONVERSANT® System Version 7.0 Administration*, 585-313-501.

If an application is running on a Tip/Ring line and requests DPR, the system tries to run DPR on the Tip/Ring card, thus freeing up the SSP circuit card for other tasks.

The Recognizer

The SSP circuit card or Tip/Ring circuit card is loaded with the DPR software package, also called the *recognizer*. The recognition type is defined in the `Recog:` field on the page 2 of the Define Prompt & Collect screen.

For a prompt of “2, 5, 8,” the application developer selects the DP258 recognition type, which is used for a response consisting of “2,” “5,” or “8.” The caller's response to the prompt is compared to the numbers 2, 5, or 8. The number of pulses that most closely matches the input determines the recognized result. A value of 2, 5, or 8 is returned in $CI_VALUE. The application decides what to do next, based on the value of $CI_VALUE.

Recognition Accuracy

False recognition can occur if the caller is in a noisy environment or on a noisy telephone line. For example, an electrical disturbance that causes a “pop” sound on the telephone line can be recognized as a dial pulse digit. Speakerphones can aggravate this condition since they amplify room noise. False recognition is reduced once training has been completed.

Training

Training is the ability of the system to learn the characteristics of each telephone. Training occurs when a caller first accesses the system. The application should ask the caller to enter a number greater than 5 for training to converge. This process increases accuracy of the system while decreasing false recognition due to extraneous noises.

For example, your first prompt may be, “Please enter ‘8’ followed by your account number.” When validating the user input, you should ignore the initial digit in the string. It is not necessary to create a separate input sequence just for training. At this point, the system has learned about the caller’s telephone characteristics.

Training is an option that can be turned on or off. When turned on, training is automatic in that it shuts off after the caller has entered the number greater than 5.

Limitations

In integrations with the DEFINITY® G3 switch, unexpected behavior from the dial-pulse recognizer may occur. DEFINITY G3 converts dial pulse digits to DTMF tones only when the input telephone is directly connected to the DEFINITY switch. This eliminates the need for DPR since the system always responds to DTMF tones.

Some central office (CO) switches may not pass dial pulse digits through once a call has been established. The caller’s local switch (in the CO) may see the loss of loop current as a hangup or a register recall and disconnect the caller. Although the problem is not common, there is nothing the INTUITY CONVERSANT system’s DPR can do in this situation.

NOTE:
DPR applications cannot receive the pound sign (#), asterisk (*), or the letters “a” through “d,” all of which are available on DTMF telephones. Applications using DPR must be redesigned accordingly.

DPR Troubleshooting

Use Table 6-2 to troubleshoot problems with DPR.

Table 6-2. DRP Troubleshooting

Problem	**Action**
DPR does not appear to be working.	A path is provided in the **feature_tst** script delivered with the system. Load this package by selecting DPR as a test path. Make sure DPR is assigned to the appropriate circuit card and is in service. The test asks you to dial any four dials pulse digits and speaks the result back to you.
DPR does not recognize digits.	A number of problems could cause DPR not to work. ■ Ensure all administration and assignments for DPR are correct. ■ Use the **feature_tst** script to determine if it is an application or system issue. ■ Check the system error logs and take appropriate corrective action. If these actions do not reveal the problem, contact your Lucent Technologies Technical support center representative at 1-(800)-242-2121.
DPR mis-recognizes digits.	If digits are consistently mis-recognized from multiple callers, some local tuning may be necessary. Contact your Lucent Technologies technical support center representative at 1-(800)-242-2121.

Common Failure Modes

DPR failure is usually a result of either noise on the line or missing (or extra) pulses.

Noise on the Line

Recognition failure due to noise on the line can most often this occurs with the digit “1” and sometimes with the digit “2”. If you have application control over the requested user input, you can avoid the use of these digits.

Missing or Extra Pulse

When the recognizer incorrectly recognizes, it is usually off by one digit. Application validation can decide to accept this and continue. For example, if the caller is asked to enter an account number and personal identification number (PIN), you may decide to accept the input even though the PIN is off by one digit.

There are standard recognition types for single-digit prompts requiring four or fewer choices that provide good recognition accuracy. This is accomplished by separating the valid choices by three pulses and mapping all invalid choices to their closest neighbor.

Putting It Together 7

Overview

This chapter describes:

- How to get the most out of speech-related features in applications
- Why certain features work well together
- When you can use speech-related features together

Recognition types, including FlexWord™ speech recognition, Whole Word speech recognition, and dial pulse recognition (DPR) are discussed, as well as the TTS (Text-to-Speech) feature.

Purpose

The purpose of this chapter is to ensure that you can use speech-related features effectively in applications.

Using WholeWord Recognition and Dial Pulse Recognition Together

WholeWord speech recognition can be used to recognize a limited set of words, as discussed in Chapter 4, "Recognizing WholeWord Speech Input". However, WholeWord speech recognition is most successful when you use it to augment a touch-tone application to handle callers who do not have or for some reason do not want to use touch-tone telephones.

WholeWord speech recognition and DPR are the only practical ways to provide connected-digit recognition. If your application requires recognition of strings of digits, you can use either recognition type.

NOTE:
DPR is not currently supported for Voice@Work applications.

For more information on how to incorporate WholeWord speech recognition into a touch-tone application, see *Application Design Guidelines*, 585-310-670.

Getting the Most Out of FlexWord™ Speech Recognition

As discussed in Chapter 5, "Recognizing FlexWord™ Speech Input", FlexWord speech recognition recognizes the caller speaking words from a vocabulary that you define and is therefore specifically tailored to your application. Allowing the caller to speak the option wanted rather than speaking a number assigned to the option can make the interaction more natural and easy to use. Included below are guidelines for getting the most out of FlexWord speech recognition.

Choosing a FlexWord Speech Recognition Vocabulary

Since FlexWord speech recognition offers you the freedom to specify your own custom vocabularies, you have the advantage of making sure your custom vocabularies are easy to use with your application. How you set up your wordlists contributes greatly to the success of your application.

FlexWord speech recognition vocabulary items can be single words or phrases, which are all referred to as "words." The words must be divided into groups called wordlists. A wordlist includes all of the words that can be spoken at a particular prompt. Each application has its own vocabulary, or set of wordlists. At any prompt, only one wordlist can be active at a time. This means that only one wordlist can be under consideration by the recognizer at any one time. The words in each wordlist must be distinct enough from each other to allow the recognizer to work effectively. If your vocabulary contains two or more words that are very similar, each of those words should be on a separate wordlist. The following sections discuss factors to consider when creating your wordlists.

Caller Error

If you find that callers tend to respond with unacceptable words or phrases during a prompt, you can add these words to the wordlist. In your application, you can handle these words as caller errors and reprompt the caller. For example, if your wordlist contains account names and the valid account names are "checking" and "savings," but you find that callers are responding with "loan" you could add "loan" to the wordlist. If the recognizer recognizes that a caller says "loan," you can design your application to say "Sorry. We did not recognize your response. Please choose 'checking' or 'savings'. "

Word Length

Words with more than one syllable are recognized more reliably than shorter words. For example, the phrase "add entry" would be better to use than the word "add." Phrases can be fairly long, although the longer the phrase, the greater the chance of callers forgetting it or making mistakes while speaking.

Word Choice

Choose words and phrases that would occur naturally to your typical caller. One way to determine this is to survey some representative callers and ask what words or phrases they would use for actions in your script. Do not rely on just your knowledge, because your impressions may be different than those of your callers.

Prompting for Input

Because FlexWord speech recognition does not support barge-in, structure your prompts to encourage callers to wait until the end of the prompt before speaking. For example enter, "Please say the department name, now," as opposed to "Now say the department name, please."

Word Sounds

Use words in your wordlists that do not have a similar sound. To the recognizer, vowels are more important than consonants. Avoid using words with similar vowel sounds. Words that differ in only one or two consonants are difficult for the recognizer to distinguish between. For example, the words "connect" and "comment" sound similar to the recognizer.

Synonyms

If you find that callers often use more than one word for the same thing, you may want to add both words to your list.

Examples of FlexWord Speech Recognition Applications

Two examples of applications suited to FlexWord speech recognition are described below.

Applications Requiring Large Wordlists

Applications that require large custom wordlists can benefit from FlexWord speech recognition. In this type of application, the voice prompt cannot enumerate all of the valid responses. As a result, the application must provide sufficient guidance to the caller to facilitate an appropriate response. An example of this type of application would be a name dialer, or automatic call router. By speaking a name, the caller is transferred to the extension of that person.

Menu-Based Applications

Menu-based applications that use words and phrases native to your business environment are suggested for use with FlexWord speech recognition. Your application flows more smoothly if it uses the same words that would occur naturally in an interaction between an agent and the customer. Figure 7-1 shows an example of a menu-based application.

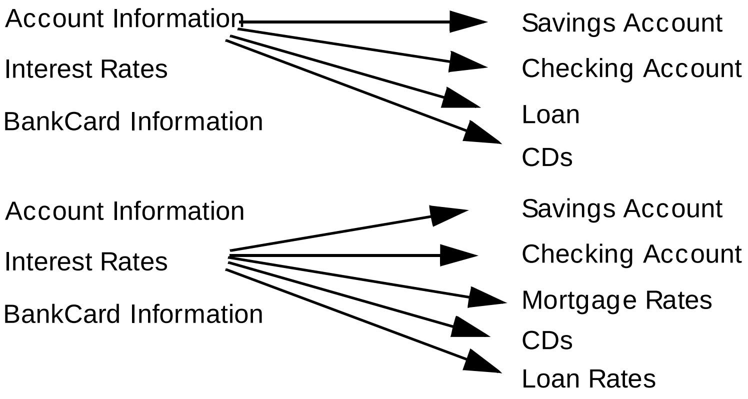


Figure 7-1. Example of Menu-Based Application

Getting the Most out of Dial Pulse Recognition

As discussed in Chapter 6, "Recognizing Dial Pulse Input", the DPR feature recognizes caller pulse inputs from a rotary telephone or from a push button telephone that generates dial pulses. The following tips can help you get the most out of the DPR feature in a Script Builder application.

NOTE:
DPR is currently not supported in Voice@Work applications.

- Design your application so that the first digit is 5 or greater to allow the system to train to the caller's telephone.
- Use the standard recognition types/grammars available.

  Grammars provided for DPR are described in "Using Dial Pulse Recognition", in Chapter 8, "Using Optional Features," of *INTUITY™ CONVERSANT® System Version 7.0 Application Development with Script Builder,* 585-313-206*.*
- Maximize recognition accuracy by initializing training when the application makes a new telephone connection.
- Operate DPR as a simultaneous recognizer, that is, use it with either WholeWord or FlexWord speech recognition.
- Use confirmation to improve script performance.
- When planning an application, if possible, avoid using the digits 1 or 2.
- For simple menu items, use recognition types DP258 or DPR1470 for best performance.

Getting the Most out of Text-to-Speech

The following tips can help you use TTS most effectively in prompts and announcements.

- When TTS prompts are used with touch-tone input, callers can interrupt the prompt just as they can when the prompt is prerecorded.
- Callers can also interrupt TTS prompts with WholeWord speech recognition input. However, when using TTS prompts with FlexWord speech recognition input, barge-in is not available. Therefore, when accepting FlexWord speech recognition input, structure your TTS prompts so that they encourage callers to wait until the end before responding.

NOTE:
TTS is available for US English only.

Using FlexWord Speech Recognition and Text-to-Speech Together

Since FlexWord speech recognition provides the ability to recognize up to 2000 words, your FlexWord speech recognition application needs a way to organize the recognized responses. This can be done by recording all the words in your wordlist, and speaking them back after the database completes a lookup.

However, TTS allows the transaction to provide a more flexible way of speaking the wordlist. TTS allows you to speak the contents of $CI_VALUE (the recognized word on the wordlist) easily. Also, if the wordlist changes, TTS still works— possibly without changes to your application. If you use prerecorded speech to speak the wordlist, you will have to record each time the wordlist changes.

NOTE:
TTS is available for US English only.

Using Text-to-Speech and Prerecorded Speech Together

A Script Builder or Voice@Work application can speak prompts and announcements in prerecorded speech only, TTS only, a combination of prerecorded speech and TTS.

Figure 7-2 shows an example of a Script Builder application for US English that uses both. The structure could be adapted for a Voice@Work application as well.

In the example, recorded speech is used for all phrases except for speaking the customer name. TTS allows the system to speak the contents of the “cust_name” variable, which contains the customer's name. To do this without TTS, you would have to record every possible customer name, save the phrase tag numbers of each, and associate the phrase tag number with the customer account number. Then the system would speak the phrase tag number corresponding to the customer's account number.

```
1.  Prompt and Collect
     Prompt
      Speak With Interrupt
       Phrase:"Please say your 6 digit account number."
        Input
         Caller Input Field: account
         Mode: US_DIG
         Min Number Of Digits: 06
        Max Number Of Digits: 06
        Checklist
         Case: "nnnnnn"
          Speak With Interrupt
            Phrase: "Thank you. Please hold while we verify
            Phrase: "your account number."
            Continue
         Case: "Not On List"
          Reprompt
         Case: "Initial Time-out"
          Reprompt
         Case: "Too Few Digits"
          Reprompt
         Case: "No More Tries"
          Quit
    End Prompt and Collect
2.  Read Table
     Table Name:  ACCOUNTS     Search From Beginning
      Field: account_num = account
3.  Prompt and Collect
     Prompt
      Speak With Interrupt
       Phrase: "Is your name"
       Field: cust_name As A
       Phrase: "Please say yes or no."
      Input
       Mode: US_YN
       Max Number Of Digits: 01
      Checklist
       Case: "Input Ok"
        Continue
       Case: "Initial Time-out"
        Reprompt
       Case: "Too Few Digits"
        Reprompt
       Case: "No More Tries"
        Quit
    End Prompt and Collect
```

Figure 7-2. Example of Application Using Prerecorded Speech and TTS Together

Using WholeWord and FlexWord Speech Recognition Together

WholeWord and FlexWord speech recognition can be used in a single application script. This allows flexibility in what you can ask callers to say. In Script Builder and Voice@Work applications, you can specify that a Prompt and Collect step use either FlexWord or WholeWord speech recognition, depending on what you want to prompt callers to say.

- If you want callers to say “yes” or “no,” a series of digits, or a single digit, choose a WholeWord speech recognition.
- If you want callers to speak a word or phrase from your custom vocabulary, choose a FlexWord speech recognition.

NOTE:
Both recognizers cannot be used in the same prompt.

The example in Figure 7-3 shows a model of a retail application that combines FlexWord wordlists and the WholeWord “yes and no” speech recognition feature. In the example, the word “blue” must appear on both the “INTERIOR” wordlist and the “EXTERIOR” wordlist. Since the word “blue” appears on two wordlists, it counts as two words. If the word “blue” is not on the “INTERIOR” wordlist, this feature will not look to the “EXTERIOR” wordlist to find it. The feature returns the word in the wordlist that it thinks is the most suitable match. Since the feature can only look at one wordlist prompt, each wordlist must contain *all* possible choices for a single prompt.

WholeWord Speech Recognition Versus FlexWord Speech Recognition Accuracy

For most applications, the accuracy of Whole Word recognition is higher than for FlexWord recognition. The following list shows the accuracy rates, arranged in order from highest to lowest:

1. WholeWord speech recognition isolated digit
2. WholeWord speech recognition connected digit
3. FlexWord speech recognition isolated word

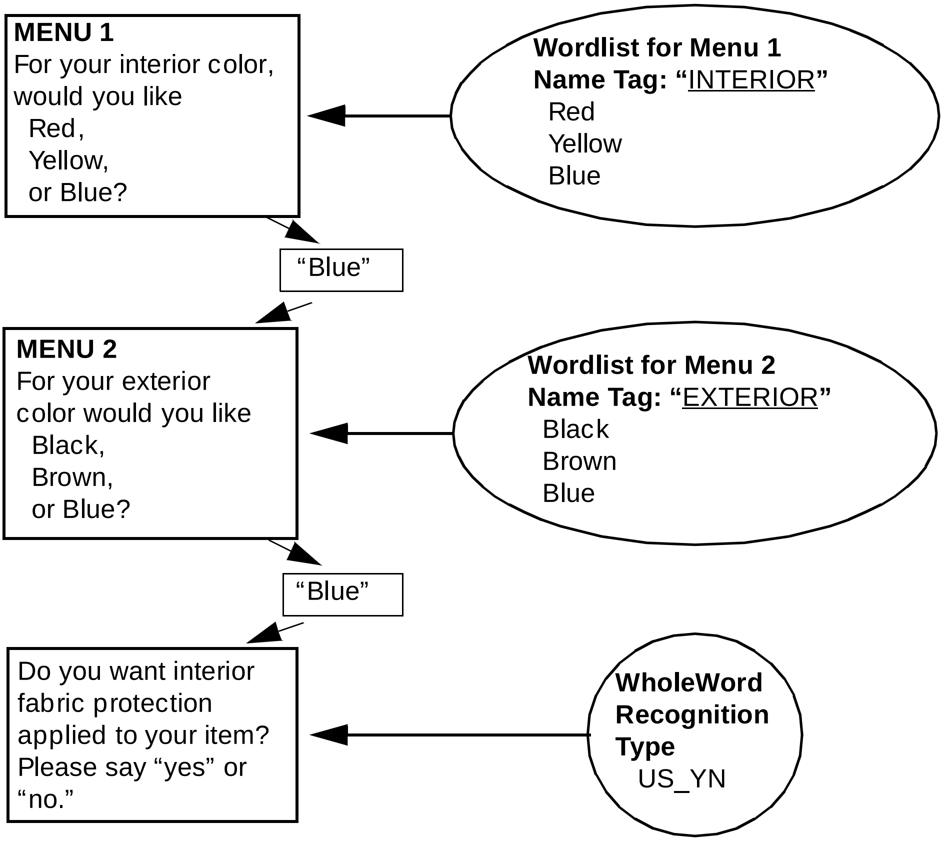

MENU 1
For your interior color, would you like
Red,
Yellow,
or Blue?
Wordlist for Menu 1
Name Tag: "INTERIOR"
Red
Yellow
Blue
"Blue"
MENU 2
For your exterior color would you like
Black,
Brown,
or Blue?
Wordlist for Menu 2
Name Tag: "EXTERIOR"
Black
Brown
Blue
"Blue"
Do you want interior fabric protection applied to your item? Please say "yes" or "no."
WholeWord Recognition Type
US_YN

Figure 7-3. Illustration of the Use of FlexWord and WholeWord Speech Recognition in a Single Application Script

Comparison of Recognition Types

Table 7-1 summarizes the similarities and differences between WholeWord speech recognition, FlexWord speech recognition, and DPR.

Table 7-1. Comparison of Recognition Types

WholeWord Speech Recognition	**FlexWord Speech Recognition**	**Dial Pulse Recognition**
Word-based	Phoneme-based	Pulse-based
Requires data collection in model building (for vocabulary words other than the digits "zero" through "nine," commonly used synonyms for those words, and the words "yes" and "no")	No data collection required in model building	No data collection required in model building
Connected digits	Single word or phrase	Connected digits
Standard and custom grammars	Customized wordlists	Standard grammars
Barge-in supported	Barge-in not supported	Barge-in not supported
Phrase screening supported	Phrase screening supported	Phrase screening supported
Limited vocabulary	2000-word vocabulary	Limited vocabulary
Word spotting supported	Word spotting supported	Word spotting not supported

Enhanced Basic Speech Formats

A

Overview

This appendix provides a complete list of the formats included in the Enhanced Basic Speech package for each available language.

Purpose

This appendix is designed to be used as a reference when you create an application using Enhanced Basic Speech phrases.

Languages Available

Table A-1 lists the languages for which Enhanced Basic Speech is available and shows the gender of the speaking voice (female or male), the phrase pool name, and the abbreviation for each language.

Table A-1. Enhanced Basic Speech Languages

Enhanced Basic Speech Version 6.0	Gender	Phrase Pool Name	Abbreviation
Australian English	female	AU_English	ebsAE
Brazilian Portuguese	female	BR_Portug	ebsBP
Canadian French	female	Can_French	ebsCF
Cantonese Chinese	female	Cantonese	ebsCT
Castilian Spanish	female	Castil_Span	ebsCS
Dutch	female	Dutch	ebsDT
French	female	French	ebsFR
German	female	German	ebsGR
Hindi	female	Hindi	ebsHD
Indonesian	female	Indonesian	ebsIN
Italian	female	Italian	ebsIT
Japanese	female	Japanese	ebsJP
Latin-American Spanish	female	LA_Spanish	ebsLS
Malay	female	Malay	ebsMA
Mandarin Chinese	female	Mandarin	ebsMD
Thai	female	Thai	ebsTH
United Kingdom English	female	UK_English	ebsUE
United States English	female	US_English	ebsUS
	male		ebsUSM

Enhanced Basic Speech Formats

The following tables (Table A-2 through Table A-19) provide a complete listing of Enhanced Basic Speech formats for all the languages listed in Table A-1.

Inflections

Three types of inflection exist with speech formats:

- Rising inflection is used in questions and at the beginning of some words.
- Medial inflection is used in the middle of a word or statement. For example, when you speak the number “101,” the “0” has medial inflection.
- Falling inflection is usually used at the end of a word or statement. For example, when you speak “2.0,” the “0” is spoken with falling inflection.

Grammatical Gender

Several languages require that a number be spoken differently depending on whether the noun the number refers to is grammatically masculine, feminine, or neutral in gender. Tables for these languages include formats that enable you to tell the system which form to use.

Converting Applications to Other Languages

An application containing Enhanced Basic Speech formats can be converted to other languages by changing the language options. However if the original application uses formats not available for the target languages, then the conversion will not be exact. Therefore, advance planning is recommended for applications that may be converted. One strategy is to create the original application using only formats available for all the target languages, create duplicate applications for the target languages, and adjust the formats on a per-application basis. Formats available for all languages are:

- D
- T
- C
- N

NOTE:
These formats, while available for all languages, may be defined differently from language to language. For example, T (time) is defined as THMAM for some languages, but as TH24M for others.

Australian English Enhanced Basic Speech Formats

Table A-2 shows the Australian English Enhanced Basic Speech formats.

Table A-2. Australian English Enhanced Basic Speech Formats

Format	Description
DMDYY	Date – 2-digit month, day, 4-digit year
DMDY	Date – 2-digit month, day, 2-digit year
DMD	Date – 2-digit month, day, but no year
DMSPDYY	Date – spoken month, 2-digit day, 4-digit year
DMSPDY	Date – spoken month, 2-digit day, 2-digit year
DMSPD	Date – spoken month, 2-digit day, but no year
D	Date – default DMSPDY
TH24M	Time – 24-hour military time
THMAM	Time – hour, minute AM/PM
T	Time – default THMAM
Nrmf	Number – total inflection
Nmmf	Number – falling inflection
Nrmm	Number – rising inflection
Nmmm	Number – no inflection
N	Number – default Nmmm
ND1	Decimal number – 1 decimal place
ND2	Decimal number – 2 decimal places
ND3	Decimal number – 3 decimal places
ND4	Decimal number – 4 decimal places
ND5	Decimal number – 5 decimal places
ND6	Decimal number – 6 decimal places
ND7	Decimal number – 7 decimal places
ND8	Decimal number – 8 decimal places
ND9	Decimal number – 9 decimal places

Continued on next page

Table A-2. Australian English Enhanced Basic Speech Formats — *Continued*

Format	Description
N$	Number – whole dollars
N$D0	Number – whole dollars
N$D2	Number – dollars and cents
N$D1	Number – dollars with 1 decimal place
N$D3	Number – dollars with 3 decimal places
N$D4	Number – dollars with 4 decimal places
N$D5	Number – dollars with 5 decimal places
N$D6	Number – dollars with 6 decimal places
N$D7	Number – dollars with 7 decimal places
N$D8	Number – dollars with 8 decimal places
N$D9	Number – dollars with 9 decimal places
NX	Speak the phrase with the specified phrase tag number. Also used to speak packed talkfile numbers and phrase numbers.
Crmf	Characters – spoken individually, total inflection
Crmm	Characters – spoken individually, rising inflection
Cmmf	Characters – spoken individually, falling inflection
Cmmm	Characters – spoken individually, no inflection
C	Characters – default Cmmm

Brazilian Portuguese Enhanced Basic Speech Formats

Table A-3 shows the Brazilian Portuguese Enhanced Basic Speech formats. Brazilian Portuguese requires that a number be spoken differently depending on whether the noun the number refers to is grammatically masculine, feminine, or neutral in gender. Table A-3 includes formats that enable you to tell the system which form to use.

Table A-3. Brazilian Portuguese Enhanced Basic Speech Formats

Format	Description
DMSPDYY	Date – spoken month, 2-digit day, 4-digit year
DMSPDY	Date – spoken month, 2-digit day, 2-digit year
DMSPD	Date – spoken month, 2-digit day, but no year
D	Date – default DMSPDYY
DMSPfYY	Date – same as DMSPDYY but with falling inflection
DMSPfY	Date – same as DMSPDY but with falling inflection
DMSPf	Date – same as DMSPD but with falling inflection
THMAM	Time – hour, minute AM/PM
TH24M	Time – military time - no seconds
THMAMf	Time – hour, minute AM/PM, falling inflection
TH24Mf	Time – military time - no seconds, falling inflection
T	Time – default THMAM
N	Number – total inflection
Nmf	Number – falling inflection
Nrm	Number – rising inflection
Nrmf	Number – total inflection
NM	Number – following masculine noun
NF	Number – following feminine noun
NN	Number – following neutral noun
ND0	Decimal number – no decimal places
ND1	Decimal number – 1 decimal place

Continued on next page

Table A-3. Brazilian Portuguese Enhanced Basic Speech Formats — *Continued*

Format	Description
ND2	Decimal number – 2 decimal places
ND3	Decimal number – 3 decimal places
ND4	Decimal number – 4 decimal places
ND5	Decimal number – 5 decimal places
ND6	Decimal number – 6 decimal places
ND7	Decimal number – 7 decimal places
ND8	Decimal number – 8 decimal places
ND9	Decimal number – 9 decimal places
N$D2	Decimal number – reais and centavos
NX	Speak the phrase with the specified phrase tag number. Also used to speak packed talkfile numbers and phrase numbers.
Crmf	Characters – spoken individually, total inflection
Crmm	Characters – spoken individually, rising inflection
Cmmf	Characters – spoken individually, falling inflection
Cmmm	Characters – spoken individually, no inflection
C	Characters – default Cmmm

Canadian French Enhanced Basic Speech Format s

Table A-4 shows the Canadian French Enhanced Basic Speech formats. Canadian French requires that a number be spoken differently depending on whether the noun the number refers to in grammatically masculine, feminine, or neutral in gender. Table A-4 includes formats that enable you to tell the system which form to use.

Table A-4. Canadian French Enhanced Basic Speech Formats

Format	Description
DMSPDYY	Date – spoken month, 2-digit day, 4-digit year
DMSPDY	Date – spoken month, 2-digit day, 2-digit year
DMSPD	Date – spoken month, 2-digit day, but no year
D	Date – default DMSPDYY
DMSPfYY	Date – same as DMSPDYY, but with falling inflection
DMSPfY	Date – same as DMSPDY, but with falling inflection
DMSPf	Date – same as DMSPD, but with falling inflection
THMAM	Time – hour, minute AM/PM
THMAMf	Time – hour, minute AM/PM, falling inflection
T	Time – default TH24M
TH24M	Time – military time, no seconds
TH24Mf	Time – military time – no seconds, falling inflection
THMAMPD	Pacific Daylight Saving Time – hour, minute AM/PM
TH24MPD	Pacific Daylight Saving Time – military time – no seconds
THMAMPDf	PDTime – hour, minute AM/PM, falling inflection
TH24MPDf	PDTime – military time – no seconds, falling inflection
TPD	PDTime – default TH24M
THMAMPS	Pacific Stand Time – hour, minute AM/PM
TH24MPS	Pacific Stand Time – military time – no seconds
THMAMPSf	PSTime – hour, minute AM/PM, falling inflection
TH24MPSf	PSTime – military time – no seconds, falling inflection

Continued on next page

Table A-4. Canadian French Enhanced Basic Speech Formats — *Continued*

Format	Description
TPS	PSTime – default TH24M
THMAMMD	Mountain Daylight Saving Time – hour, minute AM/PM
TH24MMD	Mountain Daylight Saving Time – military time – no seconds
THMAMMDf	MDTime – hour, minute AM/PM, falling inflection
TH24MMDf	MDTime – military time – no seconds, falling inflection
TMD	MDTime – default TH24M
THMAMMS	Mountain Stand Time – hour, minute AM/PM
TH24MMS	Mountain Stand Time – military time – no seconds
THMAMMSf	MSTime – hour, minute AM/PM, falling inflection
TH24MMSf	MSTime – military time – no seconds, falling inflection
TMS	MSTime – default TH24M
THMAMCD	Central Daylight Saving Time – hour, minute AM/PM
TH24MCD	Central Daylight Saving Time – military time – no seconds
THMAMCDf	CDTime – hour, minute AM/PM, falling inflection
TH24MCDf	CDTime – military time – no seconds, falling inflection
TCD	CDTime – default TH24M
THMAMCS	Central Stand Time – hour, minute AM/PM
TH24MCS	Central Stand Time – military time – no seconds
THMAMCSf	CSTime – hour, minute AM/PM, falling inflection
TH24MCSf	CSTime – military time – no seconds, falling inflection
TCS	CSTime – default TH24M
THMAMAD	Atlantic Daylight Saving Time – hour, minute AM/PM
TH24MAD	Atlantic Daylight Saving Time – military time – no seconds
THMAMADf	ADTime – hour, minute AM/PM, falling inflection
TH24MADf	ADTime – military time – no seconds, falling inflection
TAD	ADTime – default TH24M
THMAMAS	Atlantic Stand Time – hour, minute AM/PM
TH24MAS	Atlantic Stand Time – military time – no seconds

Continued on next page

Table A-4. Canadian French Enhanced Basic Speech Formats — *Continued*

Format	Description
THMAMASf	ASTime – hour, minute AM/PM, falling inflection
TH24MASf	ASTime – military time – no seconds, falling inflection
TAS	ASTime – default TH24M
N	Number – total inflection
Nmf	Number – falling inflection
Nrm	Number – rising inflection
Nrmf	Number – total inflection
NM	Number – following masculine noun
NF	Number – following feminine noun
NN	Number – following neutral noun
ND0	Decimal number – no decimal place, “virgule zero”
ND1	Decimal number – 1 decimal place
ND2	Decimal number – 2 decimal places
ND3	Decimal number – 3 decimal places
ND4	Decimal number – 4 decimal places
ND5	Decimal number – 5 decimal places
ND6	Decimal number – 6 decimal places
ND7	Decimal number – 7 decimal places
ND8	Decimal number – 8 decimal places
ND9	Decimal number – 9 decimal places
NDF0	Decimal number – no decimal places
NDF1	Decimal number – 1 decimal, decimal part spoken as number
NDF2	Decimal number – 2 decimal places
NDF3	Decimal number – 3 decimal places
NDF4	Decimal number – 4 decimal places
NDF5	Decimal number – 5 decimal places
NDF6	Decimal number – 6 decimal places
NDF7	Decimal number – 7 decimal places

Continued on next page

Table A-4. Canadian French Enhanced Basic Speech Formats — *Continued*

Format	Description
NDF8	Decimal number – 8 decimal places
NDF9	Decimal number – 9 decimal places
N$D2	Number – dollars and cents
N$	Number – dollars and cents
N$F	Number – whole dollars
N$F0	Number – whole dollars
N$F1	Number – dollars and cents
N$F2	Number – dollars with 1 decimal place
N$F3	Number – dollars with 3 decimal places
N$F4	Number – dollars with 4 decimal places
N$F5	Number – dollars with 5 decimal places
N$F6	Number – dollars with 6 decimal places
N$F7	Number – dollars with 7 decimal places
N$F8	Number – dollars with 8 decimal places
N$F9	Number – dollars with 9 decimal places
NX	Speak the phrase with the specified phrase tag number. Also used to speak packed talkfile numbers and phrase numbers.
Crmf	Characters – spoken individually, total inflection
Crmm	Characters – spoken individually, rising inflection
Cmmf	Characters – spoken individually, falling inflection
Cmmm	Characters – spoken individually, no inflection
C	Characters – default Cmmm

Cantonese Chinese Enhanced Basic Speech Formats

FTable A-5 shows the Cantonese Chinese Enhanced Basic Speech formats.

NOTE:
For US currency the word “maygum” is spoken first to indicate United States currency.

Table A-5. Cantonese Chinese Enhanced Basic Speech Formats

Format	Description
DYMDD	Date – 4-digit year, month and day
DMD	Date month followed by day
D	Date – default DTTMD
THMAM	Time – same as T
T	Time – time, period followed by time
N	Number
ND0	Decimal number – no decimal places
ND1	Decimal number – 1 decimal places
ND2	Decimal number – 2 decimal places
ND3	Decimal number – 3 decimal places
ND4	Decimal number – 4 decimal places
ND5	Decimal number – 5 decimal places
ND6	Decimal number – 6 decimal places
ND7	Decimal number – 7 decimal places
ND8	Decimal number – 8 decimal places
ND9	Decimal number – 9 decimal places
N$D0	Number – whole dollars
N$D2	Number – dollars and cents
N$D1	Number – dollars with 1 decimal place
N$D3	Number – dollars with 3 decimal places
N$D4	Number – dollars with 4 decimal places

Continued on next page

Table A-5. Cantonese Chinese Enhanced Basic Speech Formats — *Continued*

Format	Description
N$D5	Number – dollars with 5 decimal places
N$D6	Number – dollars with 6 decimal places
N$D7	Number – dollars with 7 decimal places
N$D8	Number – dollars with 8 decimal places
N$D9	Number – dollars with 9 decimal places
NYD0	Number – whole yuan
NYD1	Number – yuan with 1 decimal place
NYD2	Number – yuan, miao and seen
NYD3	Number – yuan with 3 decimal places
NYD4	Number – yuan with 4 decimal places
NYD5	Number – yuan with 5 decimal places
NYD6	Number – yuan with 6 decimal places
NYD7	Number – yuan with 7 decimal places
NYD8	Number – yuan with 8 decimal places
NYD9	Number – yuan with 9 decimal places
NX	Speak the phrase with the specified phrase tag number. Also used to speak packed talkfile numbers and phrase numbers.
C	Characters – default Cmmm
Cmmm	Characters – spoken individually, no inflection

Castilian Spanish Enhanced Basic Speech Formats

Table A-6 shows the Castilian Spanish Enhanced Basic Speech formats. Castilian Spanish requires that a number be spoken differently depending on whether the noun the number refers to is grammatically masculine, feminine, or neutral in gender. Table A-6 includes formats that enable you to tell the system which form to use.

Table A-6. Castilian Spanish Enhanced Basic Speech Formats

Format	Description
DMSPDYY	Date – spoken month, 2-digit day, 4-digit year
DMSPDY	Date – spoken month, 2-digit day, 2-digit year
DMSPD	Date – spoken month, 2-digit day
D	Date – default DMSPDYY
THMAM	Time – hour, minute AM/PM
TH24M	Time – military time-no seconds
T	Time – default THMAM
NM	Number – following masculine noun
NF	Number – following feminine noun
NN	Number – following neutral noun
N	Number – default NN
ND0	Number – no decimal places
ND1	Number – 1 decimal place
ND2	Number – 2 decimal places
ND3	Number – 3 decimal places
ND4	Number – 4 decimal places
ND5	Number – 5 decimal places
ND6	Number – 6 decimal places
ND7	Number – 7 decimal places
ND8	Number – 8 decimal places
ND9	Number – 9 decimal places

Continued on next page

Table A-6. Castilian Spanish Enhanced Basic Speech Formats — *Continued*

Format	Description
N$	Number – whole pesetas
N$D0	Number – whole pesetas
N$D2	Number – pesetas and centimos
NX	Speak the phrase with the specified phrase tag number. Also used to speak packed talkfile numbers and phrase numbers.
Crmf	Characters – spoken individually, total inflection
Crmm	Characters – spoken individually, rising inflection
Cmmf	Characters – spoken individually, falling inflection
Cmmm	Characters – spoken individually, no inflection
C	Characters – default Cmmm

Dutch Enhanced Basic Speech Formats

Table A-7 shows the Dutch Enhanced Basic Speech formats.

Table A-7. Dutch Enhanced Basic Speech Formats

Format	Description
DMSPDYY	Date – spoken month, 2-digit day, 4-digit year
DMSPDY	Date – spoken month, 2-digit day, 2-digit year
DMSPD	Date – spoken month, 2-digit day, but no year
D	Date – default DMSPDYY
TH24M	Time – military time - no seconds
T	Time – default TH24M
N	Number – default NN
ND0	Decimal number – no decimal place
ND1	Decimal number – 1 decimal place
ND2	Decimal number – 2 decimal places
ND3	Decimal number – 3 decimal places
ND4	Decimal number – 4 decimal places
ND5	Decimal number – 5 decimal places
ND6	Decimal number – 6 decimal places
ND7	Decimal number – 7 decimal places
ND8	Decimal number – 8 decimal places
ND9	Decimal number – 9 decimal places
N$	Number – whole gulden
N$D0	Number – whole gulden
N$D2	Number – gulden and centen
NX	Speak the phrase with the specified phrase tag number. Also used to speak packed talkfile numbers and phrase numbers.
Crmf	Characters – spoken individually, total inflection
Crmm	Characters – spoken individually, rising inflection

Continued on next page

Table A-7. Dutch
Enhanced Basic Speech Formats — *Continued*

Format	Description
Cmmf	Characters – spoken individually, falling inflection
Cmmm	Characters – spoken individually, no inflection
C	Characters – default Cmmm

French Enhanced Basic Speech Formats

Table A-8 shows the French Enhanced Basic Speech formats. French requires that a number be spoken differently depending on whether the noun the number refers to is grammatically masculine, feminine, or neutral in gender. Table A-8 includes formats that enable you to tell the system which form to use.

Table A-8. French Enhanced Basic Speech Formats

Format	Description
DMSPDYY	Date – spoken month, 2-digit day, 4-digit year
DMSPDY	Date – spoken month, 2-digit day, 2-digit year
DMSPD	Date – spoken month, 2-digit day, but no year
D	Date – default DMSPDYY
DMSPfYY	Date – same as DMSPDYY but with falling inflection
DMSPfY	Date – same as DMSPDY but with falling inflection
DMSPf	Date – same as DMSPD but with falling inflection
DMSPlYY	Date – same as DMSPDYY but with “le” before
DMSPlY	Date – same as DMSPDY but with “le”
DMSPl	Date – same as DMSPD but with “le”
TH24M	Time – military time - no seconds
TH24Mf	Time – military time - no seconds, falling inflection
T	Time – default TH24M
N	Number – total inflection
Nmf	Number – falling inflection
Nrm	Number – rising inflection
Nrmf	Number – total inflection
NM	Number – following masculine noun
NF	Number – following feminine noun
NN	Number – following neutral noun
ND0	Decimal number – no decimal place, “virgule zero”
ND1	Decimal number – 1 decimal place

Continued on next page

Table A-8. French Enhanced Basic Speech Formats — *Continued*

Format	Description
ND2	Decimal number – 2 decimal places
ND3	Decimal number – 3 decimal places
ND4	Decimal number – 4 decimal places
ND5	Decimal number – 5 decimal places
ND6	Decimal number – 6 decimal places
ND7	Decimal number – 7 decimal places
ND8	Decimal number – 8 decimal places
ND9	Decimal number – 9 decimal places
NDF0	Dec Number – no decimal place
NDF1	Dec Number – 1 decimal, decimal part spoken as number
NDF2	Decimal number – 2 decimal places
NDF3	Decimal number – 3 decimal places
NDF4	Decimal number – 4 decimal places
NDF5	Decimal number – 5 decimal places
NDF6	Decimal number – 6 decimal places
NDF7	Decimal number – 7 decimal places
NDF8	Decimal number – 8 decimal places
NDF9	Decimal number – 9 decimal places
N$D2	Number – francs and centimes
N$	Number – francs and centimes
N$F	Number – whole number of francs
N$F0	Number of francs with no decimal place
N$F1	Number of francs with 1 decimal place
N$F2	Number of francs with 2 decimal places
N$F3	Number of francs with 3 decimal places
N$F4	Number of francs with 4 decimal places
N$F5	Number of francs with 5 decimal places
N$F6	Number of francs with 6 decimal places

Continued on next page

Table A-8. French Enhanced Basic Speech Formats — *Continued*

Format	Description
N$F7	Number of francs 7 decimal places
N$F8	Number of francs 8 decimal places
N$F9	Number of francs 9 decimal places
NX	Speak the phrase with the specified phrase tag number. Also used to speak packed talkfile numbers and phrase numbers.
Crmf	Characters – spoken individually, total inflection
Crmm	Characters – spoken individually, rising inflection
Cmmf	Characters – spoken individually, falling inflection
Cmmm	Characters – spoken individually, no inflection
C	Characters – default Cmmm

German Enhanced Basic Speech Formats

Table A-9 shows the German Enhanced Basic Speech formats. German requires that a number be spoken differently depending on whether the noun the number refers to is grammatically masculine, feminine, or neutral in gender. Table A-9 includes formats that enable you to tell the system which form to use.

Table A-9. German Enhanced Basic Speech Formats

Format	Description
DMSPDYY	Date – spoken month, 2-digit day, 4-digit year
DMSPDY	Date – spoken month, 2-digit day, 2-digit year
DMSPD	Date – spoken month, 2-digit day
D	Date – default DMSPDYY
TH24M	Time – military time-no seconds
T	Time – default THMAM
NM	Number – following masculine noun
NF	Number – following feminine noun
NN	Number – following neutral noun
N	Number – default NN
ND0	Decimal number – no decimal place
ND1	Decimal number – 1 decimal place
ND2	Decimal number – 2 decimal places
ND3	Decimal number – 3 decimal places t
ND4	Decimal number – 4 decimal places
ND5	Decimal number – 5 decimal places
ND6	Decimal number – 6 decimal places
ND7	Decimal number – 7 decimal places
ND8	Decimal number – 8 decimal places
ND9	Decimal number – 9 decimal places
N$	Number – whole marks
N$D0	Number – whole marks

Continued on next page

Table A-9. German Enhanced Basic Speech Formats — *Continued*

Format	Description
N$D2	Number – marks and pfennig
N$DM	Number – whole deutsch marks
N$DM0	Number – whole deutsch marks
N$DM2	Number – whole deutsch marks and pfennig
NX	Speak the phrase with the specified phrase tag number. Also used to speak packed talkfile numbers and phrase numbers.
Crmf	Characters – spoken individually, total inflection
Crmm	Characters – spoken individually, rising inflection
Cmmf	Characters – spoken individually, falling inflection
Cmmm	Characters – spoken individually, no inflection
C	Characters – default Cmmm

Hindi Enhanced Basic Speech Formats

Table A-10 shows the Hindi Enhanced Basic Speech formats.

Table A-10. Hindi Enhanced Basic Speech Formats

Format	Description
DDMSP	Date – 2-digit day, spoken month
DDMSPYY	Date – 2-digit day, spoken month, 4-digit year
D	Date – default is DDMSPYY
T	Time – time period followed by time
THMAM	Time – same as T
N	Number
ND0	Number – no decimal place
ND1	Decimal number – 1 decimal place
ND2	Decimal number – 2 decimal places
ND3	Decimal number – 3 decimal places
ND4	Decimal number – 4 decimal places t
ND5	Decimal number – 5 decimal places
ND6	Decimal number – 6 decimal places
ND7	Decimal number – 7 decimal places
ND8	Decimal number – 8 decimal places
ND9	Decimal number – 9 decimal places
N$D0	Number – whole dollars
N$D1	Number – dollars with 1 decimal place
N$D2	Number – dollars and cents
N$D3	Number – dollars with 3 decimal places
N$D4	Number – dollars with 4 digits decimal places
N$D5	Number – dollars with 5 digits decimal places
N$D6	Number – dollars with 6 digits decimal places
N$D7	Number – dollars with 7 digits decimal places

Continued on next page

Table A-10. Hindi Enhanced Basic Speech Formats — *Continued*

Format	Description
N$D8	Number – dollars with 8 decimal places
N$D9	Number – dollars with 9 with decimal places
NRD0	Number – rupees
NRD1	Number – rupees with 1 decimal place
NRD2	Number – rupees with 2 decimal places
NRD3	Number – rupees with 3 decimal places
NDR4	Number – rupees with 4 decimal places
NDR5	Number – rupees with 5 decimal places
NDR6	Number – rupees with 6 decimal places
NDR7	Number – rupees with 7 decimal places
NDR8	Number – rupees with 8 decimal places
NDR9	Number – rupees with 9 decimal places
NX	Speak the phrase with the specified phrase tag number. Also used to speak packed talkfile numbers and phrase numbers.
Cmmm	Characters – spoken individually, no inflection
C	Characters – default Cmmm

Indonesian Enhanced Basic Speech Formats

Table A-11 shows the Indonesian Enhanced Basic Speech formats.

Table A-11. Indonesian Enhanced Basic Speech Formats

Format	Description
N	Number – speak an integer
ND0	Decimal number – no decimal place
ND1	Decimal number – 1 decimal place
ND2	Decimal number – 2 decimal places
ND2W	Decimal number – 2 digits after decimal point spoken as number
ND3	Decimal number – 3 decimal places
ND4	Decimal number – 4 decimal places
ND5	Decimal number – 5 decimal places
ND6	Decimal number – 6 decimal places
ND7	Decimal number – 7 decimal places
ND8	Decimal number – 8 decimal places
ND9	Decimal number – 9 decimal places
DMSPDYY	Date – spoken month, 2-digit day, 4-digit year
D	Date – default DMSPDYY
THMAM	Time – 12 hour, minute, no second
TH24M	Time – 24 hour, minute, no second
T	Time – default THMAM
N$	Money – default N$R
N$R	Indonesian currency, whole rupiah
N$D0	US$, whole dollars
N$D2	US$, dollars and cents
Cm	English characters – spoken individually, medial inflection
Cf	English characters – spoken individually, falling inflection
Cr	English characters – spoken individually, rising inflection

Continued on next page

Table A-11. Indonesian Enhanced Basic Speech Formats — *Continued*

Format	Description
CIm	Indonesian characters – spoken individually, medial inflection
CIf	Indonesian characters – spoken individually, falling inflection
CIr	Indonesian characters – spoken individually, rising inflection
C	Characters – default CIm

Italian Enhanced Basic Speech Formats

Table A-12 shows the Italian Enhanced Basic Speech formats. Italian requires that a number be spoken differently depending on whether the noun the number refers to is grammatically masculine, feminine, or neutral in gender. Table A-12 includes formats that enable you to tell the system which form to use.

Table A-12. Italian Enhanced Basic Speech Formats

Format	Description
DDMSPYY	Date – day, spoken month, 4-digit year
DDMSPY	Date – day, spoken month, 2-digit year
DDMSP	Date – spoken month, 2-digit day, no year
D	Date – default DDMSPYYf
DDMSPYYf	Date – same as DDMSPYY but with falling inflection
DDMSPYf	Date – same as DDMSPY but with falling inflection
DDMSPf	Date – same as DDMSP but with falling inflection
THMAM	Time – hour, minute AM/PM
TH24M	Time – military time, no seconds
THMAMf	Time – hour, minute AM/PM, with falling inflection
TH24Mf	Time – military time, no seconds, with falling inflection
T	Time – default TH24Mf
N	Number – medial inflection
Nmf	Number – falling inflection
Nrm	Number – rising inflection
Nrmf	Number – total inflection
NN	Number – neutral
NF	Number – female
NM	Number – male
ND0	Decimal number – no decimal place
ND1	Decimal number – 1 decimal place
ND2	Decimal number – 2 decimal places

Continued on next page

Table A-12. Italian Enhanced Basic Speech Formats — *Continued*

Format	Description
ND3	Decimal number – 3 decimal places
ND4	Decimal number – 4 decimal places
ND5	Decimal number – 5 decimal places
ND6	Decimal number – 6 decimal places
ND7	Decimal number – 7 decimal places
ND8	Decimal number – 8 decimal places
ND9	Decimal number – 9 decimal places
N$	Money – lire, medial inflection
N$f	Money – lire, falling inflection
Crmf	Characters – spoken individually, total inflection
Crmm	Characters – spoken individually, rising inflection
Cmmf	Characters – spoken individually, falling inflection
Cmmm	Characters – spoken individually, no inflection
C	Characters – default Cmmm

Japanese Enhanced Basic Speech Formats

Table A-13 shows the Japanese Enhanced Basic Speech formats.

Table A-13. Japanese Enhanced Basic Speech Formats

Format	Description
DMSPDYY	Date – spoken 4-digit year, month, 2-digit day
DMSPDY	Date – spoken 4-digit year, month, 2-digit day, day of week
D	Date – default DMSPDYY
TH24M	Time – AM/PM hour, minute
N	Number – total inflection
ND0	Decimal number – no decimal place
ND1	Decimal number – 1 decimal place
ND2	Decimal number – 2 decimal places
ND3	Decimal number – 3 decimal places
ND4	Decimal number – 4 decimal places
ND5	Decimal number – 5 decimal places
ND6	Decimal number – 6 decimal places
ND7	Decimal number – 7 decimal places
ND8	Decimal number – 8 decimal places
ND9	Decimal number – 9 decimal places
N$	Whole yen
N$D2	Yen – 2 decimal places
NX	Speak the phrase with the specified phrase tag number. Also used to speak packed talkfile numbers and phrase numbers.
C	Character – default Cmmm

Latin-American Spanish Enhanced Basic Speech Format Set

Table A-14 shows the Latin-American Enhanced Basic Speech formats. Latin-American Spanish requires that a number be spoken differently depending on whether the noun the number refers to is grammatically masculine, feminine, or neutral in gender. Table A-14 includes formats that enable you to tell the system which form to use.

Table A-14. Latin-American Spanish Enhanced Basic Speech Format Set

Format	Description
DMSPDYY	Date – spoken month, 2-digit day, 4-digit year
DMSPDY	Date – spoken month, 2-digit day, 2-digit year
DMSPD	Date – spoken month, 2-digit day
D	Date – default DMSPDYY
THMAM	Time – hour, minute AM/PM
TH24M	Time – military time
T	Time – default THMAM
NM	Number – following masculine noun
NF	Number – following feminine noun
NN	Number – following neutral noun
N	Number – default NN
ND0	Number – No decimal place
ND1	Decimal number – 1 decimal place
ND2	Decimal number – 2 decimal places
ND3	Decimal number – 3 decimal places
ND4	Decimal number – 4 decimal places
ND5	Decimal number – 5 decimal places
ND6	Decimal number – 6 decimal places
ND7	Decimal number – 7 decimal places
ND8	Decimal number – 8 decimal places
ND9	Decimal number – 9 decimal places

Continued on next page

Table A-14. Latin-American Spanish Enhanced Basic Speech Format Set — *Continued*

Format	Description
Cm	Character – default Cm
Ctry	Characters – for test
N$	Number – whole pesos
N$D0	Number – whole pesos
N$D2	Number – pesos and centavos
NDOL$	Number – whole dollars
NDOL$D0	Number – whole dollars
NDOL$D2	Number – dollars and centavos
NPN$	Number – whole pesos nuevos
NCOL$	Number – whole colones
NCOL$D0	Number – whole colones
NCOL$D2	Number – colones and centavos
NSUC$	Number – whole sucres
NSUC$D0	Number – whole sucres
NSUC$D2	Number – sucres and centavos
NQUE$	Number – whole quetzales
NQUE$D0	Number – whole quetzales
NQUE$D2	Number – quetzales and centavos
NLEM$	Number – whole lempiras
NLEM$D0	Number – whole lempiras
NLEM$D2	Number – lempiras and centavos
NCOR$	Number – whole cordobas
NCOR$D0	Number – whole cordobas
NCOR$D2	Number – cordobas and centavos
NGUA$	Number – whole guaranis
NGUA$D0	Number – whole guaranis
NGUA$D2	Number – guaranis and centimos
NSOL$	Number – whole soles

Continued on next page

Table A-14. Latin-American Spanish Enhanced Basic Speech Format Set — *Continued*

Format	Description
NSOL$D0	Number – whole soles
NSOL$D2	Number – soles and centavos
NBOL$	Number – whole bolivares
NBOL$D0	Number – whole bolivares
NBOL$D2	Number – bolivares and centimos
NPTA$	Number – whole pesetas
NPTA$D0	Number – whole pesetas
NPTA$D2	Number – pesetas and centimos
NX	Speak the phrase with the specified phrase tag number. Also used to speak packed talkfile numbers and phrase numbers.

Malay Enhanced Basic Speech Formats

Table A-15 shows the Malay Enhanced Basic Speech formats.

Table A-15. Malay Enhanced Basic Speech Formats

Format	Description
N	Number – speak an integer
ND0	Decimal number – no decimal place
ND1	Decimal number – 1 decimal place
ND2	Decimal number – 2 decimal places
ND2W	Decimal number – 2 digits after decimal point spoken as number
ND3	Decimal number – 3 decimal places
ND4	Decimal number – 4 decimal places
ND5	Decimal number – 5 decimal places
ND6	Decimal number – 6 decimal places
ND7	Decimal number – 7 decimal places
ND8	Decimal number – 8 decimal places
ND9	Decimal number – 9 decimal places
DMSPDYY	Date – spoken month, 2-digit day, 4-digit year
D	Date – default DMSPDYY
THMAM	Time – 12 hour, minute, no second
TH24M	Time – 24 hour, minute, no second
T	Time – default THMAM
N$	Money – default is NR$
N$R	Malaysia currency, whole ringgit
N$R2	Malaysia currency, ringgit and cents
N$D0	US dollars – whole dollars
N$D2	US dollars – dollars and cents
Cm	English characters – spoken individually, medial inflection

Continued on next page

Table A-15. Malay Enhanced Basic Speech Formats — *Continued*

Format	Description
Cf	English characters – spoken individually, falling inflection
Cr	English characters – spoken individually, rising inflection
C	Characters – system default Clm

Mandarin Chinese Enhanced Basic Speech Formats

Table A-16 shows the Mandarin Enhanced Basic Speech formats.

Table A-16. Mandarin Enhanced Basic Speech Formats

Format	Description
DYYMD	Date – 4-digit year, month and day
DMD	Date – month followed by day
DTWYYMD	Date – 4-digit year per Taiwanese Republic calendar, month, day
D	Date – default DYYMD
THMAM	Time – same as T
T	Time – time period followed by time
N2	Number
N	Number
ND0	Decimal number – no decimal place
ND1	Decimal number – 1 decimal place
ND2	Decimal number – 2 decimal places
ND3	Decimal number – 3 decimal places
ND4	Decimal number – 4 decimal places
ND5	Decimal number – 5 decimal places
ND6	Decimal number – 6 decimal places
ND7	Decimal number – 7 decimal places
ND8	Decimal number – 8 decimal places
ND9	Decimal number – 9 decimal places
N$D0	Number – whole dollars
N$D1	Number – dollars with 1 decimal place
N$D2	Number – dollars and cents
N$D3	Number – dollars with 3 decimal places
N$D4	Number – dollars with 4 decimal places
N$D5	Number – dollars with 5 decimal places

Continued on next page

Table A-16. Mandarin Enhanced Basic Speech Formats — *Continued*

Format	Description
N$D6	Number – dollars with 6 decimal places
N$D7	Number – dollars with 7 decimal places
N$D8	Number – dollars with 8 decimal places
N$D9	Number – dollars with 9 decimal places
NYD0	Number – whole yuan
NYD1	Number – yuan with 1 decimal place
NYD2	Number – yuan, chiao and fun
NYD3	Number – yuan with 3 decimal places
NYD4	Number – yuan with 4 decimal places
NYD5	Number – yuan with 5 decimal places
NYD6	Number – yuan with 6 decimal places
NYD7	Number – yuan with 7 decimal places
NYD8	Number – yuan with 8 decimal places
NYD9	Number – yuan with 9 decimal places
NQD0	Number – whole quai
NQD1	Number – quai with 1 decimal place
NQD2	Number – quai with 2 decimal places
NQD3	Number – quai with 3 decimal places
NQD4	Number – quai with 4 decimal places
NQD5	Number – quai with 5 decimal places
NQD6	Number – quai with 6 decimal places
NQD7	Number – quai with 7 decimal places
NQD8	Number – quai with 8 decimal places
NQD9	Number – quai with 9 decimal places
NX	Speak the phrase with the specified phrase tag number. Also used to speak packed talkfile numbers and phrase numbers.
Cmmm	Characters – spoken individually
C	Characters – default Cmmm

Thai Enhanced Basic Speech Formats

Table A-17 shows the Thai Enhanced Basic Speech formats.

Table A-17. Thai Enhanced Basic Speech Formats

Format	Description
N	Number – speak an integer
ND0	Decimal number – no decimal place
ND1	Decimal number – 1 decimal place
ND2	Decimal number – 2 decimal places
ND3	Decimal number – 3 decimal places
ND4	Decimal number – 4 decimal places
ND5	Decimal number – 5 decimal places
ND6	Decimal number – 6 decimal places
ND7	Decimal number – 7decimal places
ND8	Decimal number – 8 decimal places
ND9	Decimal number – 9 decimal places
DBY	Date – Buddha year, spoken as whole number
DBYD	Date – Buddha year, spoken digit by digit
DBYS	Date – Buddha year, spoken in short form
DWY	Date – Western year, spoken as whole number
DWYD	Date – Western year, spoken digit by digit
DWYS	Date – Western year, spoken in short form
D	Date – default DBY
THMAM	Time – 12 hour, minute, no second
TH24M	Time – 24 hour, minute, no second
T	Time – default THMAM
N$	Default currency format N$T0
N$T0	Thai currency, whole baht
N$T2	Thai currency, baht and satang

Continued on next page

Table A-17. Thai Enhanced Basic Speech Formats

Format	Description
N$D0	US dollars – whole dollars
N$D2	US dollars – dollars and cents
Cm	English characters – spoken individually, medial inflection
Cf	English characters – spoken individually, falling inflection
Cr	English characters – spoken individually, rising inflection
C	Characters – default Cm

UK English Enhanced Basic Speech Formats

Table A-18 shows the UK English Enhanced Basic Speech formats.

Table A-18. UK English Enhanced Basic Speech Formats

Format	Description
DMDY	Date – 2-digit month, day, 2-digit year
DMDYY	Date – 2-digit month, day, 4-digit year
DMD	Date – 2-digit month, day
DMSPDYY	Date – spoken month, 2-digit day, 4-digit year
DMSPDY	Date – spoken month, 2-digit day, 2-digit year
DMSPD	Date – spoken month, 2-digit day
D	Date – default DMSPDY
THMAM	Time – hour, minute AM/PM
T	Time – default THMAM
Nrmf	Number – total inflection
Nmmf	Number – falling inflection
Nrmm	Number – rising inflection
Nmmm	Number – no inflection
N	Number – default Nmmm
ND1	Decimal number – 1 decimal place
ND2	Decimal number – 2 decimal places
ND3	Decimal number – 3 decimal places
ND4	Decimal number – 4 decimal places
ND5	Decimal number – 5 decimal places
ND6	Decimal number – 6 decimal places
ND7	Decimal number – 7 decimal places
ND8	Decimal number – 8 decimal places
ND9	Decimal number – 9 decimal places
N$	Number – whole pounds

Continued on next page

Table A-18. UK English Enhanced Basic Speech Formats — *Continued*

Format	Description
N$D0	Number – whole pounds
N$D2	Number – pounds and pence
N$D1	Number – pounds with 1 decimal place
N$D3	Number – pounds with 3 decimal places
N$D4	Number – pounds with 4 decimal places
N$D5	Number – pounds with 5 decimal places
N$D6	Number – pounds with 6 decimal places
N$D7	Number – pounds with 7 decimal places
N$D8	Number – pounds with 8 decimal places
N$D9	Number – pounds with 9 decimal places
NX	Speak the phrase with the specified phrase tag number. Also used to speak packed talkfile numbers and phrase numbers.
Crmf	Characters – spoken individually, total inflection
Crmm	Characters – spoken individually, rising inflection
Cmmf	Characters – spoken individually, falling inflection
Cmmm	Characters – spoken individually, no inflection

US English Enhanced Basic Speech Formats

Table A-19 shows the US English Enhanced Basic Speech formats.

Table A-19. US English Enhanced Basic Speech Formats

Format	Description
DMDY	Date – 2-digit month, day, 2-digit year
DMDYY	Date – 2-digit month, day, 4-digit year
DMD	Date – 2-digit month, day
DMSPDYY	Date – spoken month, 2-digit day, 4-digit year
DMSPDY	Date – spoken month, 2-digit day, 2-digit year
DMSPD	Date – spoken month, 2-digit day
D	Date – default DMSPDY
THMAM	Time – hour, minute AM/PM
T	Time – default THMAM
Nrmf	Number – total inflection
Nmmf	Number – falling inflection
Nrmm	Number – rising inflection
Nmmm	Number – no inflection
N	Number – default Nmmm
ND1	Decimal number – 1 decimal place
ND2	Decimal number – 2 decimal places
ND3	Decimal number – 3 decimal places
ND4	Decimal number – 4 decimal places
ND5	Decimal number – 5 decimal places
ND6	Decimal number – 6 decimal places
ND7	Decimal number – 7 decimal places
ND8	Decimal number – 8 decimal places
ND9	Decimal number – 9 decimal places
N$	Number – whole dollars

Continued on next page

Table A-19. US English Enhanced Basic Speech Formats — *Continued*

Format	Description
N$D0	Number – whole dollars
N$D2	Number – dollars and cents
N$D1	Number – dollars with 1 decimal place
N$D3	Number – dollars with 3 decimal places
N$D4	Number – dollars with 4 decimal places
N$D5	Number – dollars with 5 decimal places
N$D6	Number – dollars with 6 decimal places
N$D7	Number – dollars with 7 decimal places
N$D8	Number – dollars with 8 decimal places
N$D9	Number – dollars with 9 decimal places
NX	Speak the phrase with the specified phrase tag number. Also used to speak packed talkfile numbers and phrase numbers.
Crmf	Characters – spoken individually, total inflection
Crmm	Characters – spoken individually, rising inflection
Cmmf	Characters – spoken individually, falling inflection
Cmmm	Characters – spoken individually, no inflection
C	Characters – default Cmmm

Speech File Formats

B

Overview

This appendix describes the encoding formats for speech files.

Purpose

The purpose of this appendix is to ensure that you

- Are aware that the speech phrases must be in a supported format
- Understand and can apply a supported format

Speech File Formats

Before the speech phrases are digitized and then stored, the speech phrases must be in a supported format. This is the process of encoding speech. The INTUITY™ CONVERSANT® system supports the following encoding formats for speech phrases:

- PCM64 (pulse code modulation at 64 kbps in Mu-law encoding format)
- ADPCM32 (adaptive differential pulse code modulation at 32 kbps)
- ADPCM16 (adaptive differential pulse code modulation at 16 kbps — not commonly used in applications)
- CELP16 (code excited linear prediction at 16 kbps)
- SBC24 (sub-band coding at 24 kbps — not commonly used in applications)
- SBC16 (sub -and coding at 16 kbps — not commonly used in applications)

PCM Encoding Formats

In a PCM speech file, speech is sampled at 8000 times a second. Each sample is digitally coded into an 8-bit pattern (allowing 256 levels), resulting in 64,000 bits per second of speech. The header, which has the values *0xAA 0xFF 0x34 0x00*, is repeated at least 10 times for every second of speech (every 800 bytes). If the value *0xAA 0xFF* occurs anywhere within the speech portion of the file, it is changed to *0xAA 0xFE*. The speech data consists of contiguous 8-bit patterns representing sampled speech. One type of PCM format is available, Mu-law. In the Mu-law PCM format, PCM coding uses predefined quasi-logarithmic steps for speech levels, encoding more steps when the speech level is low.

ADPCM Encoding Formats

ADPCM speech encoding is based on the principle that it is possible to reduce the amount of information needed to transmit speech between the sender and the receiver by using appropriate mathematical algorithms. The following types of ADPCM speech file formats are available, both of which follow similar compression algorithms:

- ADPCM32
- ADPCM16

Given that natural speech follows specific patterns (as opposed to random noise), the encoder (or the sender) predicts the present speech level by using a predefined algorithm and considering the past speech history. This predicted value is compared to the actual present speech level and the difference is encoded into digital format. The difference between the actual and the predicted values can be made as small as possible by using suitably defined predictor algorithms.

Thus, it is possible to encode the difference between the predicted and the actual speech into digital samples of either 4 bits (16 levels) or 2 bits (4 levels). At the decoder (receiver) end, the process is reversed. The decoder uses an equivalent algorithm to predict the present speech level from the past history, and makes the correction based on the received information from the sender to get the actual speech level. The accuracy and performance of the speech encoding and decoding depends on the type of algorithm used as well as the number of bits used for digital encoding of the difference between predicted and actual speech.

- In the ADPCM32 speech file, the header, which has the values *0xAA 0xFF 0x32 0x00*, is repeated at least five times for every second of speech data.
- In the ADPCM16 speech file, the header, which has the values *0xAA 0xFF 0x30 0x00*, is repeated at least five times for every second of speech data.

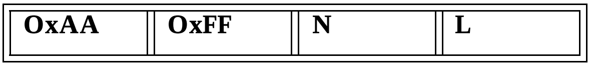
If the value *0xAA 0xFF* occurs anywhere within the speech portion of the file, it is changed to *0xAA 0xFE*. This makes a small but imperceptible change to the speech.

Processing of speech data for an application can be accomplished concurrently with application development. The only requirement is that the digitized speech must be loaded on the system before the application can be assembled.

Speech Files

Speech files are comprised of two parts:

- A header section of the format shown in Figure B-1 that is present at the beginning of the file and repeated at periodic intervals within the body of the file
- Encoded digital data representing speech



OxAA	**OxFF**	**N**	**L**

Figure B-1. Speech File Header Section

The following rules describe the header section:

- *0xAA* and *0xFF* are the 2 bytes of data with bit patterns *10011001* and *11111111.*

- *N* is a unique identifier code representing the speech encoding format, as shown in Table B-1.
- *L* (mandatory field) represents the length of a control field that follows the header bytes. The control field is optional. If it is absent, *L* is set to *0x00*. For the PCM and ADPCM type speech files, it is set to *0x00* (no control field is specified).

The following rules apply to the data section:

- Use an even number of bytes.
- Use no more than 500 bytes between headers.
- Use only CELP 16 for the pattern 0xAA 0xFF.

The length field L identifies the number of words (2 bytes each) that follow that include user defined information.

NOTE:
Coding and playback of CELP16 works properly only on the hardware that supports it, such as the SSP, IVC6, or NGTR circuit card. If coding or playback is performed on an unsupported circuit card, the result is silence or noise.

- When CELP16 is played on a circuit card that does not support CELP16, the result is silence or noise, depending on the phrase.
- When a circuit card that does not accept CELP16 is asked to code CELP16, it either runs a default mode or stays in the current mode and the coding circuit card fails to return a phrase.

Table B-1. Identifier Codes in Speech Encoding

N	Value	Channel Capacity (Playback/Coding) Per SSP Card
PCM (Mu-law)	0 x 34	120/120
ADPCM 32	0 x 32	120/120
ADPCM 16	0 x 30	120/120
SBC 24	0 x 21	100/100
SBC 16	0 x 20	100/100
CELP 16	0 x 40	120/60

Header bytes are inserted into the speech file so that the header appears at least five times for every second of speech. The headers are aligned on even byte boundaries.

Calculating the OS Index

C

Overview

The **gse** command used to start the Graphical Speech Editor (GSE) requires the identification of a channel number (chan) for each respective channel designation on the Tip/Ring circuit card (-l, -p, and -r).

gse [*-l* <*chan#*>] [*-p* <*playchan#*>] [*-r* <*recchan#*>]

The GSE does not recognize the same language as the UnixWare system. Therefore, the information that the UnixWare system supplies is not sufficient to identify the channels. However, a calculation using the operating system (OS) index supplied by the UnixWare system can derive the OS index in a form that can be used in the **gse** command.

NOTE:
Do *not* use this procedure to calculate channel numbers for the **wl_edit** command or any command associated with the FlexWord™ editor. Channels specified for the FlexWord editor are those listed in the output of the **disp ca tr** command, with no calculation required.

See *INTUITY™ CONVERSANT® System Version 7.0 Administration*, 585-313-501, for more information about the **gse** command. See Chapter 3, ‘‘Editing Speech’’ for information about using the GSE.

Purpose

This appendix provides a procedure to calculate the OS index used to identify channels on the Tip/Ring circuit card for the **gse** command.

Procedure to Calculate the OS Index

Use this procedure to determine the appropriate channel numbers for the l, p, and r channel designations on the Tip/Ring circuit card:

1. Identify the system channels that you want to use before accessing the system.

   For the example in this procedure, assume 7 and 8 to be the identified channels.

2. Log in to the system:

   a. Enter **root** at the system prompt.

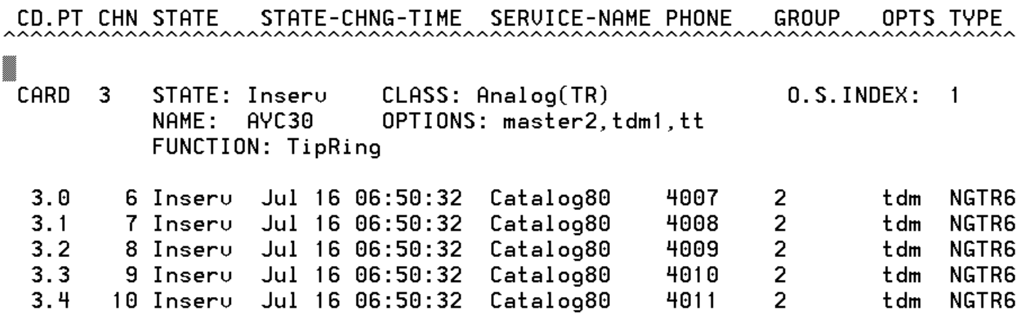
      The system displays the password prompt.

   b. Enter your valid password.

      The system displays the UnixWare system prompt (#).

3. Enter **disp ca tr**

   The system displays the current Tip/Ring channel settings, as shown in the example in Figure C-1.



```
 CD.PT CHN STATE    STATE-CHNG-TIME   SERVICE-NAME PHONE    GROUP    OPTS TYPE
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

 CARD   3    STATE: Inserv     CLASS: Analog(TR)             O.S.INDEX:  1
             NAME:  AYC30      OPTIONS: master2,tdm1,tt
             FUNCTION: TipRing

  3.0     6 Inserv  Jul 16 06:50:32  Catalog80     4007     2        tdm   NGTR6
  3.1     7 Inserv  Jul 16 06:50:32  Catalog80     4008     2        tdm   NGTR6
  3.2     8 Inserv  Jul 16 06:50:32  Catalog80     4009     2        tdm   NGTR6
  3.3     9 Inserv  Jul 16 06:50:32  Catalog80     4010     2        tdm   NGTR6
  3.4    10 Inserv  Jul 16 06:50:32  Catalog80     4011     2        tdm   NGTR6
```

Figure C-1. Example of Tip/Ring Channel Settings

4. Find the `O.S. Index:` field for the card on which the identified channel is located.

   In the example in Figure C-1, channel 7 and channel 8 (under `CHN`, second column) are both located on Card 3. Card 3 has 1 as its OS index number.

5. Find the channel offset.

   This number ranges from 0 through 5 and is the 1-digit suffix listed under `CD.PT` (first column). The offset for channel 7 is 1 because the `CD.PT` field contains the value 3.1; the offset for channel 8 is 2 because the `CD.PT` field contains the value 3.2.

6. Use the following formula to calculate the OS index for the channel.

   channel OS index = 6 x (card OS index) + (channel offset)

   In the examples in Figure C-1:

   - The OS index for channel 7 is 6 x 1 + 1 = 7
   - The OS index for channel 8 is 6 x 1 + 2 = 8

This OS index can now be used as an argument when invoking the GSE.

Text-to-Speech Formats

D

Overview

Text-to-Speech allows you to speak ASCII text with a synthesized voice from an INTUITY™ CONVERSANT® voice application. When speaking a text field or phrase, assigning a field format helps TTS speak the text properly. This appendix describes the formats supported in the Text-to-Speech (TTS) package for US English.

- For information on using TTS in Script Builder, applications see "Using Text-to-Speech" in Chapter 8, "Using Optional Features," in *INTUITY™ CONVERSANT® System Version 7.0 Application Development with Script Builder*, 585-313-260.
- For information on using the Variables Manager in Voice@Work, see the chapter on variables in *Voice@Work*, 585-313-207. For information on further customizing TTS, see Chapter E, ‘‘Advanced Text-to-Speech Features’’.

Purpose

This appendix is designed to be used as a reference when you create an application using TTS.

US English Text-to-Speech Formats

Table D-1 shows the format TTS uses to speak text for US English.

Table D-1. US English Text-to-Speech Format Set

Format	Description
A	Verbatim text (default format); speaks a character string as words, if possible (example: Lucent = “Lucent”)
AC	Alphabetics (example: Lucent = L U C E N T, with each letter pronounced individually”)
ADMD	Numeric month and numeric day with no year (example: “five one”)
ADMDY	Numeric month and numeric day with 2-digit year (example: “five one ninety-eight”)
ADMDYY	Numeric month and numeric day with 4-digit year (example: “five one nineteen ninety-eight”)
ADMSPD	Spoken month and ordinal day with no year (example: “May first”)
ADMSPDY	Spoken month and ordinal day with 2-digit year (example: “May first ninety-eight”)
ADMSPDYY	Spoken month and ordinal day with 4-digit year (example: “May first nineteen-ninety-eight”)
AN$D2	Dollars and cents, with the last two digits as decimal places (example: 12345 = “one hundred twenty-three dollars and forty-five cents”)
AN$D0	Whole dollars (example: 12345 = “twelve thousand three hundred forty-five dollars”
AN$D1	Dollars with one decimal place (example: 12345 = “one thousand two hundred thirty-four point five dollars”)
AN$D3	Dollars with three decimal places (example:12345= “twelve point three four five dollars”)
AN$D4 AN$D5 AN$D6 AN$D7 AN$D8 AN$D9	Dollars with the respective number of decimal places (see AN$D3). **NOTE:** If the number has fewer digits than the number of decimal places specified by the format, TTS inserts zeros before the number.
AN	Whole number (example: 12345= “twelve thousand three hundred forty-five”)

Continued on next page

Table D-1. US English Text-to-Speech Format Set — *Continued*

Format	Description
AND1	Number with one decimal place (example: 12345 = “one thousand two hundred thirty-four point five”)
AND2 AND3 AND4 AND5 AND6 AND7 AND8 AND9	Number with respective number of decimal places (see AND1) **NOTE:** If the number has fewer digits than the number of decimal places specified by the format, TTS inserts zeros before the number.
ATHMAM	Time— hours, minutes, AM/PM (example: “twelve thirty-four PM” **NOTE:** Input to this field type is assumed to be in 24-hour time format. TTS designates AM and PM.

Advanced Text-to-Speech Features

E

Overview

For some Text-to-Speech (TTS) applications, you may need to customize the way TTS speaks to improve understandability or comprehension. Methods of customization include using escape sequences to:

- Insert silent pauses or delays at the beginning, middle, or end of phrases
- Change the gender (male or female) of the synthesized voice
- Change the speaking rate of the synthesized voice
- Mark text as a member of a text class and specifying a detector type that determines how certain words are spoken

NOTE:
TTS is available for US English only.

Purpose

The purpose of this chapter is to enable you to use escape sequences to customize the way TTS speaks.

Escape Sequences

Customization of TTS involves using strings of characters called escape sequences that reset the TTS default parameters. These methods of customizing TTS can be used in:

- Script Builder applications in Prompt and Collect action steps and Announce action steps
- Voice@Work applications in Prompt and Collect, Announce, Menu, Automenu, and Voice Capture nodes.

The exact method of changing the defaults depends on the parameter and the application development tool.

Text and Variables

In both Script Builder and Voice@Work applications, you can enter the escape sequences in text or variable field definitions.

For information on field definitions in Script Builder, see Chapter 3, “Data Management,” in *INTUITY™ CONVERSANT® System Version 7.0 Application Development with Script Builder*, 585-313-260. For information on using the Variables Manager in Voice@Work, see the chapter on variables in *Voice@Work*, 585-313-207.

Controls

In Voice@Work applications, you can achieve some types of customization by inserting TTS controls with the Insert Control dialog box in the Prompt Editor. When code is generated, the controls are translated to the escape sequences.

For information on inserting controls in Voice@Work, see the chapter on prompts in *Voice@Work*, 585-313-207.

Inserting Silent Delays and Pauses

Inclusion of silent delays or pauses in an application helps make the speech more natural sounding and may help increase listener comprehension. Table E-1 lists the escape sequences used to insert TTS silences into synthesized speech.

Table E-1. TTS Escape Sequences for Silence

Sequence	Description
\!si*n* (initial)	■ Specifies *n* centiseconds of silence at the beginning of the current text phrase, where 1< *n* < 100 ■ Must be placed at the beginning of a text phrase and therefore cannot be used in the same phrase as a final silence (\!sf*n*) **NOTE:** Since this sequence must occur at the beginning of a a text phrase, it must either be placed first in the Script Builder action step or Voice@Work node or the previous phrase must include end punctuation indicating the end of a phrase— either a period (.), question mark (?), or exclamation mark (!).
\!sf*n* (final)	■ Specifies *n* centiseconds of silence at the end of the current phrase, where 1< *n* < 100 ■ Must be placed at the beginning of a text phrase and therefore cannot be used in the same phrase as an initial silence (\!si*n*) (This sequence is entered at the beginning of a phrase but affects the end of the phrase.) **NOTE:** Since this sequence must occur at the beginning of a a text phrase, it must either be placed first in the Script Builder action step or Voice@Work node or the previous phrase must include end punctuation indicating the end of a phrase— either a period (.), question mark (?), or exclamation mark (!).
\!(*[*n*])	Specifies *n* centiseconds of silence within the current phrase, where 1< *n* < 60. Can be placed anywhere within a text phrase and can be used in the same phrase with either an initial silence (\!si*n*) or a final silence (\!sf*n*).

NOTE:
When typed in text or a variable, these sequences must be followed by one blank space.

WARNING:
Using values outside the limits could cause problems in your application.

Examples for Silence— Using Text

The following examples illustrate use of escape sequences in text to add silent pauses and delays in an application.

- In the following example from a Script Builder application, a silence of 0.5 seconds occurs before "Hello" is spoken. The delay helps secure the listeners' attention.

```
Announce
  Speak with Interrupt
Text: “\!si50 Hello!”
Text: “Welcome to your Home Shopping Line!”
```

  The same text would be displayed as follows in the Voice@Work Prompt Editor for an Announce node.

```
\!si50 Hello!
Welcome to your Home Shopping Line!
```

- In the following example from a Script Builder application, a silence of 0.7 seconds is added after the phrase "Welcome to the Home Shopping Line!" is spoken. There is no silence between the word "Hello!" and the text that follows. The silence helps emphasize the end of the announcement.

```
Announce
  Speak with Interrupt
Text: “\!si50 Hello!”
Text: “\!sf70 Welcome to your Home Shopping Line!”
```

  The same text would be displayed as follows in the Voice@Work Prompt Editor for an Announce node.

```
\!si50 Hello!
\!sf70 Welcome to your Home Shopping Line!
```

  NOTE:
  The end punctuation (exclamation point) after the first phrase ("Hello!") is necessary for TTS to speak the 0.7-second silence at the end of the second phrase ("Welcome to your Home Shopping Line!"). Without end punctuation in the first phrase, this escape sequence would be ignored.

- In the following example from a Script Builder application, a text phrase contains two silences— a 0.5-second silence at the beginning of the phrase and a 0.7-second silence between the words "Hello" and "and." The placement of these silences helps make the synthesized speech more natural sounding.

```
Announce
 Speak with Interrupt
Text: “\!si50 Hello \!(*[70]) and welcome to your”
Text: “Home Shopping Line!”
```

The same text would be displayed as follows in the Voice@Work Prompt Editor for an Announce node.

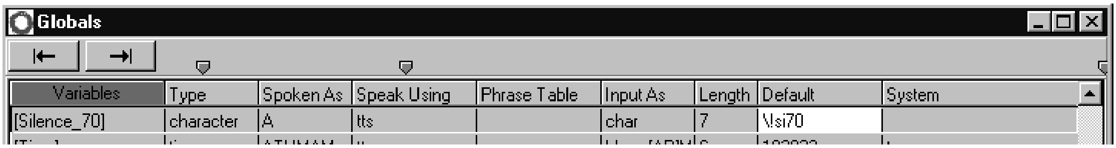
```
\!si50 Hello \!(*[70]) and welcome to your
Home Shopping Line!
```

Examples for Silence— Using a Variable

- The following example from a Script Builder application shows a silence delay defined as a variable called Silence_70 and used in an Announce action step. A silence of 0.7 seconds occurs between the first and second phrase.

```
Answer Phone
Set Field Value
  Field: Silence_70 = “\!si70 “
Announce
  Speak With Interrupt
    Text: “Welcome to your Home Shopping Line!”
    Field: Silence_70 As A
    Text: “To place an order, please have your”
    Text: ”account number ready.”
```

- Figure E-1 shows a detail of the Voice@Work Variables Manager where a variable named [Silence_70] has been created for a silent pause of 0.7 seconds. The escape sequence \!si70 is set in the `Default` field.



Variables	Type	Spoken As	Speak Using	Phrase Table	Input As	Length	Default	System
[Silence_70]	character	A	tts		char	7	\!si70	

Figure E-1. Variable Definition for a Silent Delay in Voice@Work

Once the variable has been created, you can insert it into the application by use of the Insert Variable dialog box in the Prompt Editor. In the following example, two 0.7-second silences occur, one before the first phrase is spoken and one before the second phrase is spoken. As the example illustrates, the Prompt Editor displays variables in square brackets — [ ].

```
[Silence_70]
Welcome to your Home Shopping Line!
[Silence_70]
To place an order, please have your account number
ready!
```

Changing the Gender of the Speaking Voice

TTS permits speech in both male and female voices. The default is male voice. Female voice provides the same capabilities and features as male voice. Voices can be switched by use of the escape sequences shown in Table E-2.

Table E-2. TTS Escape Sequences for Speech Gender Selection

Sequence	Description
\!uSm	Male voice (default)
\!uSf	Female voice

NOTE:
When typed in text or a variable, these sequences must be followed by one blank space.

The voice selection stays in effect until an event occurs that removes speech inhibition. At that point, speech reverts to the default (male) voice. Therefore, to create an entire application in female voice, you must reset the voice repeatedly.

The following events remove speech inhibition:

- In a Script Builder application, one of the following action steps occurs:
  - Get Host Screen
  - Send Host Screen
  - Read Table
  - Modify Table
  - Transfer Call
- In a Voice@Work application any Database node (including Query Table, Update Table, Add Record, and Delete Record) or Host node (including Host Connection, Host Send, and Host Send-Receive) occurs
- A message is played that does not allow barge-in (interrupt) in a Prompt and Collect or an Announce action (Script Builder) or in a Prompt and Collect or an Announce node (Voice@Work)
- An external function is called that contains a wait-causing instruction

Example for Speaking Voice Gender— Using Text

- The following example from a Script Builder application shows use of the escape sequence to set the TTS female speaking voice.

```
Prompt & Collect
  Prompt
    Speak with Interrupt
      Text: “\!uSf Using touch tones, please enter”
      Text: “your 3-digit account number.”
```

- The same text would be displayed as follows in the Voice@Work Prompt Editor for a Prompt and Collect node:

```
\!uSf Using touch tones, please enter
your 3-digit account number.
```

Examples for Speaking Voice Gender— Using a Variable

- In the Script Builder example above, you can define a variable as follows instead of using text to change the speaking voice gender:

```
Set Field Value
  Field: Female_Voice = “\!uSf “
Prompt & Collect
  Prompt
    Speak with Interrupt
      Field: Female_Voice as A
      Text: “Using touch tones, please enter”
      Text: “your 3-digit account number.”
```

- For a Voice@Work application, you can define the same variable as in the example above by entering \!uSf in the `Default` field in the Variables Manager. (See Figure E-1 above for an illustration of the Voice@Work Variables Manager.) Once you insert the variable into a Prompt and Collect node by using the Insert Variable dialog box in the Prompt Editor, the Prompt Editor displays the variable in square brackets— [ ].

```
[Female_Voice]
Using touch tones, please enter
your 3-digit account number.
```

Changing the Rate of Speech

Slowing the rate of speech for important words can provide emphasis and increase listener comprehension in an application. Table E-3 shows escape sequences used to control the rate of speech for TTS.

Table E-3. TTS Escape Sequences for Speaking Rate

Sequence	Description
\!rfastest \!rfast \!rnormal \!rslow \!rslowest	The default rate is “normal” (\!rnormal). Each rate is effective until the end of a sentence, indicated by punctuation such as a period, question mark, or exclamation mark. In text, to change the rate before the end of a sentence, insert a different escape sequence. In Voice@Work, equivalents of \!rfastest, \!rnormal, and \!rslowest can be inserted as controls (\!rfast and \!rslow are not supported as controls).

NOTE:
When typed in text or a variable, these sequences must be followed by one blank space.

In Voice@Work applications, some of the sequences can also be inserted as controls. When code is generated, the controls are translated to the escape sequences shown in Table E-3.

NOTE:
The escape sequences in Table E-3 are used for TTS on the speech and signal processor (SSP) circuit card only and cannot be used for TTS on the older signal processor (SP) circuit card. (The SP circuit card is not supported for INTUITY CONVERSANT system releases beginning with Version 7.0). Equivalent sequences supported for the SP circuit card were: \!r0 (fastest), \!r1, \!r2, \!r3, and \!r4 (slowest). See ‘‘Voice@Work Controls for Rate of Speech’’ below for more details regarding Voice@Work.

Example for Rate of Speech— Using Text

- In the following example from a Script Builder application, the slower rate is in effect only while the order number is spoken. The rest of the announcement is spoken with the default (\!rnormal) rate.

```
Announce
  Speak with Interrupt
    Text: “Your order number is”
    Text: “\!rslowest 104235\!rnormal.”
    Text: “Please use this number”
    Text: “when checking your order.”
```

- The same text would be displayed as follows in the Voice@Work Prompt Editor for an Announce node:

```
Your order number is
\!rslowest 104235\!rnormal.
Please use this number
when checking your order.
```

Examples for Rate of Speech— Using a Variable

- In the Script Builder example above, you can set a variable such as the following instead of using text to change the rate of speech.

```
Set Field Value
  Field: Rate_Slowest = “\!rslowest “
```

- In a Voice@Work application, you can define the same variable as in the example above by entering \!rslowest in the `Default` field in the Variables Manager. (See Figure E-1 above for an illustration of the Voice@Work Variables Manager.) Once you insert the variable into the application by using the Insert Variable dialog box in the Prompt Editor, the Prompt Editor displays the variable as follows:

```
[Rate_Slowest]
```

Voice@Work Controls for Rate of Speech

In Voice@Work applications, you can use the Insert Control dialog box in the Prompt Editor to insert controls that change the rate of speech. When code is generated, these controls are translated to the escape sequences shown in Table E-3. The following controls are available:

- {tts fast} — equivalent to \!rfastest
- {tts normal} — equivalent to \!rnormal
- {tts slow} — equivalent to \!rslowest

NOTE:
When you use these controls in a Version 7.0 application, ensure that the `Use SP style TTS speech rate codes` checkbox is unselected on the Voice@Work Code Generation Options window. This allows the appropriate escape sequences for the SSP circuit card to be generated. The `Use SP style speech rate codes` checkbox should be selected only for applications on systems using the older SP circuit card (not supported in INTUITY CONVERSANT system Version 7.0). See the chapter on the code generation tool in *Voice@Work*, 585-313-207, for information.

Setting Text Classes and Detectors

In an application, you can designate the format of a field (Script Builder) or a variable (Voice@Work) to be spoken by TTS as:

- Character
- Time
- Date
- Number
- Dollar amount
- Currency real (Voice@Work only)

(See Chapter D, ''Text-to-Speech Formats'' for a list of the formats supported for TTS).

This information is often sufficient for TTS to speak text correctly. However, when the context for the text is ambiguous or requires a specific style of speaking, more information must be provided. To illustrate the problem, consider the example of TTS trying to speak separate fields of information containing a person's name, street address, and city, state, and zip code obtained from a database in the form below. TTS may not know whether to speak "22587" as part of an address or as a number because it is a separate field from the rest, presented with little context, and it could be either.

customer name field = `Jane Doe`
city/state field = `Columbus OH  43213`
zip code field = `43213`
address field =`22587 S. Hamilton St.`

To solve this sort of problem, TTS allows you to explicitly assign text to any of the following four classes:

- Telephone number
- Proper name
- Address
- Fraction

For TTS to speak text assigned to these classes, you must also assign one of the following class detector modes:

- Off
- Conservative mode (default)
- Risky mode

NOTE:
Initially all detectors are set to conservative mode.

The detector mode determines how TTS speaks ambiguous text.

- Risky mode sets a priority for the designated class to resolve ambiguity about how the information is to be classified. The setting of risky mode will be the highest priority at the time for the field to be spoken. TTS will then know, for example, that “22587” in the problem described above is part of an address.
- Conservative mode requires more contextual information to resolve ambiguous text.

Table E-4 shows the escape sequences used for class detector modes.

- In both Script Builder and Voice@Work applications, these sequences can either be entered in text or defined as variables. The class detector mode remains in the new state until it is changed or until the application ends.

  NOTE:
  One technique of resetting the mode is to use an Announce action step or node to speak only the escape sequence.

- In Voice@Work applications, a subset of the class detector modes can be inserted as controls. When code is generated, the controls are translated to the escape sequences shown in Table E-4. The control sets the class detector to risky mode and affects only the next line in the Prompt Editor. The class detector is then automatically reset to conservative mode.

Observe the following guidelines when using class detectors:

- As a general rule, do not use risky mode unless in tests of your application TTS fails to pronounce words or phrases correctly or the text is known to be a member of the specified text class (for example, telephone numbers) or contains ambiguous abbreviations. Otherwise, always try conservative mode first (that is, do not explicitly set a class detector).
- More than one class detector can be activated at the same time, though results may be unpredictable.
- After using risky mode, reset detectors to conservative mode if special detection is not needed.
- Do not set a detector mode to off unless conservative mode is causing an undesirable effect.

Since TTS speaks text better in context, you may want to concatenate fields to be spoken. For Voice@Work, see information on the **concat** expression in sections describing the Set and Test node in the chapter on node descriptions in *Voice@Work*, 585-313-297. For Script Builder, see information on the **concat** function in Chapter 11, “Using Advanced Features,” of *INTUITY™ CONVERSANT® System Version 7.0 Application Development with Script Builder*, 585-313-260.

Table E-4. TTS Escape Sequences for Class Detectors

Sequence	Description
Telephone number `\!npo` (off) `\!npr` (risky mode) `\!npc` (conservative mode)	In conservative mode TTS speaks telephone numbers as individual digits, except for 800 in area code position, which is spoken "eight hundred". For example, 555 1365 is spoken "five five five (pause) one three six five," not "five million (pause) five hundred fifty-one thousand (pause) three hundred sixty-five." Risky mode also ensures that "vanity" telephone numbers, such as 1 800 SHOPPER, are spelled out.
Proper name `\!nno` (off) `\!nnr` (risky mode) `\!nnc` (conservative mode)	Risky mode is needed only when the data may contain name-related abbreviations or titles. Otherwise, use conservative mode. In risky mode, titles such as "Mr." and ambiguous abbreviations such as "Dr." are spoken correctly. Abbreviations such as "Sr." (as in "John Doe Sr.") and "Jr." expand correctly in either risky or conservative mode because there is only one possible interpretation of the abbreviations. **NOTE:** TTS will speak a proper name like "John Paul III" with three I's. For TTS to pronounce this field successfully, you can enter either "John Paul third" or "John Paul 3rd". This can be done in either conservative or risky mode because the "3rd" and "third" cannot be misinterpreted.
Address `\!nao` (off) `\!nar` (risky mode) `\!nac` (conservative mode	Use risky mode to detect partial addresses, and to speak ambiguous address abbreviations correctly, such as "drive" for "Dr." (not "doctor") and "street" for "St." not "saint"). Standard post office abbreviations apply.
Fraction `\!nfo` (off) `\!nfr` (risky mode) `\!nfc` (conservative mode	Use risky mode so that, for example, "1/4" is spoken as "one fourth" and not "one slash four." (To have TTS speak "1/4" as a date, use conservative date mode.)

NOTE:
When typed in text or a variable, these sequences must be followed by one blank space. Sequences to turn detectors off (\!npo, \!nno, \!nao, \!nfo), are not available as controls in Voice@Work (see "Voice@Work Controls for Class Detectors" below.

Examples for Class Detectors — Using Text

The examples below illustrate use of class detectors set in text for the telephone number, address, and proper name classes in Script Builder applications. Examples for Voice@Work applications displayed in the Prompt Editor would be similar except that no quotation marks are required around the text.

Telephone Number

The following example sets the telephone number detector to risky mode, speaks a “vanity” telephone number, and then turns the detector off because no more telephone numbers are expected.

```
Announce
 Speak with Interrupt
   Text: “Please call \!npr 1-800 SHOPPER \!npo “
   Text: “for more information.”
```

Address

The example below assumes that a Script Builder application has obtained from a database the following data for the specified fields:

Customer_address:	12345 Mulberry St.
Customer_state	CT

The Announce action sets the address detector to risky mode, speaks an address from the Customer_address and Customer_state database fields, and then sets the detector back to conservative mode. Risky mode is needed to rank possibilities for the address so that ““St.” is spoken as “street” (and not “saint” and CT” is spoken as “Connecticut” (and not “court” or “count”). The detector is set back to conservative mode, without priority, in preparation to process the next field.

```
Announce
  Speak with Interrupt
    Text: “\!nar“
    Text: “Your address is“
  Field: Customer_address as A.
  Field: Customer_state as A
    Text: “\!nac“
```

Proper Name

The following example sets the proper name detector to risky mode, speaks a customer name from a Customer_name database field, and then sets the detector back to conservative mode. Risky mode is needed because the data may contain name-related abbreviations or titles.

```
Prompt & Collect
  Prompt
    Speak with Interrupt
      Text: “\!nnr”
      Text: “Your name is“
    Field: Customer_Name as A
      Text: “\!nnc“
      Text: “Press 1 for yes, 2 for no.”
```

Examples for Class Detectors — Using a Variable

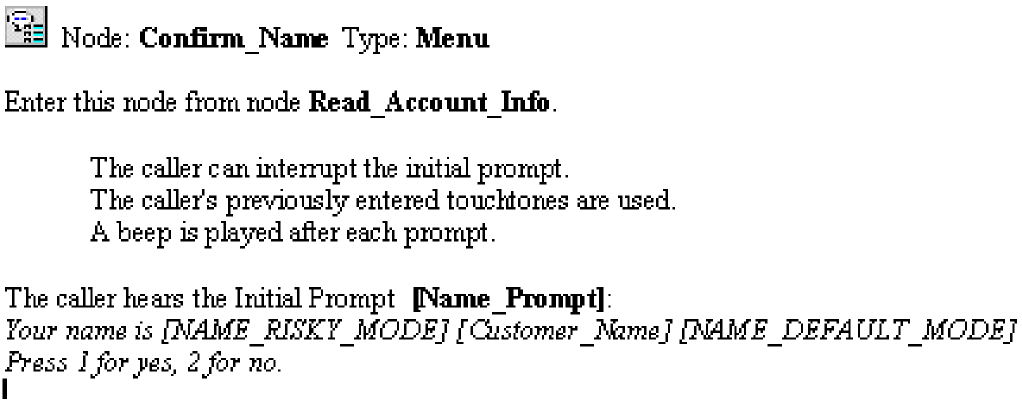
- In the Script Builder example above for the proper name detector, you can set variables such as the following instead of using text to change the class detector:

```
Set Field Value
  Field: Name_Risky_Mode = “\!nnr “
  Field: Name_Default_Mode = “\!nnc “
```

- In an equivalent Voice@Work application for the proper name detector, you can achieve the same result by creating variables with the following definitions in the Variables Manager (see Figure E-1 above for an illustration of the Voice@Work Variables Manager):

Name	Type	Spoken As	Speak Using	Phrase Table	Input As	Length	Default
[NAME_DEFAULT_MODE]	character	A	tts		char	10	\!nnc
[NAME_RISKY_MODE]	character	A	tts		char	10	\!nnr

You can then use the Insert Variable dialog box in the Prompt Editor, to insert the variables into the application. If you print the node information, the output for this portion of the node resembles Figure E-2.



Node: **Confirm_Name** Type: **Menu**

Enter this node from node **Read_Account_Info**.

The caller can interrupt the initial prompt.
The caller's previously entered touchtones are used.
A beep is played after each prompt.

The caller hears the Initial Prompt **[Name_Prompt]**:
Your name is [NAME_RISKY_MODE] [Customer_Name] [NAME_DEFAULT_MODE]
Press 1 for yes, 2 for no.

Figure E-2. Example of Address Detector Variables in Voice@Work Node Information

Voice@Work Controls for Class Detectors

Voice@Work allows you to use the Insert Control dialog box in the Prompt Editor for the following class detectors:

- {tts address}
- {tts fraction}
- {tts proper name}
- {tts phone number}

Inserting the control sets the class detector to risky mode and affects only the next line in the Prompt Editor. The class detector is then automatically reset to the default (conservative mode). Therefore, if risky mode is needed for several lines, you can either insert the control repeatedly or use an escape sequence to set the class detector in a variable or text statement.

NOTE:
The class detectors cannot be turned off (o mode) with controls. To turn off the class detectors, you must use an escape sequence in text or a variable.

Enabling Barge-In for Risky Mode

For Script Builder and Voice@Work applications, the barge-in feature works for TTS in risky mode only if a field is used to turn it on, not a text statement (or a control in Voice@Work).

For example, in a Script Builder application the address detector can be set to risky mode in text as follows when the ability of the caller to barge in is *not* needed.

```
Announce
  Speak With Interrupt
    Text: “\!nar”
```

If barge-in is needed, a variable such as the following can be defined for the address detector in risky mode:

```
Set Field Value
  Field: Adsress_Risky_Mode = “\!nar”
```

When needed, the variable can be included as:

```
Field: Address_Risky_Mode As A
```

Recognition Post-Processing

F

Overview

This appendix describes:

- Data interface processes (DIPs) associated with dial pulse recognition (DPR), WholeWord speech recognition, and FlexWord™ speech recognition
- Processing features that increase the system accuracy of recognition, including:
  - — Return Top 4 (RT4)
  - — Prefix List Check (LSC)
  - — Luhn Check (LNC)
  - — Expected Value
  - — Close Match

NOTE:
The post-processing features are supported for Script Builder applications only.

Purpose

This appendix is designed to help you understand how processing features can improve recognition accuracy and how to use the features in a Script Builder application.

Further Processing

A data interface process (DIP) called **recog_dip** is associated with DPR, WholeWord speech recognition, and FlexWord speech recognition. The **recog_dip** is an optional, post-processing mechanism for enhancing the performance of some recognition tasks. It furnishes additional recognition processing beyond what is already provided by the models and grammars of the recognizer.

This DIP is located in **/vs/bin/vrs/recog_dip**. Whether the **recog_dip** is used during a particular recognition task is determined in the **/vs/data/sr_file** file.

When the **recog_dip** is in use, the recognizer sends its top four choices (and a score for each) to the DIP, which in turn makes a call to the customer-editable function **find_best.c** located in the source code file **/att/asr/find_best.c**. The **find_best.c** routine is responsible for making a final selection from the list of recognition choices and passing it back to the script as $CI_VALUE or other specified customer input variable.

Many general post-processing algorithms have been provided in the generic version. Look to see if the existing functionality exists before adding custom features. If you do edit this code, be sure to back it up prior to updating generics or reinstalling the recognizer package.

The default version of the **find_best.c** routine (the one that comes with the recognition package) simply selects the result with the highest score that falls within the constraint set for the minimum and maximum number of digits. Therefore, if the recognizer is applying a fixed-length grammar, the default version of the **find_best.c** routine adds no value to the recognition procedure. If the recognizer is applying a variable-length grammar, the default routine **find_best.c** is useful for range checking.

NOTE:
Where possible, application constraints should be placed in a custom grammar to gain the most improvement in accuracy.

To access the post-processing features described in this appendix, it is necessary to create a custom recognition type. Custom recognition types should be defined in directory **/att/asr/sr_files/**.

Figure F-1 is an example of defining the recognition type file format.

```
class type   min  max  recognizer      interrupt  grammar  dip        subr buf_sz
--------------------------------------------------------------------------------------
SR RT4_an    8    8   IRD_WHOLE_WORD    Y         BP_yn    im."recog_dip"
```

Figure F-1. Example of Defining the Recognition Type File Format

After supplying the custom recognition type, run the command **/att/asr/tools/bin/merge** to update the INTUITY™ CONVERSANT® system.

The real power of the **recog_dip** becomes apparent when the **find_best.c** routine is customized to place added constraints on the recognizer results. This helps limit the valid recognition results and thereby improves recognition accuracy. For example, an application is attempting to recognize a 10-digit telephone number and it knows that when the area code is 614, the final 4 digits of the telephone number will always fall between 5,500 and 9,000, and when the area code is 216 or 513 the final 4 digits will always be between 1,000 and 4,400. Customizing the **find_best.c** routine to account for these data dependencies, should provide an increase in recognition performance. The point to note here is that a certain amount of dependency between digits can be accomplished with a custom grammar, which gets implemented at the recognizer level. However, more complex patterns or dependencies within the expected user input are best handled with the **find_best.c** routine and the **recog_dip**.

After making and compiling changes made to the **find_best.c** routine, you will need to remake the **recog_dip**. **Makefile** for this is included in the **/att/asr** directory. Test any changes you make thoroughly, as mistakes could severely impact your application's recognition accuracy.

Return Top 4 (RT4)

RT4 is a processing feature that returns up to four choices, separated by the pound (#) character, for what the system determined the caller said, sorted by recognition score. For example, the candidates can be validated in order against a database within the script. By using this processing feature, more information is used to ensure greater correctness.

In the following examples of returns from RT4 processing, N refers to a digit [0-9].

NNN#NNN#NNN#NNN#	Four results were returned, each of length three.
?#NNN#	The best result was rejected ("?"), with only one other valid result.
####	All results were rejected (empty string).

Script Builder Example

The following program was created in Script Builder to demonstrate the use of the RT4 recognition DIP:

```
1. Answer Phone
     main:
2. Prompt & Collect
      Prompt
        Speak With Interrupt
          Text: "say a 5 digit number"
        Input
          Min Number Of Digits: 05
          Max Number Of Digits: 05
          RECOG: SR   RT4_BP5
        Checklist
          Case: "Input Ok"
            Continue
          Case: "Initial Timeout"
            Reprompt
          Case: "Too Few Digits"
            Reprompt
          Case: "No More Tries"
            Quit
    End Prompt & Collect
    loop:
3.  External Function
      Function Name: length
      e Arguments: $CI_VALUE
        Return Field:  Len
4.  Evaluate
    If Len   > 5
5.  External Function
      Function Name: parse
      Use Arguments: cand  $CI_VALUE  "#"
6.   Announce
       Speak With Interrupt
         Text: "You said"
         Field: cand As AC
7.  Goto loop
      End Evaluate
8.  Goto main
9.  Announce
      Speak With Interrupt
        Text: "good bye"
10. Disconnect
```

This example illustrates the following key points:

- In the Prompt and Collect action, the recognition type is set to **RT4_DIG** (to enable the RT4 DIP, the first three characters must be “RT4”). In this example **RT4_DIG** is a custom recognition type from a custom package.
- The **loop** label and **Goto loop** action provides the looping construct that helps unpack the values from the returned variable `temp_val`.
- The external function **length** allows the application to decide how many values are packed in `temp_val` using the evaluate action.
- The external function **parse** provides the unpacking mechanism of `temp_val`. It also strips the trailing “#” symbol from the value.

Prefix List Check (LSC)

LSC compares recognizer results with values in the file **/att/asr/lst.prefix**. The entries in the list need not be the same length. The post-processing marks as valid only those results that match any list entry as a prefix. This feature currently supports only one list with a maximum of 256 entries per list.

Luhn Check (LNC)

The Luhn check is a processing feature that tests the validity of an account number. For example, most credit card numbers are encoded with a “check digit”, usually appended to a number, that validates the authenticity of the number. A simple algorithm is applied to the other digits of the number which yields the check digit. By running the algorithm, and comparing the check digit obtained from the algorithm with the check digit encoded with the credit card number, you can verify that you have correctly read all of the digits and that they are a valid combination.

Validation Formula

Check **/att/asr/mod10.c** for an explanation of the check sum formula. The algorithm can be found in the comments section of the **mod10.c** file that accompanies the WholeWord recognition package.

Expected Value (EXV)

An EXV for a recognition can be passed from a script to the recognition post-processing function and can be used to validate or to reject a particular speech recognition candidate.

In the script, the expected value should be placed in the system variable $FINDBEST prior to calling the recognizer in the Prompt and Collect action. The $FINDBEST value is accessible in the customer-editable function `find_best`, which is within the recognizer post-processing routine **recog_dip**.

For example, assume that at a point in the call flow it has been determined that the caller is located in Columbus, Ohio, and that the caller is asked to speak his or her US English 5-digit zip code. The EXV can be set to 432 because the first three digits for all zip codes in this city are 432. In the post-processing for the case of the recognition type for zip code, the first 3 digits of the recognition result can be compared to 432. The first candidate that matches the pattern can be returned as the best choice.

Close Match (CLM)

CLM is a function available to the customer-editable post processing that compares two strings. The function does an **align** in **/att/asr/aligner.c** and if the two strings differ only by one substitution error or one insertion/deletion pair error, the function returns 1. An example of the insertion/deletion pair would be if the caller speaks the digits, “43230” and the system recognizes, “64320.” In this case, the system has incorrectly inserted the “6” and deleted the “3” before the zero. As already stated, this function returns 1.

Combining the use of CLM and EXV can improve application accuracy. For example, assume that at a point in the call flow the caller has been identified and that a caller profile has been read into the script. If the script requests verification of the house number of the caller's address, the EXV can be set to the house number prior to the Prompt & Collect step. In the post-processing for the recognition type for the house number, the recognition result string can be compared to the EXV string to determine if they closely match. This assumes that if the strings do not exactly match, it is most likely a recognition error. If the strings match within 1 error, then the result is accepted as correct.

Further Processing Prefixes and Combinations

Each of the processing functions above and all combinations are invoked by using an appropriate prefix to specify a custom recognition type. Table F-1 shows the prefix names for the further processing functions.

Table F-1. Prefix Names For Further Processing Functions

Function	Prefix
Returns (up to) the top four results to the script	RT4
Performs a Luhn (**mod10**) checksum calculation	LNC
Performs a prefix list check	LSC
Performs an expected value check	EXV
Performs a close match check	CLM

For combinations, prefixes are strung together at the beginning of the recognition type. Table F-2 illustrates the functions of some combination prefixes.

Table F-2. Examples of Combination Prefixes and Functions

Function	Prefix
Returns (up to) the top four results that satisfy the Luhn check	LNCRT4
Returns (up to) the top four results that satisfy the close match check	CLMRT4
Returns (up to) the top four results that satisfy the Luhn check *and* list check	LNCLSCRT4

Glossary

Numerics

23B+D
23 bearer (communication) and 1 data (signaling) channel on a T1 PRI circuit card.

30B+D
30 bearer (communication) and 1 data (signaling) channel (plus framing channel 0) on an E1 PRI circuit card.

3270 interface
A link between one or more INTUITY™ CONVERSANT® machines and a host mainframe. In INTUITY CONVERSANT system documentation, the 3270 interface specifically means the link between one or more system machines and an IBM host mainframe.

47B+D
47 bearer (communication) and 1 data (signaling) channel on two T1 PRI circuit cards.

4ESS®
A large Lucent central office switch used to route calls through the telephone network.

A

AC
alternating current

ACD
automatic call distributor

AD
application dispatch

AD-API
application dispatch application programming interface

adaptive differential pulse code modulation
A means of encoding analog voice signals into digital signals by adaptively predicting future encoded voice signals. This adaptive modulation method reduces the number of bits required to encode voice. See also “pulse code modulation.”

adjunct products

Products (for example, the Adjunct/Switch Application Interface) that the INTUITY system administers via cut-through access to the inherent management capabilities of the product itself; this is in opposition to the ability of the INTUITY CONVERSANT system to administer the switch directly.

Adjunct/Switch Application Interface

An optional feature package that provides an Integrated Services Digital Network-based interface between Lucent Technologies PBXs and adjunct processors.

ADPCM

adaptive differential pulse code modulation

ADU

asynchronous data unit

advanced speech recognition

A speech recognition ability that allows the system to understand WholeWord and FlexWord™ inputs from callers.

affiliate

A business organization that Lucent controls or with which Lucent is in partnership.

AGL

application generation language

alarm relay unit

A unit used in central office telecommunication arrangements that transmits warning indicators from telephone communications equipment (such as an INTUITY CONVERSANT system) to audio.

ALERT

System alerter process

alerter

A system process that responds to patterns of events logged by the “logdaemon” process.

American Standard Code for Information Interchange

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

analog

An analog signal, such as voice or music, that varies in a continuous manner. An analog signal may be contrasted with a digital signal, which represents only discrete states.

ANI

automatic number identification

announcement

A message the system plays to the caller to provide information. The caller is not asked to give a response. Compare to “prompt.”

API

Application programming interface

application

The automated transaction (interactions) among the caller, the voice response system, and any databases or host computers required for your business. See also “application script.”

application administration

The component of the INTUITY CONVERSANT system 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 system invokes the TSM script assembler for the specific application name and moves files to the appropriate directories.

application script

The computer program that controls the application (the transaction between the caller and the system). The INTUITY CONVERSANT system provides several methods for creating application scripts, including Voice@Work, Script Builder, Transaction Assembler Script (TAS) language, and the Intuity Response Application Programming Interface (IRAPI).

application verification

A process in which the INTUITY CONVERSANT system verifies that all the components needed by an application are complete.

ASCII

American Standard Code for Information Interchange

ASI

analog switch integration

ASR

advanced speech recognition

asynchronous communication

A method of data transmission in which bits or characters are sent at irregular intervals and spaced by start and stop bits rather than by time. Compare to “synchronous communication.”

asynchronous data unit

An electronic communications device that allows computer systems to communicate over asynchronous lines more than 50 feet (15 m) in length.

automatic call distributor

That part of a telephone system that recognizes and answers incoming calls and completes these calls based on a set of instructions contained in a database. The ACD can send the call to an operator or group of operators as soon as the operator has completed a previous call or after the system has played a message to the caller.

automatic number identification

A method of identifying the calling party by automatically receiving a string of digits that identifies the calling station of a particular customer.

AYC5B

The IVP6 Tip/Ring (analog) circuit card.

AYC10

The IVC6 Tip/Ring (analog) circuit card.

AYC21

The E1/T1 (digital) circuit card.

AYC30

The NGTR (analog) circuit card.

AYC43

The speech and signal processor (SSP) circuit card.

B

back up
The preservation of the information in a file in a different location, so that the data is not lost in the event of hardware or system failure.

backing up an application
Using a utility that makes an archive copy of a completed application or an interim copy of an application in progress. The back-up copy can be restored to the system if the on-line version is damaged, or if you make revisions and want to go back to the previous version.

barge-in
A capability provided by WholeWord speech recognition and Dial Pulse Recognition (DPR) that allows callers to speak or enter their responses during the prompt and have those responses recognized (similar to the Speak with Interrupt capability). See also "echo cancellation."

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

BB
bulletin board

binary synchronous communications
A character-oriented synchronous link protocol.

blind transfer protocol
A protocol in which a call is completed as soon as the extension is dialed, without having to wait to see if the telephone is busy or if the caller answered.

bps
bits per second

BRDG
call bridging process

bridging
The process of connecting one telephone network connection to another over the INTUITY CONVERSANT system 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
binary synchronous communications

bundle
In the context of the Enhanced File Transfer package, this term is used to denote a single file, a group of files (package), or a combination of both.

byte
A unit of storage in the computer. On many systems, a byte is 8 bits (binary digits), which is the equivalent of one character of text.

C

call classification analysis

A process that enables application designers to use information available within the system to classify the disposition of originated and transferred calls. Intelligent CCA is provided with the system. Full CCA is an optional feature package.

call data event

A parameter that specifies a list of variables that are appended to a call data record at the end of each call.

call data handler process

A software process that accumulates generic call statistics and application events.

called party number

The number dialed by the person making a telephone call. Telephone switching equipment can use this number to selectively route an incoming call to a particular department or agent.

caller

The party who calls for a service, gets connected to the INTUITY CONVERSANT system, and interacts with it. As the INTUITY CONVERSANT system can also make outbound calls for service, the caller can also be the person who responds to those outbound calls.

call flow

See ”transaction.”

call progress tones

Standard telephony sounds that indicate the status of the call. These sounds include busy, fast busy, ringback, reorder, etc.

card cage

An area within a INTUITY CONVERSANT system 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.

CAS

channel associated signalling

caution

An admonishment or advisory statement used in INTUITY CONVERSANT system documentation to alert the user to the possibility of a service interruption or a loss of data.

CCA

call classification analysis

CDH

call data handler process

CELP

code excited linear prediction

central office

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

central processing unit

See "processor."

CGEN

Voice system general message class

channel

See "port."

channel associated signaling

A type of signaling that can be used on E1 circuit cards. It occurs on channel 16.

CICS

Customer Information Control System

circuit card upgrade

A new circuit card that replaces an existing card in the platform. Usually the replacement is an updated version of the original circuit card to replace technology made obsolete by industry trends or a new system release.

cluster controller

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

CMS

Call Management System

CO

central office

code excited linear prediction

A means of encoding analog voice signals into digital signals that provides excellent quality with use of minimum disk space.

command

An instruction or request the user issues to the system software to make the system perform a particular function. An entire command consists of the command name and options.

configuration

The arrangement of the software and hardware of a computer system or network. The INTUITY CONVERSANT system configuration includes either a standard or custom processor, peripheral equipment (for example, printers and modems), and software applications. Configuration also refers to the way the switch network is set up; that is, the types of products that are in the network and how those products communicate.

configuration management

The component of the system that allows you to manage the current configuration of voice channels, host sessions, and database connections, assign scripts to run on specific voice channels or host sessions, assign functionality to SSP and E1/T1 circuit cards, and perform various maintenance functions.

connect and disconnect (C and D) tones

DTMF tones that inform the system when the attendant has been connected (C) and when the caller has been disconnected (D).

connected digits
A sequence of digits that the system can process as a group, rather than requiring the caller to enter the digits one at a time.

Converse Data Return (conv_data)
A Script Builder action that supports the DEFINITY® call vectoring (routing) feature by enabling the switch to retain control of vector processing in the system environment. It supports the DEFINITY "converse" vector command to establish a two-way routing mechanism between the switch and the system to facilitate data passing and return.

controller circuit card
A circuit card used on a computer system that controls its basic functionality and makes the system operational. These circuit cards are used to control magnetic peripherals, video monitors, and basic system communications.

copying an application
A utility in which information from a source application is directed into the destination application.

coresidency
The ability of two products or services to operate and interact with each other on a single hardware platform. An example of this is the use of an INTUITY CONVERSANT system along with a package from a different vendor on the same system platform.

CPE
customer provided equipment or customer premise equipment

CPN
called party number

CPT
call progress tones

CPU
central processing unit

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

CSU
channel service unit

custom speech
Unique words or phrases to be used in INTUITY CONVERSANT system voice prompts that Lucent Technologies custom records on a per-customer basis.

custom vocabulary
A specialized package of unique words or phrases created on a per-customer basis and used by WholeWord or FlexWord speech recognition.

Customer Information Control System
Part of the operating system that manages resources for running applications (for example, IND$FILE). Note that TSO and CMS provide analogous functionality in other host environments.

CVS
converse vector step

D

danger
An admonishment or advisory statement used in INTUITY CONVERSANT system documentation to alert the user to the possibility of personal injury or death.

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 record
The information in a database for a person, product, event, etc. The database record is made up of individual fields for each information item.

database table
A structure, made up of columns and rows, that holds information in a database. Database tables provide a means of storing information that changes too often to “hard-code,” or store permanently, in the transaction outline.

dB
decibel

DB
database

DBC
database checking process

DBMS
database management system

DC
direct current

DCE
data communications equipment

DCP
digital communications protocol

debug
The process of locating and correcting errors in computer programs; also referred to as “troubleshooting.”

default
The way a computer performs a task in the absence of other instructions.

default owner
The owner of a channel when no process takes ownership of that channel. The default owner holds all idle, in-service channels. In terms of the IRAPI, this is typically the Application Dispatch process.

diagnose
The process of performing diagnostics on a bus or on Tip/Ring, E1/T1, or SSP circuit cards.

dial ahead
The ability to collect and process touch-tone inputs in sequence, even when they are received before the prompts.

dial pulse recognition
A method of recognizing caller pulse inputs from a rotary telephone.

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

dial through
A capability provided by touch-tone and dial pulse recognition that allows callers to enter their responses during the prompt and have those responses recognized (similar to the Speak with Interrupt capability). See also “barge-in” and “echo cancellation”.

dictionary
A reference book containing an alphabetical list of words, with information given for each word including meaning, pronunciation, and etymology.

DIMM
dual in-line memory module

DIO
disk input and output process

DIP
data interface process

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

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

DMA
direct memory address

DNIS
dialed number identification service

DPR
dial pulse recognition

DSP
digital signal processor

DTE
data terminal equipment

DTMF
dual tone multi-frequency

DTR
data terminal ready

dual 3270 links

A feature that provides an additional physical unit (PU) for a cost-effective means of connecting to two host computers. The customer can connect a system to two separate FEPs or to a single FEP shared by one or more host computers. Each link supports a maximum of 32 LUs.

dual tone multi-frequency

A touch-tone sound that is an audio signal including two different frequencies. *DTMF feedback* is the process of the "switch" providing this information to the system. *DTMF muting* is the process of ignoring these tones (which might be simulated by human speech) when they are not needed for the application.

dump space

An area of the disk that is fixed in size and should equal the amount of RAM on the system. The operating system "dumps" an image of core memory when the system crashes. The dump can be fetched after rebooting to help in analyzing the cause of the crash.

E

E&M

Ear and Mouth

E1 / T1

Digital telephony interfaces, commonly called *trunks*. E1 is an international standard at 2.048 Mbps. T1 is a North American standard at 1.544 Mbps.

Ear and Mouth

A common T1 trunking protocol for connection between two "switches."

EBCDIC

Extended Binary Coded Decimal Interexchange Code

echo cancellation

The process of making the channel quiet enough so that the system can hear and recognize WholeWord and dial pulse inputs during the prompt. See also "barge-in."

ECS

Enterprise Communications Server

editor system

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

EFT

Enhanced File Transfer

EIA

Electronic Industries Association

EISA

Extended Industry Standard Architecture

EMI

electromagnetic interference

Enhanced Basic Speech
Pre-recorded speech available from Lucent Technologies in several languages. Sometimes called "standard speech."

Enhanced File Transfer
A feature that allows the transferring of files automatically between the INTUITY CONVERSANT system 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.

Enterprise Communications Server
The telephony equipment that connects your business to the telephone network. Sometimes called a "switch."

error message
A message on the screen indicating that something is wrong with a possible suggestion of how to correct it.

ESD
electrostatic discharge

ESDI
Enhanced Serial Data Interface

ESS
electronic switching system

EST
Enhanced Software Technologies, Inc.

ET
error tracker

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

event
The notification given to an application when some condition occurs that is generally not encountered in normal operation.

EXTA
external alarms feature message class

external actions
Specific predefined system tasks that Script Builder can call or *invoke* to interact with other products or services. When an external action is invoked, the systems displays a form that provides choices in each field for the application developer to select. Examples are Call_Bridge, Make_Call, SP_Allocate, SR_Prompt, etc. In Voice@Work, external actions are treated as "external functions."

external functions
Specific predefined (or customer-created) system tasks that Voice@Work or Script Builder can call or *invoke* to interact with other products or services. The function allows the application developer to enter the argument(s) for the function to act on. Examples are concat, getarg, length, substring, etc. See also "external actions."

F

FAX Actions
An optional feature package that allows the system to send fax messages.

FCC
Federal Communications Commission

FDD
floppy disk drive

feature
A function or capability of a product or an application within the INTUITY CONVERSANT system.

feature package
An optional package that may contain both hardware and software resources to provide additional functionality to a standard system.

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

FEP
front end processor

FFE
Form Filler Plus feature message class

field
See "database field."

FIFO
first-in-first-out processing order

file
A collection of data treated as a basic unit of storage.

file transfer
An option that allows you to transfer files interactively or directly to and from UNIX using the file transfer system (FTS).

filename
Alphabetic characters used to identify a particular file.

FlexWord™ speech recognition
A type of speech recognition based on subword technology that recognizes phonemes or parts of words in a specific language. See also "subword technology."

foos
facility out-of-service state

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

FTS
file transfer process message class

Full CCA

A feature package that augments the types of call dispositions that Intelligent CCA can provide.

function key

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

G

GEN

PRISM logger and alerter general message class

grammar

The inputs that a recognizer can match (identify) from a caller.

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 for applications. The editing is done on the INTUITY CONVERSANT system.

GSE

Graphical Speech Editor

GUI

graphical user interface

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 Resource Allocator

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

hardware upgrade

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

HDD

hard disk drive

High Level Language Applications Programming Interface

An application programming interface that allows a user to write custom applications that can communicate with a host computer via an API.

HLLAPI
High Level Language Applications Programming Interface

HOST
host interface process message class

host computer
A computer linked to a network to provide a range of services, such as database access and computation. The host computer operates in a time-sharing manner with other computers linked to it via the network.

hwoos
hardware out-of-service state

Hz
Hertz

I

IBM
International Business Machines

iCk or ICK
The system integrity checking process.

ID
identification

IDE
integrated disk electronics

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

IE
information element

IND$FILE
The standard SNA file transfer utility that runs as an application under CICS, TSO, and CMS. IND$FILE is independent of link-level protocols such as BISYNC and SDLC.

independent software vendor
A company that has an agreement with Lucent Technologies to develop software to work with the INTUITY CONVERSANT system to provide additional features required by customers.

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

industry standard architecture
A PC bus standard that allows processors and other circuit cards to communicate with each other.

INIT
voice system initialization message class

initialize
To start up the system for the first time.

inserv
in-service state

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

Integrated Voice Processing (IVP) circuit card
The IVP6 circuit card that provides Tip/Ring connections. The NGTR (AYC30) card also provides the same functions.

intelligent CCA
Monitoring 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. The monitoring capabilities are dependent on the network interface circuit card and protocol used

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

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

INTUITY Response Application Programming Interface
A library of commands that provide a standard development interface for voice-telephony applications.

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

IVC6 circuit card (AYC10)
A Tip/Ring (analog) circuit card with six channels.

IVP6 circuit card (AYC5B)
A Tip/Ring (analog) card with six channels.

K

Kbps
kilobytes per second

Kbyte
kilobyte

keyboard mapping
In emulation mode, this feature enables the keyboard to send 3270 keyboard codes to the host according to a configuration table set up during installation.

keyword spotting
A capability provided by WholeWord speech recognition that allows the system to recognize a single word in the middle of an entire phrase spoken by a caller in response to a prompt.

L

LAN
local area network

LDB
local database

LED
light-emitting diode

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

LIFO
last-in-first-out processing order

line side E1
A digital method of interfacing an INTUITY CONVERSANT system to a PBX or “switch” using E1-related hardware and software.

line side T1
A digital method of interfacing an INTUITY CONVERSANT system to a PBX or “switch” using T1-related hardware and software.

listfile
An ASCII catalog that lists the contents of one or more talkfiles. Each application script is typically associated with a separate listfile. The listfile maps speech phrase strings used by application scripts into speech phrase numbers.

local area network
A data communications network in a limited geographical area. The LAN provides communications between computers and peripherals.

local database
A database residing on the INTUITY CONVERSANT system.

LOG
INTUITY CONVERSANT system logger process message class

logical unit
A type of SNA Network Addressable Unit.

logdaemon
A UNIX system information and error logging process.

logger
See “logdaemon.”

logging on/off
Entering or exiting the INTUITY CONVERSANT system software.

LSE1
line side E1

LST1
line side T1

LU
logical unit

M

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

main screen
The INTUITY CONVERSANT system screen from which you are able to enter either the System Administration or Voice System Administration menu.

maintenance process
A software process that runs temporary diagnostics and maintains the state of circuit cards and channels.

manoos
manually out-of-service state

MAP/100P
multi application platform 100P

MAP/100C
multi application platform 100C

MAP/40P
multi application platform 40P

MAP/5P
multi application platform 5P

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

master
A circuit card that provides clock information to the TDM bus.

Mbps
megabits per second

MByte
megabyte

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

menu
Options presented to a user on a computer screen or with voice prompts.

MF
multifrequency

MHz
megahertz

Microsoft
A manufacturer of 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
millisecond

msec
millisecond

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

MTC
maintenance process

multifrequency
Dual tone digit signalling (similar to DTMF), used for trunk addressing between network switches or by network operators.

multithreaded 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

NCP
Network Control Program

NEBS
Network Equipment Building Standards

NEMA
National Electrical Manufacturers Association

netoos
network out-of-service state

NetView
An optional feature package that transmits high-priority (major or critical) messages to the host as operator-generated alerts (OGAs) over the 3270 host link. The NetView Alarm feature package does not require a dedicated LU.

next generation Tip/Ring (AYC30) circuit card
An analog circuit card with six channels.

NFAS
non-facility associated signalling

NFS
network file sharing

NGTR
next generation Tip/Ring (AYC30) circuit card

NM-API
Network Management - Application Programming Interface

NMVT
network management vector transport

nonex
nonexistent state

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

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

NRZ
non return to zero

NRZI
non return to zero inverted

null value
An entry containing no value. A field containing a null value is normally displayed as blank and is different from a field containing a value of zero.

O

obsolete hardware
Hardware that is no longer supported on the INTUITY CONVERSANT system.

OEM
original equipment manufacturer

OGA
operator-generated alert

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

operator-generated alert
A system-monitoring message that is transmitted from the INTUITY CONVERSANT system or other computer system to an IBM host computer and is classified as critical or major.

option
An argument used in a command line to modify program output by modifying the execution of a command. When you do not specify any options, the command executes according to its default options.

ORACLE
A company that produces relational database management software. It is also used as a generic term that identifies a database residing on a local or remote system that is created and maintained using an ORACLE RDBMS product.

P

P&C
Prompt and Collect Script Builder action step

PBX
private branch exchange

PC
personal computer

PCB
printed circuit board

PCI
peripheral component interconnect

PCM
pulse code modulation

PEC
price element code

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

peripheral component interconnect
A newer, higher speed PC bus that is gradually displacing ISA for many components.

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

phoneme

A single basic sound of a particular spoken language. For example, the English language contains 40 phonemes that represent all basic sounds used with the language. The English word "one" can be represented with three phonemes, "w" - "uh" - "n." Phonemes vary between languages because of guttural and nasal inflections and syllable constructs.

phrase filtering (screening)

The rejection of unrecognized speech. The WholeWord and FlexWord speech recognition packages can be programmed to reprompt the caller if the INTUITY CONVERSANT system does not recognize a spoken response.

phrase tag

A string of up to 50 characters that identifies the contents of a speech phrase used by an application script.

platform migration

See "platform upgrade."

platform upgrade

The process of replacing the existing platform with a new platform.

pluggable

A term usually used with speech technologies, in particular standard speech, to indicate that a basic algorithmic technique has been implemented to accept one or more sets of parameters that tailors the algorithm to perform in one or more languages.

poll

A message sent from a central controller to an individual station on a multipoint network inviting that station to send if it has any traffic.

polling

A network arrangement whereby a central computer asks each remote location whether it wants to send information. This arrangement enables each user or remote data terminal to transmit and receive information on shared facilities.

port

A connection or link between two devices that allows information to travel to a desired location. See "telephone network connection."

PRI

Primary Rate Interface

Primary Rate Interface

An ISDN term for connections over E1 or T1 facilities that are usually treated as trunks.

private branch exchange

A private switching system, either manual or automatic, usually serving an organization, such as a business or government agency, and usually located on the customer's premises.

processor

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

prompt

A message played to a caller that gives the caller a choice of selections in a menu and asks for a response. Compare to "announcement."

prompt and collect (P and C)
A message played to a caller that gives the caller a choice of selections in a menu and asks for a response. The responses is collected and the script progresses based on the caller's response.

pseudo driver
A driver that does not control any hardware.

PS&BM
power supply and battery module

PSTN
public switch telephone network

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

R

RAID
redundant array of independent disks

RAID array
An assembly of disk drives configured to provide some level of RAID functionality.

RAM
random access memory

RDMBS
ORACLE relational database management system

RECOG
speech recognition feature message class

recognition type
The type of input the recognizer can understand. Available types include touch-tone, dial pulse, and Advanced Speech Recognition (ASR), which includes WholeWord and FlexWord speech recognition.

recognizer
The part of the system that compares caller input to a grammar in order to correctly match (identify) the caller input.

record
See "database record."

recovery
The process of using copies of the INTUITY CONVERSANT system software to reconstruct files that have been lost or damaged. See also "restore."

remote database
Information stored on a system other than the INTUITY CONVERSANT system that can be accessed by the INTUITY CONVERSANT system.

remote maintenance circuit card

An INTUITY CONVERSANT system circuit card, available with a built-in modem, that allows remote personnel (for example, field support) to access all INTUITY CONVERSANT system machines. This card is standard equipment on all new MAP/100, MAP/40, and MAP/5P purchases.

REN

ringer equivalence number

reports administration

The component of INTUITY CONVERSANT system that provides access to system reports, including call classification, call data detail, call data summary, message log, and traffic reports.

restore

The process of recovering lost or damaged files by retrieving them from available back-up tapes or from another disk device. See also "recovery."

restore application

A utility that replaces a damaged application or restores an older version of an application.

reuse

The concept of using a component from a source system in a target system after a software upgrade or platform migration.

RFS

remote file sharing

RM

resource manager

RMB

remote maintenance circuit card

roll back

To cancel changes to a database since the point at which changes were last committed.

rollback segment

A portion of the database that records actions that should be undone under certain circumstances. Rollback segments are used to provide transaction rollback, read consistency, and recovery.

RTS

request to send

S

SBC

sub-band coding

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 system to follow during a transaction.

Script Builder
An optional software package that provides a menu-oriented interface designed to assist in the development of custom voice response applications on the INTUITY CONVERSANT system (see also “Voice@Work”).

SCSI
small computer system interface

SDLC
synchronous data link control

SDN
software defined network

shared database table
A database table that is used in more than one application.

shared speech
Speech that is a part of more than one application.

shared speech pools
A parameter that allows the user of a voice application to share speech components with other applications.

SID
station identification

signal processor circuit card (AYC2, AYC2B, AYC2C, or AYC9d)
A speech processing circuit card that is an older, lower-capacity version of the speech and signal processor (SSP) circuit card (AYC43).

SIMMs
single inline memory modules

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.

SLIP
serial line interface protocol

small computer system interface
A disk drive control technology in which a single SCSI adapter circuit card plugged into a PC slot is capable of controlling as many as seven different hard disks, optical disks, tape drives, etc.

SNA
systems network architecture

SNMP
simple network management protocol

software
The set or sets of programs that instruct the computer hardware to perform a task or series of tasks — for example, UnixWare software and the INTUITY CONVERSANT system software.

software upgrade
The installation of a new version of software in which the existing platform and circuit cards are retained.

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

speech and signal processor circuit card (AYC43)
The high-performance signal processing circuit card introduced in V6.0 capable of simultaneous support for various speech technologies.

speech energy
The amount of energy in an audio signal. Literally translated, it is the output level of the sound in every phonetic utterance.

speech envelope
The linear representation of voltage on a line. It reflects the sound wave amplitude at different intervals of time. This envelope can be plotted on a graph to represent the oscillation of an audio signal between the positive and negative extremes.

speech file
A file containing an encoded speech phrase.

speech filesystem
A collection of several talkfiles. The filesystem is organized into 16-Kbyte blocks for efficient management and retrieval of talkfiles.

speech modeling
The process of creating WholeWord speech recognition algorithms by collecting thousands of different speech samples of a single word and comparing them all to obtain a statistical average of the word. This average is then used by a WholeWord speech recognition program to recognize a single spoken word.

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

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

speech recognition
The ability of the system to understand input from callers.

SPIP
signal processor interface process

SPPLIB
speech processing library

SQL
structured query language

SR
speech recognition

SSP
speech and signal processor circuit card (AYC43)

standard speech
The speech package available in several languages containing simple words and phrases produced by Lucent Technologies for use with the INTUITY CONVERSANT system. This package includes digits, numbers, days of the week, and months, each spoken with initial, medial, and falling inflection. The speech is in digitized files stored on the hard disk to be used in voice prompts and messages to the caller. This feature is also called Enhanced Basic Speech.

standard vocabulary
A standard package of simple word speech models provided by Lucent Technologies and used for WholeWord speech recognition. These phrases include the digits “zero” through “nine,” “yes,” “no,” and “oh,” or the equivalent words in a specific local language.

string
A contiguous sequence of characters treated as a unit. Strings are normally bounded by white spaces, tabs, or a character designated as a separator. A string value is a specified group of characters symbolized by a variable.

structured query language
A standard data programming language used with data storage and data query applications.

subword technology
A method of speech recognition used in FlexWord recognition that recognizes phonemes or parts of words. Compare to “WholeWord speech recognition.”

switch
A software and hardware device that controls and directs voice and data traffic. A customer-based switch is known as a “private branch exchange.”

switch hook
The device at the top of most telephones that is depressed when the handset is resting in the cradle (in other words, is *on hook*). The device is raised when the handset is picked up (in other words, when the telephone is *off hook*).

switch hook flash
A signaling technique in which the signal is originated by momentarily depressing the “switch hook.”

switch interface administration
The component of the INTUITY CONVERSANT system that enables you to define the interaction between the INTUITY CONVERSANT system 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 telephone switching systems.

synchronous communication
A method of data transmission in which bits or characters are sent at regular time intervals, rather than being spaced by start and stop bits. Compare to “asynchronous communication.”

SYS
UNIX system calls message class

sysgen
system generation

System 75
An advanced digital switch supporting up to 800 lines that provides voice and data communications for its users.

System 85
An advanced digital switch supporting up to 3000 lines that provides voice and data communications for its users.

system administrator
The person assigned the responsibility of monitoring all INTUITY CONVERSANT system software processing, performing daily system operations and preventive maintenance, and troubleshooting errors as required.

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

system message
An event or alarm generated by either the INTUITY CONVERSANT system or end-user process.

system monitor
A component of the INTUITY CONVERSANT system that tests to verify that each incoming telephone line and its associated Tip/Ring or T1 circuit card is functional. Through the “System Monitor” component, you are able to see displays of the Voice Channel and Host Session Monitors.

T

T1
A digital transmission link with a capacity of 1.544 Mbps.

table
See “database table.”

talkfile
An ASCII file that contains the speech phrase tags and phrase tag numbers for all the phrases of a specific application. The speech phrases are organized and stored in groups. Each talkfile can contain up to 65,535 phrases, and the speech filesystem can contain multiple talkfiles.

talkoff
The process of a caller interrupting a prompt, so the prompt message stops playing.

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

TAS
transaction assembler script

TCC
Technology Control Center

TCP/IP
transmission control protocol/internet protocol

TDM
time division multiplexing

TE

terminal emulator

telephone network connection

The point at which a telephone network connection terminates on an INTUITY CONVERSANT system. Supported telephone connections are Tip/Ring, T1, and E1.

terminal emulator

Software that allows a PC or UNIX process to look like a specific type of terminal. In particular, it allows the INTUITY CONVERSANT system to temporarily transform itself into a “look alike” of an IBM 3270 terminal. In addition to providing full 3270 functionality, the terminal emulator enables you to transfer files to and from UNIX.

Text-to-Speech

An optional feature that allows an application to play US English speech directly from ASCII text by converting that text to synthesized speech. The text can be used for prompts or for text retrieved from a database or host, and can be spoken in an application with prerecorded speech. Text-to-Speech application development is supported through Voice@Work and Script Builder.

ThickNet

A 10-mm (10BASE5) coaxial cable used to provide interLAN communications.

ThinNet

A 5-mm (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

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

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

transaction assembler script

The computer program code that controls the application operating on the voice response system. The code can be produced from Voice@Work, Script Builder, or by writing directly in TAS code.

transaction state machine process

A multi-channel IRAPI application that runs applications controlled by TAS script code.

transient process

A process that is created dynamically only when needed.

TRIP
Tip/Ring interface process

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

TSO
Technical Services Organization

TSO
time share operation

TSM
transaction state machine process

TTS
Text-to-Speech

TWIP
T1 interface process

U

UK
United Kingdom

US
United States of America

UNIX Operating System
A multiuser, multitasking computer operating system originally developed by Lucent Technologies.

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

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

usability
A measurement of how easy an application is for callers to use. The measurement is made by making observations and by asking questions. An application should have high usability to be successful.

USOC
universal service ordering code

UVL
unified voice library

V

VDC
video display controller

vi editor
A screen editor used to create and change electronic files.

virtual channel
A channel that is not associated with an interface to the telephone network (Tip/Ring, T1, LSE1/LST1, or PRI). Virtual channels are intended to run “data-only” applications which do not interact with callers but may interact with DIPs. Voice or network functions (for example, coding or playing speech, call answer, origination, or transfer) will not work on a virtual channel. Virtual channel applications can be initiated only by a “virtual seizure” request to TSM from a DIP.

vocabulary
A collection of words that the INTUITY CONVERSANT system 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@Work
An optional software package that provides a graphical interface to assist in development of voice response applications on the INTUITY CONVERSANT system (see also “Script Builder”).

voice channel
A channel that is associated with an interface to the telephone network (Tip/Ring, T1, E1, LSE1/LST1, or PRI). Any INTUITY CONVERSANT system application can run on a voice channel. Voice channel applications can be initiated by being assigned to particular voice channels or dialed numbers to handle incoming calls or by a “soft seizure” request to TSM from a DIP or the **soft_szr** command.

voice processing co-marketer
A company licensed to purchase voice processing equipment, such as the INTUITY CONVERSANT system, 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 SSP circuit cards) and data storage devices (for example, hard disk, etc.)

voice response unit
A computer connected to a telephone network that can play messages to callers, recognize caller inputs, access and update a databases, and transfer and monitor calls.

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

VPC
voice processing co-marketer

VROP
voice response output process

VRU
voice response unit

W

warning
An admonishment or advisory statement used in INTUITY CONVERSANT system documentation to alert the user to the possibility of equipment damage.

WholeWord speech recognition
An optional feature, available in several languages, based on whole-word technology that can recognize the numbers one through zero, “yes”, and “no” (the key words). This feature is reliable, regardless of the individual speaker. This feature can identify the key words when spoken in phrases with other words. A string of key words, called *connected digits*, can be recognized. During the prompt announcement, the caller can speak or use touch tones (or dial pulses, if available). See also “whole-word technology.”

whole-word technology
The ability to recognize an entire word, rather than just the phoneme or a part of a word. Compare to “subword technology.”

wink signal
An interruption of current to a busy lamp indicating that there is a line on hold.

word
A unique utterance understood by the recognizer.

wordlist
A set of words available for FlexWord recognition by an application during a Prompt & Collect action step.

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

Index