Lucent Technologies
Bell Labs Innovations

# INTUITY™ CONVERSANT® System

Version 6.0
Speech Development, Processes,
and Recognition

585-310-762
Comcode 107852436
Issue 2.0
December 1996

**Copyright © 1996, Lucent Technologies**
**All Rights Reserved**
**Printed in U.S.A.**

**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 shoud 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 Technical Service Center Toll Fraud Intervention Hotline at 1 800 643-2353.

**Federal Communications Commission Statement**
**Part 15: Class B Statement.** This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses, and can radiate radio-frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient the receiving television or radio antenna where this may be done safely.
- To the extent possible, relocate the receiver with respect to the telephone equipment.
- Where the telephone equipment requires AC power, plug the telephone into a different AC outlet so that the telephone equipment and receiver are on different branch circuits.

**Part 15: Personal Computer Statement.** This equipment has been certified to comply with the limits for a Class B computing device, pursuant to Subpart J of Part 15 of FCC Rules. Only peripherals (computing input/output devices, terminals, printers, etc.) certified to comply with the Class B limits may be attached to this computer. Operation with noncertified peripherals is likely to result in interference to radio and television reception.

**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 317 361-5353
Fax 1 800 457-1764 International Fax 317 361-5355
**Write**: Lucent Technologies Publications Center
P.O. Box 4100
Crawfordsville, IN 47933
**Order**: Document No. 585-310-762
Comcode 107852436
Issue 1.0, December 1996

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 the Product Documentation, Lucent Technologies, Columbus, OH.

RECYCLED PAPER

# Contents

About This Book xi

- Purpose xi
- Intended Audiences xi
- How This Book Is Organized xii
- How to Use This Book xiii
- Conventions Used in This Book xiv
  - Terminology xiv
  - Terminal Keys xvii
  - Screen Displays xvii
  - Other Typography xviii
  - Safety Labels xix
- Related Resources xix
  - Documentation xix
  - Training xx
- Electronic Updates to This Book xx
- Trademarks and Service Marks xxi
- How To Make Comments About This Book xxii
- Disclaimer xxii

1 Overview of Speech 1-1

- Overview 1-1
- Purpose 1-1
- Understanding the Speech Filesystem 1-2
  - Default Speech Directory 1-2
  - Specifying a Speech Directory 1-3
  - Saving and Restoring 1-3
  - Adding a Second Speech Disk 1-4
  - Recovering from a Corrupted Speech Disk 1-4
  - Changing the Size of the Speech Filesystem 1-4
  - Creating and Removing Filesystems 1-4
  - Displaying Filesystem Attributes 1-4
  - Defining Phrases 1-5

# Contents

Defining Phrase Tags 1-5
Defining Phrase Numbers 1-6
Defining Talkfiles 1-6
Defining Speech Files 1-7
Defining Speech Filesystem 1-7

- Overview of Speech Development Features 1-7

Script Builder 1-7
Text-to-Speech 1-8
Graphical Designer Release 1.6 1-9
Speech Recognition 1-11
Dial Pulse Recognition 1-14
Form Filler Plus 1-14

- SSP Circuit Card 1-16

Support 1-16
Memory 1-16
Capacity 1-17

- Signal Processor (SP) and Companion (CMP) Circuit Card 1-18

Support 1-18
Memory 1-18
Capacity 1-18
Companion Circuit Card 1-19

2 Developing Speech 2-1

- Overview 2-1
- Purpose 2-1
- Overview of Creating Speech 2-2
- Determining the Transaction 2-3
- Planning the Voice Script 2-3
- Writing the Voice Script 2-5

Using Framing in Voice Scripts 2-5
Analyzing Speech Inflections 2-7

- Selecting a Speech Development Method 2-7

# Contents

Hiring a Professional Speaker 2-8

Producing Self-Recorded Custom Speech 2-9

Purchasing the Enhanced Basic Speech Package 2-9

Purchasing a Lucent Technologies Custom Speech Package 2-10

Using Text-to-Speech 2-10

Using the Graphical Speech Editor 2-11

Sharing Speech 2-12

Importing Speech 2-12

- Recording Speech 2-13

Using a Professional Speaker 2-13

Purchasing a Custom Speech Package from Lucent Technologies 2-15

Producing Self-Recorded Custom Speech with Script Builder 2-16

- Digitizing Speech Phrases 2-16
- Installing Speech 2-17

Installing Speech in a Script Builder or TSM Application 2-17

Installing Speech in an IRAPI Application 2-17

3 **Editing Speech** 3-1

- Overview 3-1
- Purpose 3-1
- Introduction to the Graphical Speech Editor 3-2

Making Changes with GSE 3-2

Working with Segments of Speech 3-2

- Accessing the Graphical Speech Editor 3-3

Installing and Administering the Mouse 3-3

Installing the Graphical Speech Editor 3-3

Setting Up Graphical Speech Editor Software 3-3

Initiating the Graphical Speech Editor 3-4

- Graphical Speech Editor User Interface 3-5

Title Bar 3-5

Menu Bar 3-5

Tool Bar 3-8

Speech Display Area 3-10

Scroll Bar 3-11

Information Bar 3-12

- Editing a Speech File with GSE 3-13

Using the Mouse 3-13

Opening a File 3-14

Setting the Marker 3-15

Selecting a Segment of Speech 3-15

Listening to Speech 3-16

Cutting and Pasting Speech 3-17

Copying and Pasting Speech 3-18

Adjusting the Volume 3-19

Undoing Changes 3-19

Saving Changes 3-20

Exiting 3-20

- Converting Speech Files 3-21

Retrieving Files from Speech Pools 3-22

Restoring Files to Speech Pools 3-23

Retrieving Files from the Speech Filesystem 3-24

Restoring Files to the Speech Filesystem 3-26

- Using Script Builder for Editing 3-26

**4 Recognizing WholeWord Speech Input** 4-1

- Overview 4-1
- Purpose 4-1
- Introduction to WholeWord Speech Recognition 4-2
- Components of WholeWord Speech Recognition 4-2

Global Support 4-2

Bilingual WholeWord Speech Recognition Capabilities 4-3

Standard Vocabulary 4-4

## Contents

Standard WholeWord Speech Recognition Types 4-5
Connected-Digit Recognition 4-9
Recognition DIP 4-10
Word Spotting 4-10
Speaker Independence 4-11
Phrase Screening 4-11
Recognition Confirmation 4-11
Barge-In 4-12
Dual Tone Multifrequency (DTMF) Support 4-13
Custom Grammars 4-14
Custom Vocabulary 4-14
Custom DIPs 4-14

- WholeWord Speech Recognition Accuracy 4-15

Positive Influences on WholeWord Speech Recognition Accuracy 4-15
Negative Influences on WholeWord Speech Recognition Accuracy 4-17

- How WholeWord Speech Recognition Works 4-19

Caller Response 4-19
The Recognizer 4-19
Computational Processing 4-20
Further Processing 4-20

5 **Recognizing FlexWord™ Speech Input** 5-1

- Overview 5-1
- Purpose 5-1
- Introduction to FlexWord Speech Recognition 5-2

Defining Phonemes 5-2
FlexWord’s Capacity 5-5

- Components of FlexWord Speech Recognition 5-6

FlexWord Toolkit 5-6
Vocabularies 5-6
Keyword Spotting 5-7

Phrase Screening 5-7
Global Support 5-7
Barge-In 5-8
Further Processing 5-8
- Preparing a FlexWord Speech Recognition Application 5-8
Designing the FlexWord Speech Recognition Application 5-8
- Creating Wordlists 5-12
Purchasing a Custom Speech Package from Lucent Technologies 5-12
Using the FlexWord Toolkit 5-12
- Special FlexWord Recognition Accents 5-23
Spanish FlexWord Recognition Accents 5-23
Japanese FlexWord Recognition Accents 5-23
French FlexWord Recognition Accents 5-24
German FlexWord Recognition Accents 5-25
- FlexWord Recognition Vocabulary Administration 5-25
Installing Wordlists 5-25
Activating Wordlists 5-26
- FlexWord Speech Recognition Accuracy 5-28
Positive Influences on FlexWord Speech Recognition Accuracy 5-28
Negative Influences on FlexWord Speech Recognition Accuracy 5-32
- WholeWord Speech Recognition Versus FlexWord Speech Recognition Accuracy 5-33

6 **Recognizing Dial Pulse Input** 6-1
- Overview 6-1
- Purpose 6-1
- Introduction to Dial Pulse Recognition 6-2
- Components of Dial Pulse Recognition 6-2
Global Support 6-2

Simultaneous Dial Pulse Capabilities 6-2

Standard Dial Pulse Recognition Types 6-3

Grammars 6-3

Recognition Confirmation 6-4

Barge-In 6-4

- How Dial Pulse Recognition Works 6-5

Computational Processing 6-5

The Recognizer 6-5

- Recognition Accuracy 6-6

Recognition 6-6

Training 6-7

DPR Troubleshooting 6-7

Common Failure Modes 6-8

7 **Putting It Together** 7-1

- Overview 7-1
- Purpose 7-1
- Using WholeWord and DPR Together 7-2
- Getting the Most Out of FlexWord™ Speech Recognition 7-3

Building a FlexWord Speech Recognition Vocabulary 7-3

Choosing a FlexWord Speech Recognition Vocabulary 7-3

Caller Error 7-4

Word Length 7-4

Word Choice 7-4

Prompting for Input 7-4

Sound of a Word 7-4

Synonyms 7-4

Examples of FlexWord Speech Recognition Applications 7-5

- Getting the Most out of Dial Pulse Recognition 7-6

- Getting the Most out of Text-to-Speech 7-6
- Using FlexWord Speech Recognition and Text-to-Speech 7-7
- Using Text-to-Speech and Prerecorded Speech 7-7
- Using WholeWord and FlexWord Speech Recognition Together 7-9
- WholeWord versus FlexWord Speech Recognition 7-11
- WholeWord Speech Recognition versus Dial Pulse Recognition 7-12

A Enhanced Basic Speech Formats A-1

- Overview A-1
- Purpose A-1
- Languages Available A-2
- Enhanced Basic Speech Formats A-2
  - Speech Pool Names and Abbreviations A-3
  - Australian English Enhanced Basic Speech Format Set A-4
  - Brazilian Portuguese Enhanced Basic Speech Format Set A-6
  - Canadian French Enhanced Basic Speech Format Set A-8
  - Cantonese Chinese Enhanced Basic Speech Format Set A-12
  - Castilian Spanish Enhanced Basic Speech Format Set A-14
  - Dutch Enhanced Basic Speech Format Set A-16
  - French Enhanced Basic Speech Format Set A-18
  - German Enhanced Basic Speech Format Set A-21
  - Hindi Enhanced Basic Speech Format Set A-23

Japanese Enhanced Basic Speech Format Set A-25

Latin-American Spanish Enhanced Basic Speech Format Set A-26

Mandrian Enhanced Basic Speech Format Set A-29

UK English Enhanced Basic Speech Format Set A-31

US English Enhanced Basic Speech Format Set A-33

B Speech File Formats B-1

- Overview B-1
- Purpose B-1
- Speech File Formats B-2
  - PCM Speech File Formats B-2
  - ADPCM Speech File Formats B-3
- About Speech Files B-4

C Calculating O.S. Index C-1

- Overview C-1
- Purpose C-1
- Calculating the O.S. Index C-2

D Advanced Text-to-Speech Features D-1

- Overview D-1
- Purpose D-1
- Introduction D-2
- Using Escape Sequences D-2
  - Silence Delays D-2

Speaking Rate on an SP Circuit Card D-4

Speaking Rate on an SSP Circuit Card D-5

Additional Text Classes D-6

Sample Application Using Escape Sequences D-10

E **Recognition Post-Processing** E-1

- Overview E-1
- Purpose E-1
- Further Processing E-2

Return Top 4 (RT4) E-4

Prefix List Check (LSC) E-5

Luhn Check (LNC) E-6

Expected Value (EXV) E-6

Close Match (CLM) E-7

Further Processing Prefixes and Combinations E-7

ABB **Abbreviations** ABB-1

GL **Glossary** GL-1

IN **Index** IN-1

# About This Book

## Purpose

Most applications, whether generated with Script Builder or developed using the transaction state machine (TSM) script-level language and/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 Audiences

The primary audiences for the *INTUITY™ CONVERSANT® System Version 6.0 Speech Development, Processing & Recognition* book are 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 (ISV) and Voice Processing CoMarketers (VPC).

## How This Book Is Organized

This document is organized into the following chapters:

- Chapter 1 — “Overview of Speech”

  This chapter provides an overview of speech on the INTUITY CONVERSANT System. Information about the speech file system and the different parts of speech are discussed. An overview of all of the speech-related features available for the development of applications on the system is also covered in this chapter.

- Chapter 2 — “Developing Speech”

  This chapter defines and 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”

  This chapter includes information on using the Graphical Speech Editor, Script Builder, and third-party speech editing systems for editing speech.

- Chapter 4 — “Recognizing WholeWord Speech Input”

  This chapter includes an introduction to speech recognition, along with a detailed discussion of WholeWord speech recognition. Also included is a section on creating vocabularies and improving the accuracy of speech recognition.

- Chapter 5 — “Recognizing FlexWord™ Speech Input”

  This chapter includes a detailed discussion of FlexWord speech recognition and it’s features.

- Chapter 6 — “Recognizing Dial Pulse Input”

  This chapter discusses the recognition of dial pulses on rotary or push button telephones.

- Chapter 7 — “Putting It Together”

  This chapter discusses the speech-related features and how to make them work well together in your application.

- Appendix A — “Enhanced Basic Speech Formats”

  This appendix provides a complete listing of all enhanced basic speech formats.

- Appendix B — “Speech File Formats”

  This appendix includes information about speech files and speech file formats, including Pulse Code Modulation (PCM) and Adaptive Differential Pulse Code Modulation (ADPCM).

- Appendix C— “Calculating O.S. Index”

  This appendix includes the procedure for computing channel numbers for the GSE.

- Appendix D— “Advanced Text-to-Speech Features”

  This appendix discusses several ways to customize synthesized speech, including using escape sequences to add silence delays, change the speaking rate, and mark text as belonging to a specific text category.

- Appendix E— “Recognition Post-Processing”

  This appendix describes Data Interface Process (DIP) associated with WholeWord speech recognition, and FlexWord speech recognition. It also establishes five new processing features that increase the system accuracy of recognition: Return Top 4, Prefix List Check, Luhn Check, Expected Value and Close Match.

- Abbreviations

  This section provides a list of abbreviations, including acronyms, used in system documentation.

- Glossary

  This section provides a definition of terms used in system documentation.

- Index

  This section provides an alphabetical listing of principal subjects.

## How to Use This Book

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

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

This book is organized in a logical, 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 (Chapters 4, 5, 6). Finally, you learn how to effectively use speech-related features together (Chapter 7).

# Conventions Used in This Book

This section describes the conventions used in this book.

## 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.

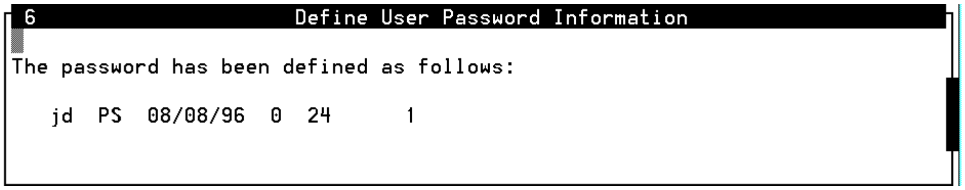
- The word "enter" means to type a value and then press ENTER. For example, an instruction to type the letter “y” and press ENTER is shown as

  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 *windows, screens,* and *menus*. Windows and screens both show and request system information (Figure 1 through Figure 4). Menus (Figure 5) present options from which you can choose to view another menu, or a screen or window.

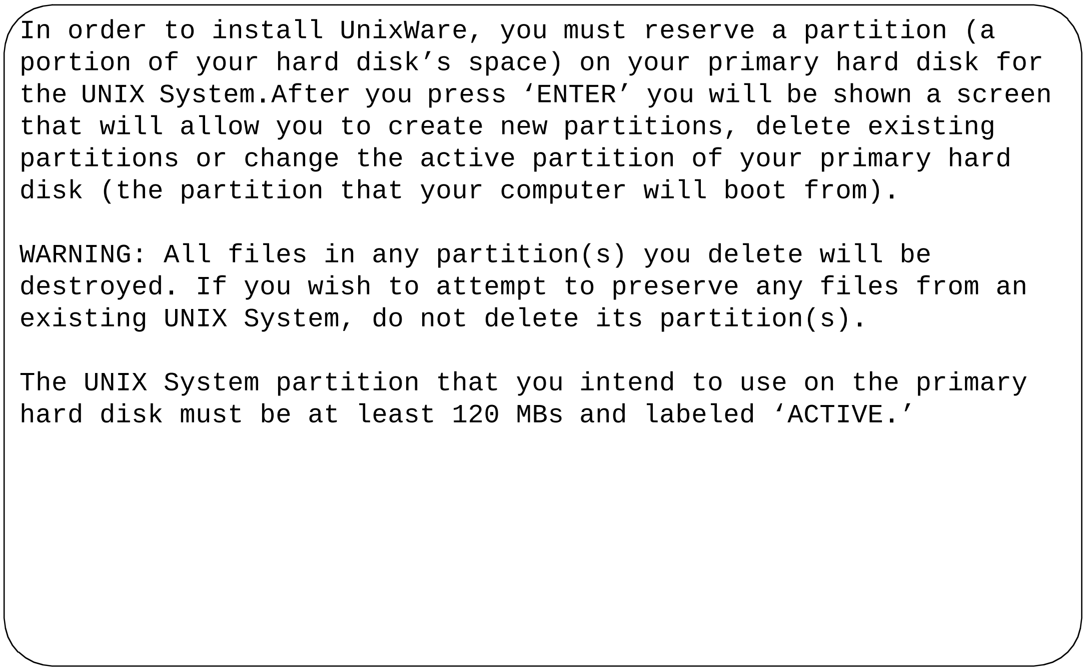


```
 6                    Define User Password Information

The password has been defined as follows:

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

**Figure 1. Example of an INTUITY CONVERSANT Window Showing Information**

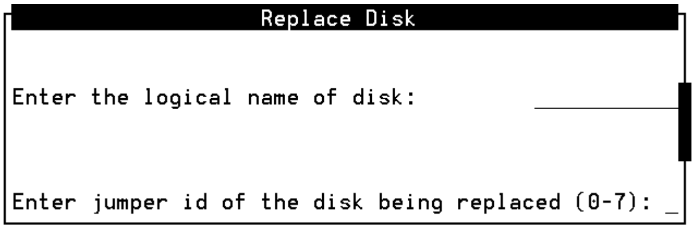


```
In order to install UnixWare, you must reserve a partition (a
portion of your hard disk's space) on your primary hard disk for
the UNIX System.After you press 'ENTER' you will be shown a screen
that will allow you to create new partitions, delete existing
partitions or change the active partition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

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

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

**Figure 2. Example of an INTUITY CONVERSANT Screen Showing Information**

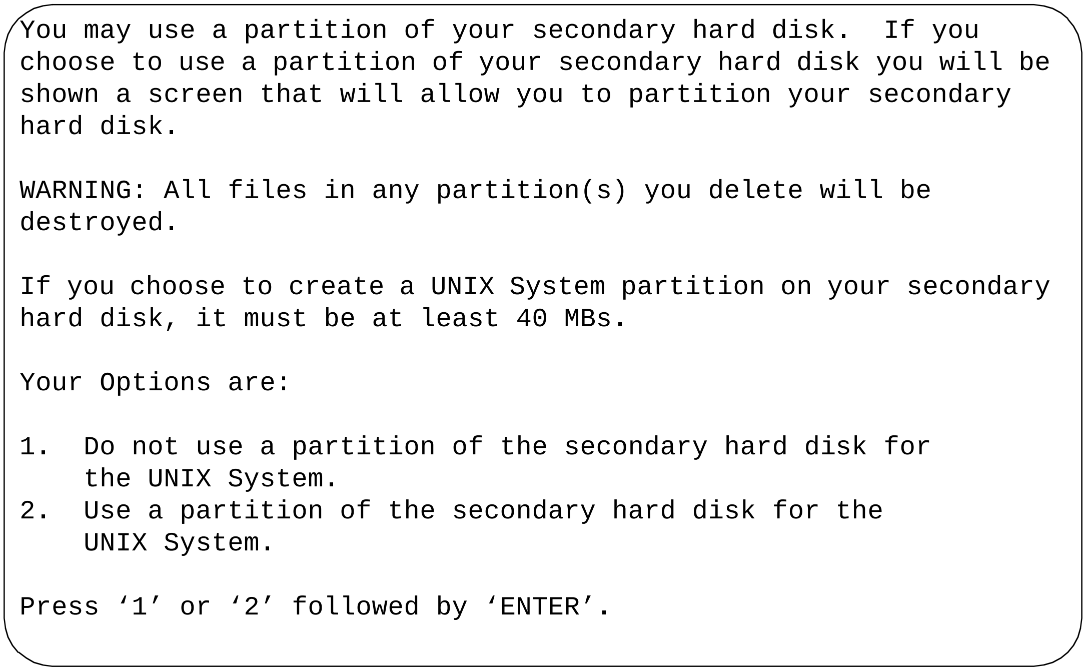


```
                        Replace Disk

Enter the logical name of disk:          __________


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

**Figure 3. Example of an INTUITY CONVERSANT Window Requesting Information**



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

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

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

Your Options are:

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

Press ‘1’ or ‘2’ followed by ‘ENTER’.
```

**Figure 4. Example of an INTUITY CONVERSANT Screen Requesting Information**

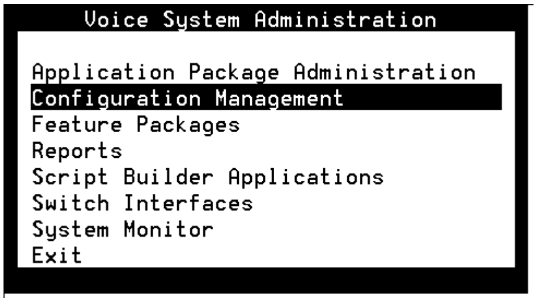

Voice System Administration
Application Package Administration
Configuration Management
Feature Packages
Reports
Script Builder Applications
Switch Interfaces
System Monitor
Exit

**Figure 5. Example of an INTUITY CONVERSANT Menu**

## Terminal Keys

- 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:

  Example 1:

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

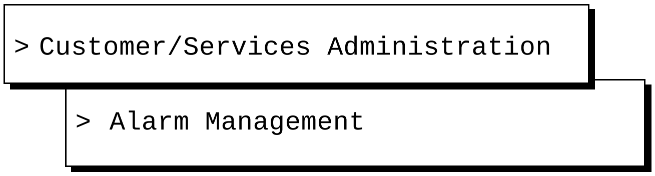
  Example 2:

  ```
  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:

  ```
  > Customer/Services Administration
      > Alarm Management
  ```

  In this example, you would access the INTUITY CONVERSANT Main Menu and select the Customer/Services Administration menu. From the Customer/Services Administration menu, you would then select the Alarm Management screen.

- 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:

  Example 1:

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

  Example 2:

  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

This section describes additional documentation and training available for you to learn more about the INTUITY CONVERSANT product.

### Documentation

**NOTE:**
The *INTUITY*™ *CONVERSANT® System Version 6.0 System Description*, 585-310-241, contains a detailed description of all books included in V6.0 INTUITY CONVERSANT documentation library. Always refer to the appropriate book for specific information on planning, installing, administering, or maintaining an INTUITY CONVERSANT system.

- *INTUITY™ CONVERSANT® System Version 6.0 System Description*, 585-310-241
- *INTUITY*™ *CONVERSANT® System Version 6.0 MAP/40 Maintenance*, 585-310-181
- *INTUITY*™ *CONVERSANT® System Version 6.0 MAP/100C Maintenance*, 585-310-180
- *INTUITY*™ *CONVERSANT® System Version 6.0 MAP/100 Maintenance*, 585-310-179
- *INTUITY™ CONVERSANT® System Version 6.0 Application Development with Script Builder*, 585-310-760

- *INTUITY™ CONVERSANT® System Version 6.0 Application Development with Advanced Methods*, 585-310-761
- *INTUITY™ CONVERSANT® System Version 6.0 MAP/40 New System Installation*, 585-310-178
- *INTUITY™ CONVERSANT® System Version 6.0 MAP/100C New System Installation*, 585-310-177
- *INTUITY™ CONVERSANT® System Version 6.0 MAP/100 New System Installation*, 585-310-176
- *INTUITY™ CONVERSANT® System Version 6.0 Administration, 585-310-591*
- *Veritas Documentation*, 585-350-906
- The Novell UnixWare Documentation Set, 585-450-908

### 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.

### 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

## 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.

## 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.
- 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.
- CLEO and LINKix are trademarks 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

## How To Make Comments About This Book

A comment card is included following the title page 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 located behind the title page.

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

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

Fax: (303)538-1741

Please include the name and document number of this book.

## 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 describes

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

## Purpose

The purpose of this chapter is to ensure that you

- Have an understanding of the software features available 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 your application resides as UnixWare files in a mounted UnixWare filesystem. By default, speech filesystems reside in **/voice1** which is linked to **/home2**. Similarly, these two may be used interchangeably. Users upgrading a V5.0 system have **/home2** which in turn is connected to **/voice1**. With the INTUITY™ CONVERSANT® system, you are able to define where in **/voice1** or **/home2** you want your speech files stored.

Each speech phrase requires a minimum of 8 Kbytes of space. Depending on your coding 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. Coding 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 more information about the encoding methods and capacities see “Speech Storage Capacities” and “Speech Administration Capacities” in Chapter 4, “Features,” in *INTUITY™ CONVERSANT® System Version 6.0 System Description*, 585-310-241.

## 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

To change the speech directory in which your speech files are stored (from the default), perform the following procedure:

1. Stop the voice system.

   For instructions on how to stop or start the voice system see "System Control" in Chapter 3, "Configuration Management," of *INTUITY*™ *CONVERSANT*® *System Version 6.0 Administration*, 585-310-591.

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 your 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 only available for use with speech files stored in the speech directory defined by the SPEECHDIR variable in the **irAPI.rc** file.

### Saving and Restoring

Speech will be backed up when a **mkimage** system backup is performed. The **spsav** (save) and **spres** (restore) commands are also available for backing up speech as described in *INTUITY*™ *CONVERSANT*® *System Version 6.0 Administration*, 585-310-591.

**CAUTION:**
*Use the* ***spsave*** *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 your system has more or less than 72 channels (telephone network connections), you have the option of adding a second disk for storing speech. See Chapter 6, "Replacing the Hard Disk Drive," in the INTUITY CONVERSANT maintenance book specific to your platform, for information on how to add a second disk. The second disk in your system is used to store speech filesystems that reside in **/home3**.

⚠ **CAUTION:**
*You must make sure that you use a tape large enough to hold all of the data.*

## Recovering from a Corrupted Speech Disk

For information on recovering from a corrupted speech disk, see Appendix D, "Disaster Recovery Checklists," in the INTUITY CONVERSANT maintenance book specific to your platform.

## Changing the Size of the Speech Filesystem

With the Veritas Advanced Filesystem, you can change the size of your speech filesystem without having to reload UnixWare. For detailed information about increasing and decreasing the size of your filesystem, see "Basic Filesystem Operations" in Chapter 5, "OA&M Menu Interface Operation," in the *Veritas Documentation*, 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," in the *Veritas Documentation*, 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 your filesystems see the section titled Basic Filesystem Operations, Underlying Speech Concepts, in Chapter 5, "OA&M Menu Interface Operation," in the *Veritas Documentation*, 585-350-906.

## Defining Phrases

A phrase is a unit of speech (for example, letter, number, word, sentence, or a paragraph) that is spoken to the 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
  - Dollars
  - Cents
  - Time
  - Weekdays
  - Months
  - Numbers

  The Enhanced Basic Speech package includes prerecorded speech formats corresponding to the above types of phrases. For a list of the speech formats that come with the Enhanced Basic Speech package, see Appendix A, ‘‘Enhanced Basic Speech Formats’’

- Custom speech

  Custom phrase tags are designed specifically for the application you are developing and are usually more than one word in length. For example

  — "Your account balance is"

  — "Please enter your five digit account number"

  — "The current interest rate is"

  Script Builder uses predefined phrase tags for spoken output, such as digits and letters in various inflections. For more information about the Script Builder predefined phrase tags, see Chapter 9, " Speech Administration," in *INTUITY*™ *CONVERSANT*® *Application Development with Script Builder*, 585-310-760.

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

## 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 your phrase when you add the phrase to your script (for example, adding a phrase through a Prompt & Collect action in your Script Builder 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 may contain up to 65,535 phrases. Your INTUITY CONVERSANT system can have up to 16,384 talkfiles, although your system is limited by the size of your 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 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 section,"Understanding the Speech Filesystem" on page 1-2 at the beginning of this chapter for more information about INTUITY CONVERSANT system filesystems.

## Overview of Speech Development Features

Several speech development features are available for the creation, editing, recognition, and inclusion of speech in your application. The following sections summarize all of the features available for speech development. The available features are as follows:

- Script Builder
- Text-to-Speech
- Graphical Designer Release 1.6
- Graphical Speech Editor
- WholeWord Speech Recognition
- FlexWord™ Speech Recognition
- Dial Pulse Recognition
- Form Filler Plus

### Script Builder

Nearly all applications involve playing recorded speech to the caller. Script Builder provides one way of producing this recorded speech. Script Builder allows you to design and develop system applications that automate most functions performed by operators or agents. For additional general information about producing speech with Script Builder, see Chapter 2, "Developing Speech"

## Text-to-Speech

Text-to-Speech (TTS) is another way of producing speech for your application. TTS takes text as input and produces synthesized speech spoken in a male voice.

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

### Uses For TTS

Uses for TTS include:

- Speaking text from a database
- Speaking text spoken in an application from a host file
- Producing text for prompts

To use TTS in your application, type the text to be spoken into your Script Builder application or your TSM script via the keyboard.

TTS is usually used to speak text that changes frequently, for example:

- Large databases of names and addresses
- Dynamic text, such as electronic mail

When using prototype applications it also eliminates the need to record custom phrases.

See “Using Text-to-Speech” of Chapter 8, “Using Optional Features,” in *INTUITY*™ *CONVERSANT*® *System Version 6.0 Application Development with Script Builder*, 585-310-760, for more information about using TTS with Script Builder. See Appendix B, “Summary of TAS Script Instructions,” in *INTUITY*™ *CONVERSANT*® *System Version 6.0 Application Development with Advanced Methods*, 585-310-761, for more information about using TTS in your TAS application.

### Requirements For TTS

The requirements for Text-to-Speech are as follows:

- Text-to-Speech software package
- AYC9 Signal Processor (SP) circuit card or
- AYC43 Speech and Signal Processor (SSP) circuit card

For the software installation procedure or instructions on how to install the AYC9 SP or the AYC43 SSP circuit card, see the INTUITY CONVERSANT maintenance book specific to your platform.

## Graphical Designer Release 1.6

The Graphical Designer feature package allows you to design applications on your personal computer. This tool allows you to develop applications by specifying every detail of the interaction between the 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 Graphical Designer. In addition, you can develop a full range of interactive voice response services including banking by telephone, processing insurance claims, paying bills, purchasing tickets, shopping by catalog, and registering for classes.

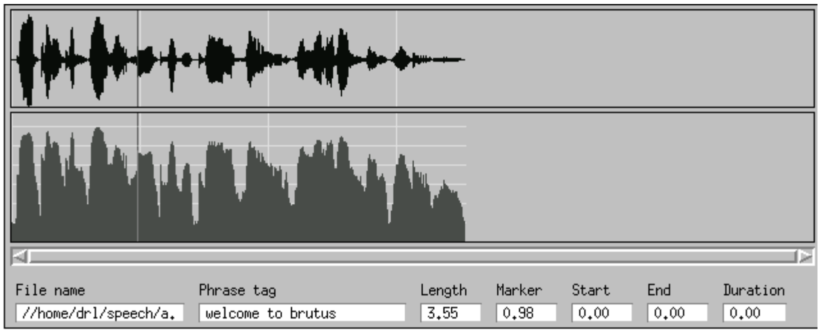
Once your application is designed, you can test, generate, transfer, and install the application.

Using the Graphical Designer feature package, you can:

- Create applications on your windows-based PC, as opposed to performing application development on your system. This eliminates the need to dedicate your system to application development.
- Develop complex applications easier and faster by creating subroutines and menus.
- Select speech phrases to play in an announcement from a directory callers set up.
- Work within a standard graphical user interface (GUI).
- Test call flow and hear what callers will hear if you have recorded speech on a multimedia personal computer.
- Use context-sensitive and on-line Help.
- Access Dial Pulse Recognition (DPR) if you have installed the optional DPR feature package on the INTUITY CONVERSANT system.
- Develop language-independent applications, as well as work with more than one language on the INTUITY CONVERSANT system.

For more information on Graphical Designer, see *INTUITY™ CONVERSANT® System Version 6.0 Application Development with Graphical Designer,* 585-310-764.

The Graphical Speech Editor (GSE) provides a graphical interface for creating and editing speech files. This user-friendly interface includes menus that you can access to perform different functions associated with retrieving, creating, editing, and saving speech files. 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 enhances the editing process because it provides the user with a visual representation of the speech file being edited. For more information about the GSE, see Chapter 3, ‘‘Editing Speech’’



| 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 For Graphical Speech Editor

The requirements for the GSE are as follows:

- Graphical Speech Editor software package
- Mouse (serial port connection COMP 2 *Only*)
- At least a 50-mHz CPU

For more information about system capacities and requirements for the Graphical Speech Editor, see “Graphical Speech Editor” in Chapter 5, “Feature Packages,*” INTUITY*™ *CONVERSANT*® *System Version 6.0 System Description*, 585-310-241. For information about recording and editing speech with the GSE, see Chapter 3, ‘‘Editing Speech’’

## 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. Speech recognition is offered as either WholeWord speech recognition or FlexWord speech recognition.

**NOTE:**
Previous system releases have offered speech technologies on a per system basis. Version 6.0 offers speech technologies on a per-channel basis. The SSP, SP, 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 only one word. For example, “calling card” can be recognized as a whole word. WholeWord speech recognition is available in the following languages:

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

The languages new to version 6.0, French, Dutch and Brazilian Portuguese are only available when using the SSP circuit card.

### Standard Versus Custom Vocabulary

Lucent Technologies provides a standard WholeWord speech recognition vocabulary that includes the digits 0–9 (and their commonly used synonyms) and 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,” for example, a survey application, or for number-intensive applications such as a banking application that requires callers to enter checking account numbers. You may choose to purchase a custom vocabulary if the standard WholeWord speech recognition vocabulary does not meet your needs.

### Requirements

The requirements for WholeWord speech recognition are as follows:

- WholeWord Speech Recognition Base package and one or two language package(s)
- A Signal Processor (SP) circuit card and Companion (CMP) circuit card or a Speech and Signal Processor (SSP) circuit card

**NOTE:**
You must use the Companion circuit card when using the SP circuit card, but you have the choice to use the SP or the SSP circuit card. You may choose to use both the SP circuit card with companion circuit card *and* the SSP circuit card.

For more information on WholeWord speech recognition, see Chapter 4, ‘‘Recognizing WholeWord Speech Input’’

## FlexWord Speech Recognition

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

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

The languages new to version 6.0, French and Japanese are only available while using the SSP circuit card.

Because FlexWord recognition provides a cost-effective way of designing large, customized vocabularies and menu options, it is ideal for word- or phrase-intensive applications. For example, a name dialer, an application which allows employees to speak the name of another employee instead of dialing a telephone extension, could be designed using FlexWord speech recognition. Since names can be built from existing phoneme models, the expense of custom whole-word data collection processes are not necessary.

### FlexWord Toolkit

The optional FlexWord Toolkit allows you to:

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

See Chapter 5, “Recognizing FlexWord Speech Input,” for detailed information on how to use the FlexWord Toolkit to create words, wordlists, and vocabularies.

### Requirements

The requirements for FlexWord speech recognition are as follows:

- FlexWord Speech Recognition Base package and one language package
- A Signal Processor (SP) circuit card and Companion (CMP) circuit card or a Speech and Signal Processor (SSP) circuit card

**NOTE:**
You must use the Companion circuit card when using the SP circuit card, but you have the choice to use the SP or the SSP circuit card. You may choose to use both the SP circuit card with companion circuit card *and* the SSP circuit card.

For more information about FlexWord Speech Recognition, see Chapter 5, “Recognizing FlexWord Speech Input.”

For the installation procedure, see the INTUITY CONVERSANT new system installation book for your specific platform. You cannot use the FlexWord Toolkit with releases prior to INTUITY CONVERSANT System Version 5.0. Also, you cannot build wordlists with the FlexWord Toolkit and move them back to a CONVERSANT System Version 4.0 system.

## 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/clicks” on the line to dial pulses. DPR is limited to the installation of systems that contain the SSP circuit card. DPR will not run on an SP circuit card. DPR supports digits “0” through “9” on analog and digital interfaces.

## Form Filler Plus

The Form Filler Plus feature provides the capability for application scripts to record callers’ responses to prompts for later transcription and review. As many as 10 responses may 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 to suit your needs, 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.

Whether you use the Form Filler application template or the action steps provided with the Form Filler Plus package to develop your own application, the Form Filler Plus package facilitates the development of voice capture and transcription scripts. Form Filler Plus application developers are able to select the coding rate that best suits their application needs and are given access to 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 will prompt the user for a mode, password, and Script ID. (Transcribe and review mode passwords may 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 will bypass 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.

For detailed information about how to use Form Filler Plus feature, see Chapter 8, “Using Optional Features with Script Builder,” *INTUITY*™ *CONVERSANT*® *System Version 6.0 Application Development with Script Builder*, 585-310-760.

# SSP Circuit Card

The speech and signal processor circuit card (SSP) a high-performance signal processor circuit card introduced in Version 6.0. It is capable of simultaneous support for various speech technologies. The SSP circuit card is the replacement for the Companion circuit card set (1SP + 2 CMPs), but the SP set is still supported. The first version of an SSP circuit card is the AYC43.

## Support

The SSP circuit card does not need a Companion circuit card interface, as current SP circuit cards do but can perform the functions of the SP and Companion circuit card set (1 SP + 2 CMPs). The SSP is able to mix all functions on the same circuit card. For example, a single SSP circuit card can run Text-to-Speech, WholeWord speech recognition, FlexWord speech recognition, voice code and play back, and Full CCA simultaneously. The ability of the SSP circuit card to do the work done by three circuit cards (1 SP + 2 CMPs) also allows for more available space in the system.

## Memory

The greater memory on the SSP eliminates several of the limitations imposed by the SP/CMP.

## Capacity

Table 1-2 lists the SSP speech channel capacities.

**Table 1-2. SSP Circuit Card Channel Capacities**

| Feature | Max. Number of Simultaneous Transactions for SSP |
|---|---|
| TTS | 60 |
| FlexWord speech recognition | 15 |
| WholeWord speech recognition without barge-in | 15 |
| WholeWord speech recognition with barge-in | 10 |
| DPR | 120 |

**NOTE:**
Channel counts assume that the entire SSP circuit card is dedicated to the specified feature. For a list of SSP circuit card channel capacities for features other than the ones listed above, see “Speech and Signal Processor Card” in Chapter 2, “Hardware,” in *INTUITY*™ *CONVERSANT*® *System Version 6.0 System Description*, 585-310-241.

## Signal Processor (SP) and Companion (CMP) Circuit Card

The signal processor (SP) circuit card is used for voice response or voice coding applications, speech recognition, call classification, etc. The SP circuit card does not connect directly to the telephone network and must be used with at least one T1 or Tip/Ring card. The Companion (CMP) circuit card is used only to support speech recognition applications.

### Support

The SP and CMP circuit card set is more limited than the SSP in the types of features it can simultaneously support. Speech recognition on the SP can be *either* WholeWord or FlexWord, but *not both*. There are two types of SP circuit cards being supported, the AYC2C and the AYC9.

### Memory

The SP and CMP circuit card set includes restrictions on which languages can be used in bilingual pairings, the size of the models, and allowing more than 10 digits of input without reducing the number of channels.

### Capacity

Table 1-3 lists the SP speech channel capacities.

**Table 1-3. SP Circuit Card Channel Capacities**

| | Max. Number of Simultaneous Transactions for SP | |
|---|---|---|
| Feature | AYC2C | AYC9 |
| Text-to-Speech[1] | Not supported | 6 |
| FlexWord speech recognition[2] | 8 | 8 |
| WholeWord speech recognition[2] | 12 | 12 |

[1] TTS is only available on the AYC9 SP circuit card.

[2] Requires two CMP circuit cards per SP circuit card.

## Companion Circuit Card

The Companion (CMP) circuit card is used only to support speech recognition applications. The CMP is connected to the SP circuit card via an SP-to-CMP interface cable and requires its own ISA slot. All of the signal interfaces come from the SP circuit card.

Version 6.0 systems support two versions of the CMP circuit card.

- AYC7 (supported in upgrades to Version 6.0)
- AYC7B (supported in upgrades to Version 6.0)

Each CMP circuit card can support a maximum of four channels of FlexWord speech recognition or six channels of WholeWord speech recognition.

**NOTE:**
A single SP circuit card can support a maximum of two CMP circuit cards.

# Developing Speech

# 2

## Overview

This chapter describes

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

## Purpose

The purpose of this chapter is to ensure that you have the knowledge 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 a tone signal (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 script are downloaded by the system from a hard disk drive to a Tip/Ring, Signal Processing (SP), or a Speech and Signal Processor (SSP) circuit card. The Tip/Ring, SP circuit card, or SSP circuit card actually plays the speech.

Processing the recorded speech so that it can be accessed by the system 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

The sections that follow discuss each of these activities.

## 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.

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/prompts are necessary. For example, you need to 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. See the following for more information:

- *INTUITY*™ *CONVERSANT® System Version 6.0 Application Design Guidelines*, 585-310-670, for information about planning your voice response application
- *INTUITY*™ *CONVERSANT® System Version 6.0 Application Development with Script Builder*, 585-310-760, for information on developing your application using Script Builder
- *INTUITY*™ *CONVERSANT® System Version 6.0 Application Development with Advanced Methods,* 585-310-761, *f*or more information on developing your application using INTUITY Response Application Programming Interface (IRAPI), the Transaction State Machine process (TSM) script level language, and/or C language

## Planning the Voice Script

After determining the transaction, you can begin 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:

- Write out every word that you expect to be spoken. Edit the voice script to change any poorly written or repetitive phrases. The voice script should read as clearly as possible because professional speakers can use the voice script to record phrases.
- Ensure that changes are written into the voice script if changes are made during recording.
- Track the contents of the voice script by using phrase numbers. Number each phrase in the written voice script.
- 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 of a sentence, in the middle of the sentence, 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 in the next section, “Writing the Voice Script.”
- Avoid a long string of adjectives. The following is a description 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 your 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.

## Placing Frame Words

Place words and/or phrases before and after the word and/or phrase that you need recorded if possible. These phrases should be familiar phrases that guide the speaker into speaking the word and/or phrase with a certain inflection. For example, if you want an accurate recording of the word "and" with medial inflection, you could, for example, 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.

## Examining Certain Speech Sounds

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 as follows:

August "eighth"

## Inspecting Voiceless Stops and Sounds

Voiceless stops are sounds like < *p* >, < *t* >, and < *k* >. Voiceless stops or sounds before and after your target word will also help you to make an accurate recording. In the example above, the final < *t* > of "August" provides a silence that makes it easy to isolate "eighth." Other voiceless sounds to use to end or begin on a frame or space are < *f* > or < *s* >.

### 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 “How 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 “101,” 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.

  Rising, medial, and falling inflections can all be found in the tables of Appendix A, ‘‘Enhanced Basic Speech Formats’’

## 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 Script Builder
- 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)

The sections that follow discuss all of the above options.

## Hiring a Professional Speaker

Hiring a professional speaker, such as an actor or an announcer, gives you recorded speech of a 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.
- Make sure 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 alpha and numeric characters that are to be recorded with rising, medial, and falling inflections are spoken with the appropriate inflections.
- Use the same speaker for all speech associated with a specific application.

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

See the section, ''Recording Speech'' on page 2-13 below for information on setting up a recording session with a professional speaker.

## Producing Self-Recorded Custom Speech

Script Builder allows you to record speech yourself and store it on the hard disk drive. 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 in Script Builder to record your own speech for phrases unique to your application.

**NOTE:**
The quality of speech recorded through Script Builder is not as high as the quality produces from professionally recorded speech.

For detailed information about producing speech with Script Builder, see Chapter 9, “Speech Administration,” of *INTUITY*™ *CONVERSANT® System 6.0 Application Development with Script Builder*, 585-310-760.

## Purchasing the Enhanced Basic Speech Package

You can purchase the professionally recorded Enhanced Basic Speech Package from Lucent Technologies. The Enhanced Basic Speech Package 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 enhanced basic speech phrases available from Lucent Technologies, see Appendix A, ‘‘Enhanced Basic Speech Formats’’

**NOTE:**
Contact your Lucent Technologies representative if you are interested in purchasing the Enhanced Basic Speech Package.

## 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 representative if you are interested in purchasing a custom speech package.

## Using Text-to-Speech

Text-to-Speech (TTS) is an advanced option that eliminates the need for recording speech. You enter the phrases to be spoken, and TTS synthesizes the speech.

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

Basically, 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 what 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.” could 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 INTUITY CONVERSANT Script Builder menu interface as well as with a TSM script instruction. The talk-off function and other system features for voice response work with TTS as they work with other speech files. TTS also allows an application developer to use both prerecorded phrases and TTS 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 D, ‘‘Advanced Text-to-Speech Features’’ for examples of each).

See *INTUITY™ CONVERSANT® System Version 6.0 Application Development with Script Builder*, 585-310-760, for more information on using Text-to-Speech in your Script Builder application.

For more information on using Text-to-Speech in your TSM application or in your IRAPI application see *INTUITY™ CONVERSANT® System Version 6.0 Application Development with Advanced Methods, 585*-310-761.

## Using the Graphical Speech Editor

The 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 created with Script Builder. New phrases can be added, changed, or deleted to these prerecorded files.

## 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

For more information on how to share speech in your Script Builder application, see Chapter 9, "Speech Administration," in *INTUITY*™ *CONVERSANT® System Version 6.0 Application Development with Script Builder*, 585-310-760.

## Importing Speech

You can import speech phrases from other applications and edit them with the GSE or Script Builder for use in your application. For example, suppose the following phrase was already recorded for another application:

> "Thank you, please call Bank America again."

You could import the phrase and edit it using GSE to say "Thank you, please call again," and include it in your application. If you just wanted the "thank you" part of the phrase, you could 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 your speech phrase.

Another example of a phrase already recorded for another application is as follows:

> "For account balance, press 1."

You could import the phrase and edit it using Script Builder to say "Press 1."

**NOTE:**
With importing, two copies of the speech exist on the hard disk.

For information on how to import speech into your Script Builder application, see Chapter 10, "Speech Administration," in *INTUITY*™ *CONVERSANT® System Version 6.0 Application Development with Script Builder*, 585-310-760. For more information about editing speech, see Chapter 3, ‘‘Editing Speech’’

# Recording Speech

The following speech development methods involve you recording speech or having the speech recorded for you:

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

## Using a Professional Speaker

This section discusses what is 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 and Recording Parameters

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.

### Equipment Specifications

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 the sections earlier in this chapter called ‘‘Planning the Voice Script’’ and ‘‘Writing the Voice Script’’ 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 preceding 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 alpha characters are recorded as separate phrases). See “Recording Alpha and Numeric Characters with Inflections” later in this chapter for tips to record alpha and numberic characters.

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

> “The Dow was at” “eighteen” “forty” “three,” “up” “7”
> “at the close of trading.”

This entire sentence is recorded, but the sentence can be encoded as seven separate speech files that are concatenated later. The seven phrases which are encoded separately are shown in quotation marks (“ ”).

### 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 Lucent Technologies 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 ‘‘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. See Chapter 3, ‘‘Editing Speech’’ for more information on editing speech.

### Recording Alpha and Numeric Characters with Inflections

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

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

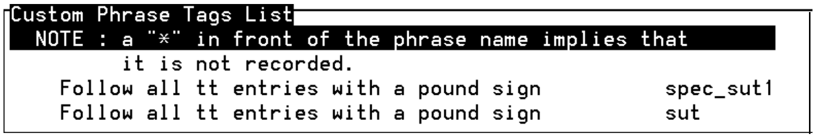
**NOTE:**
Remember that the Lucent Technologies Enhanced Basic Speech Package includes letters and numbers. See Chapter A, ‘‘Enhanced Basic Speech Formats’’ for a complete list of all phrases included.

### 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

  The easiest way to do this is to use the Script Builder F5 (LIST) function to generate a paper copy of all phrase tags in the application (unrecorded phrases are marked with an asterisk). See Figure 2-1.



```
Custom Phrase Tags List
  NOTE : a "*" in front of the phrase name implies that
         it is not recorded.
    Follow all tt entries with a pound sign          spec_sut1
    Follow all tt entries with a pound sign          sut
```

**Figure 2-1. Listing Custom Phrase Tags in Script Builder**

- A list of the complete text of all the phrases that need to be recorded

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

- A clear indication of the phrase tags that corresponds to each phrase

- Specifications of what type of speech coding should be used

  Script Builder uses Adaptive Delta Pulse Code Modulation (ADPCM), Code Excited Linear Prediction (CELP), and Sub Band Coding (SBC).

- Specifications of whether the phrases are to be recorded using a male or female voice

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

### Producing Self-Recorded Custom Speech with Script Builder

You can record speech directly into the system using Script Builder. See Chapter 9, “Speech Administration,” of *INTUITY™ CONVERSANT® System Version 6.0 Application Development with Script Builder*, 585-310-760, for instructions on recording speech in Script Builder.

## Digitizing Speech Phrases

Recorded speech phrases are input from a reel-to-reel tape recorder, amplifier and microphone, or from a person over a telephone line. 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:

- With the Graphical Speech Editor

  If you record speech using the GSE, it is digitized automatically. See “Using the Graphical Speech Editor” for information about recording speech with the GSE

- With Script Builder

  If you have recorded the speech using Script Builder, it is digitized automatically. For more information about Script Builder’s speech administration capabilities, see Chapter 9, “Speech Administration,” of *INTUITY™ CONVERSANT® System Version 6.0 Application Development with Script Builder,* 585-310-760.

- 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, you need to install it onto your system.

## Installing Speech in a Script Builder or TSM Application

If your application has been created in Script Builder, see “Restoring Speech” in Chapter 9, “Speech Administration,” in *INTUITY*™ *CONVERSANT*® *System Version 6.0 Application Development with Script Builder*, 585-310-760, for specific information on how to install speech onto your system.

## Installing Speech in an IRAPI Application

If your application has been created using the C language, perform the following procedure to install speech:

1. Insert the floppy disk into the drive.
2. Enter the appropriate speech filesystem. For example, if you are using the default speech filesystem, enter the following:

   **cd /voice1/vfs/talkfiles**

3. Enter **mkdir *<talkfile #>***

   Make sure that you select a talkfile number not already in use.

   For example, if you are using talkfile100, you would enter the following:

   **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 6.0 Application Development with Advanced Methods*, 585-310-761.

# Editing Speech

# 3

## Overview

This chapter describes

- Using the Graphical Speech Editor (GSE) to edit speech files
- Using Script Builder to edit speech files

## Purpose

The purpose of this chapter is to ensure that you are proficient in using Graphical Speech Editor (GSE) and Script Builder to edit speech files.

## Introduction to the Graphical Speech Editor

The Graphical Speech Editor (GSE) is an X-Windows based graphical user interface (GUI) that can be accessed and operated with a three-button mouse and a keyboard.

The GSE is able to 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 standard 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.

### Making Changes with GSE

GSE provides a simple way of making changes to existing speech phrases by allowing you to cut, copy, and paste speech segments within a speech file or across multiple speech files. It also provides the ability to change the volume of individual speech segments.

### Supported Speech Formats

The GSE package supports the editing and manipulation of speech recorded in 16- or 32-kbps Adaptive Differential Pulse Code Modulation (ADPCM) only.

The GSE provides the ability to convert user-selected ADPCM formats. The GSE buffer is capable of handling up to 4 minutes of speech, whether the speech consists of one phrase or a series of phrases.

### Working with Segments of Speech

The GSE feature allows you to display a specific segment of speech that appears as a graphical waveform on your system monitor. You mark an area by moving your 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.

# Accessing the Graphical Speech Editor

The GSE user interface includes windows that can be accessed to perform different functions associated with the retrieving, creating, editing, and saving of speech files. This section also describes installing and administering the mouse, mouse operations, and how to use the keyboard to perform editing functions.

## Installing and Administering the Mouse

For instructions on installing the mouse, see the INTUITY CONVERSANT new system installation book specific to your platform. For instructions on administering the mouse, see *System Setup and Configuration* in the Novell UnixWare Documentation Set, 585-350-908.

⚠ **CAUTION:**

When referring to the section "Adding a Mouse Device" Chapter 1, "Setting Up the Work Environment," in *System Setup and Configuration* of the Novell UnixWare Documentation Set, 585-350-908, note the following:

In order to test the changes to the mouse configuration that you have just added, you need to select **U** (Update Mouse Configuration and Quit) and then rerun **mouseadmin** before you can test the mouse configuration by selecting **T** (Test your mouse configuration). The procedure is shown in the reverse order in Chapter 1, "Setting Up the Work Environment."

## Installing the Graphical Speech Editor

The procedure for installing the GSE is described in the INTUITY CONVERSANT maintenance book specific to your platform. See the INTUITY CONVERSANT new system installation book specific to your platform for information about GSE hardware configurations.

**NOTE:**

The INTUITY CONVERSANT system must be stopped while using the GSE. Entering the **gse** command results in a system prompt asking you if it is okay to stop the system.

## Setting Up Graphical Speech Editor Software

For a detailed description of the setup for the GSE and how to determine the correct video card, see "Setting Up the Monitor" in Chapter 9, "Installing Base System Software," of the INTUITY CONVERSANT maintenance book specific to your platform.

## Initiating the Graphical Speech Editor

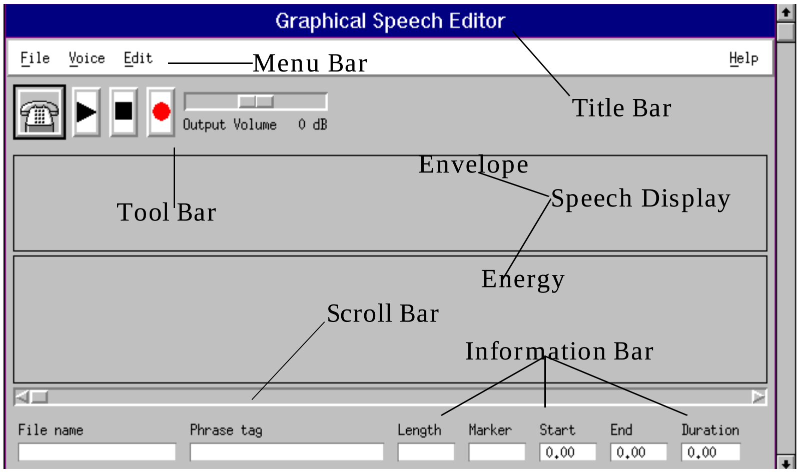
To initiate the GSE, enter:

```
gse [-l<chan#>] [-p<playchan#>] [-r<recchan#>]
```

where:

***-l*** = listen channel (input and output)
***-p*** = play channel (output only)
***-r*** = record channel (input only)

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



File name
Phrase tag
Length
Marker
Start 0.00
End 0.00
Duration 0.00

**Figure 3-1. GSE Screen**

The designated channels must be configured for both input and output; this facilitates recording and playing speech phrases. The ***-l*** channel is used for telephone lines and incorporates both audio input and output. Alternatively, the ***-p*** and ***-r*** channels must be used together to establish the 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. See Appendix C, “Computing GSE Channel Numbers,” for an explanation of how to determine channel numbers for the l, p, and r channels. For more information about the **gse** command, see *INTUITY*™ *CONVERSANT*® *Administration System Version 6.0*, 585-310-591.

# Graphical Speech Editor User Interface

This section describes the layout and operation of the GSE user interface. The GSE user interface enables you to record and edit speech files for INTUITY CONVERSANT applications. The following components comprise the GSE user interface:

- Title bar
- Menu bar
- Tool bar
- Speech display
  - Envelope
  - Energy
- Scroll bar
- Information bar

Refer to Figure 3-1 which illustrates all of these components.

## Title Bar

The title bar (Figure 3-2) expands across the top of the GSE screen to identify the graphical user interface feature of the INTUITY CONVERSANT system.

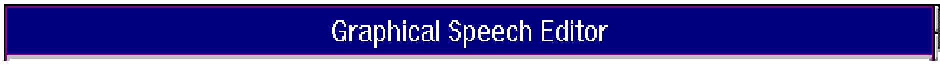


**Figure 3-2. Title Bar**

## Menu Bar

The menu bar (Figure 3-3) is comprised of the following subfunctions:

- File
- Voice
- Edit
- Help

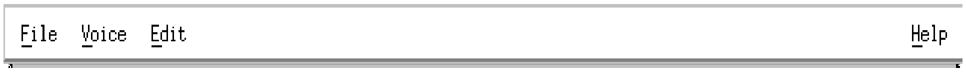


**Figure 3-3. Menu Bar**

### File Menu

The File menu offers the following choices:

- Open

  To access an existing speech file, select `Open` from the `File` menu.

  **NOTE:**
  You must first create and save new files before you can open them.

- Save

  After you have edited a speech file, save your changes by selecting `Save` from the `File` menu.

- Save As

  You must name and save new speech files. To do this, select `Save As` from the `File` menu.

- Exit (Ctrl+Q)

  Select `Exit` to close the GSE and return you to the system prompt. You can also exit the editor at any time by pressing Ctrl Q .

### Voice Menu

The Voice menu offers the following choices:

- Hook

  The telephone receiver (handset) can either be `on-hook` or `off-hook`. To play or record speech, the receiver must be `off-hook`.

  When you select this item from the menu, a submenu is displayed from which you can identify `on-hook` or `off-hook` status.

- Play

  After a speech file has been opened and the telephone receiver is off-hook, select `Play` from the `Voice` menu to hear the message contained in the speech file.

- Stop

  Select `Stop` to end a recording session or to discontinue a message playback.

- Record

  Select `Record` to start a recording session. Typically, you would select `Record` after the telephone receiver is `off-hook`.

**NOTE:**
You can also select all Voice menu items from their corresponding Tool menu icons.

### Edit Menu

The Edit menu offers the following choices:

- Cut

  This editing function enables you to remove a selected segment of speech from the speech file. You may want to cut a segment of speech either to delete it entirely from the speech file, or to paste it at a different location.

- Copy

  You can copy segments of speech into a buffer for pasting at different locations in the speech file.  You must select the segment before you can implement the Copy function.

- Paste

  You can paste segments of speech that have either been cut or copied into the buffer anywhere in the speech file. The selected segment of speech is pasted at the present location of the marker.

  Use the middle-button of the mouse to move the marker across the speech display.

- Volume

  Select this editing function to set the volume at the level the speech file is to be heard by the caller.

- Undo

  While editing a speech file, use this function to undo the last change made.  The Undo function applies to all the editing features offered by the GSE.

**NOTE:**
Use either the mouse or the corresponding key sequences found under the edit pull-down menu to perform any of the above listed editing functions.

### Help Menu

The Help menu is reserved for future use.

## Tool Bar

The tool bar (Figure 3-4) is comprised of five icons that you can use to perform the basic editing functions offered by GSE. Each of these icons represents a function that you can also perform via appropriate selections on the menu bar. These functions are described below.

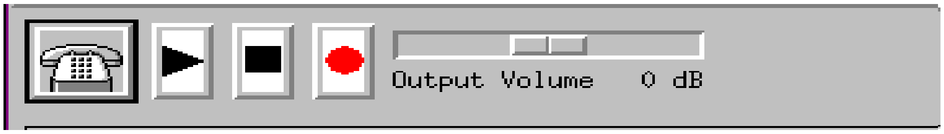


**Figure 3-4. Tool Bar**

- Telephone Icon

  This icon reflects the *on-hook* or *off-hook* state of the telephone handset. To place the telephone in an off-hook state (Figure 3-5), click on the telephone icon.

  **NOTE:**
  The telephone handset must be off-hook prior to recording speech or listening to a playback via the Play function.

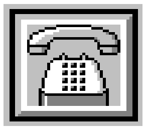


**Figure 3-5. Telephone Icon Showing the Off-Hook State**

- Play Icon

  Activate the play icon (Figure 3-6) to listen to a playback of recorded speech.

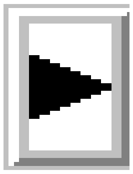


**Figure 3-6. Play Icon**

- Stop Icon

  Activate the stop icon (Figure 3-7) to stop recording or playing speech.

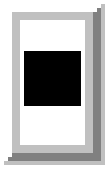


**Figure 3-7. Stop Icon**

- Record Icon

  Activate the record icon (Figure 3-8) to start recording speech.

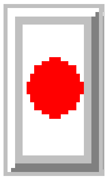


**Figure 3-8. Record Icon**

- Output Volume Icon

  To adjust the volume at which a speech file is played, drag the volume button (Figure 3-9) in either direction with the mouse. The increased or decreased output level, measured in decibels, is reflected below the icon.

  **NOTE:**
  Changing the volume in this manner only affects the speech being played during the present GSE session. To permanently change the volume, you need to change it through the Edit menu.

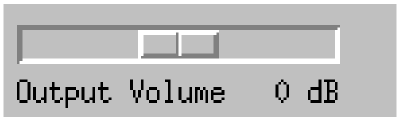


**Figure 3-9. Volume Icon**

## Speech Display Area

The speech display area represents a graphical image of speech. Approximately 7 seconds of speech, from a speech file that can be up to 4 minutes long, are displayed at a time.

Two windows concurrently display separate graphical representations of the same digitized speech file (Figure 3-10). The envelope and energy representations of the audio signal are essentially the same. Their difference lies in their respective interpretative values to you.

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 3-10) are graphical representations of an audio signal oscillating between the positive and negative extremes.

Literally translated, energy is the output level of sound behind every phonetic utterance of speech. This energy is vertically measured in 10-dB increments and graphically plotted on the display. The energy window displays the amount of energy in the audio signal.

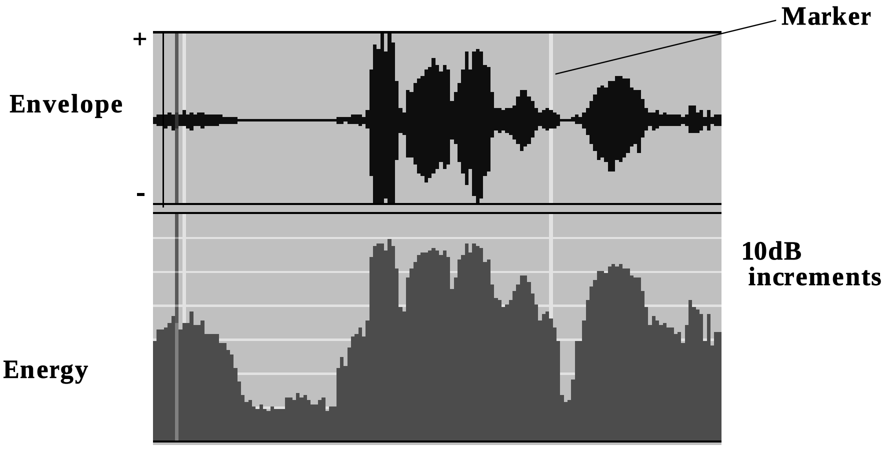


**Figure 3-10. Graphical Speech Editor Speech Display**

## Scroll Bar

The scroll bar (Figure 3-11) enables you to pace through the contents of the speech file, by clicking on the end arrows in either direction. Using the middle button on the mouse accelerates this process.

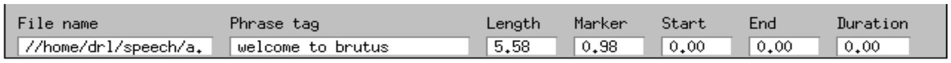
A rectangular column within the scroll bar identifies the location of the displayed segment in relation to the total contents of the speech file. The width of this column reflects the proportion of the speech file being displayed.

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

**Figure 3-11. Scroll Bar**

## Information Bar

The information bar (Figure 3-12) is located at the bottom of the screen. It provides information pertinent to GSE operations, such as the speech file name, phrase tag, length of speech, etc.



| 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 3-12. Information Bar**

The information bar is comprised of the following windows:

- File Name Window

  The File name window identifies the name of the speech file.

- Phrase Tag Window

  The Phrase tag window is used for applications developed on Script Builder. The phrase tag is for identification purposes only. See Chapter 1, "Overview of Speech on the INTUITY CONVERSANT System," for information about phrase tags.

- Length Window

  The Length window identifies the entire length of the speech file in seconds.

- Marker Window

  The Marker window identifies the position of the marker (vertical cursor) while speech is measured in seconds from the start of the file.

- Start Window

  The Start window 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 Window

  The End window 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 Window

  The Duration window identifies the duration of the selected segment, or the difference between the start and end limits of the segment in seconds.

# Editing a Speech File with GSE

This section provides a detailed description of how to use the different editing functions provided with the GSE. The following issues are discussed:

- 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

This section summarizes how to use the mouse.

- Use the left button for the following:
  - Selecting menu items
  - Selecting a segment of speech
  - Modifying the length of the selected segment
  - Moving horizontally in either direction with the scroll bar
  - Moving vertically up or down with the scroll bar (opening files)
- Use the middle button for:
  - Moving the marker
  - Setting a position on the scroll bar

**NOTE:**
Functions of the right button are reserved for future use.

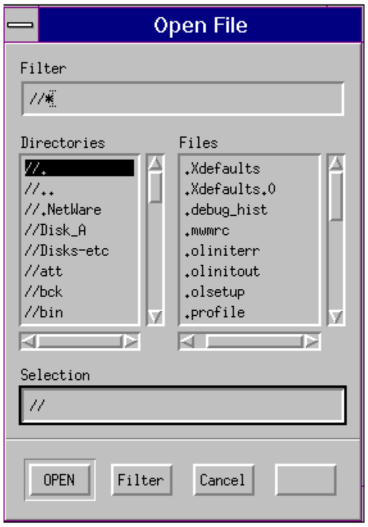
## Opening a File

Before you can edit an existing speech file, you must open it or bring it into the GSE.

To open a file, perform the following procedure:

1. From the GSE screen (Figure 3-1) select `File` from the menu bar.
2. Click on `Open` with the left button on your mouse.

   The system displays the Open File screen (Figure 3-13).



Filter: //*

Directories:

```
//.
//..
//.NetWare
//Disk_A
//Disks-etc
//att
//bck
//bin
```

Files:

```
.Xdefaults
.Xdefaults.0
.debug_hist
.mwmrc
.oliniterr
.olinitout
.olsetup
.profile
```

Selection: //

**Figure 3-13. Open File Screen**

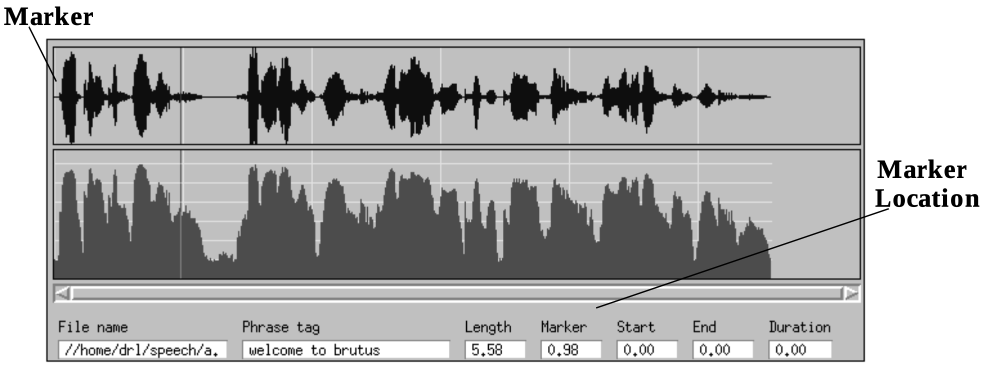
This window provides you with scrolling tools that facilitate the identification of files within a specific directory. In addition, the Filter button expands the identification of a path, or pattern, from which a particular speech file can be retrieved.

3. Select the Open button to open the file or press Cancel to return to the speech-editing screen.

## 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 screen (Figure 3-14).

To move the marker, press the middle button on the mouse. After moving the cursor to the desired location, release the mouse button to position the marker at a specific 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 |

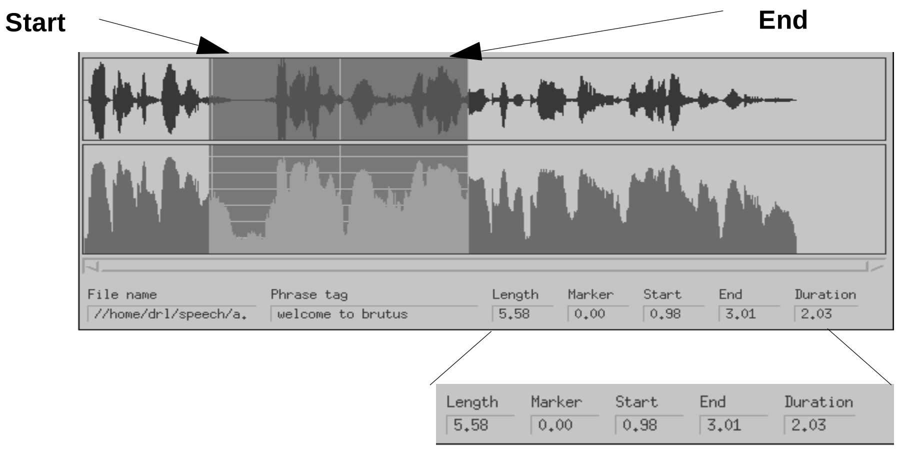
**Figure 3-14. Speech Marker**

## Selecting a Segment of Speech

To select a segment of speech, perform 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 then highlighted within the identified starting and ending boundaries (Figure 3-15).



| 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 3-15. Selecting Segments of Speech**

If you look at the information bar at the bottom of the screen, you will notice the Start and End delimiters of the selected segment presented in seconds (Figure 3-15). The Duration window identifies the total time that the selected segment of speech stretches across.

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

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

## Listening to Speech

To listen to a segment of speech, perform the following procedure:

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 on the [Open] button.

## Cutting and Pasting Speech

To remove a selected segment of speech from the speech file, you can use the Cut function. 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.

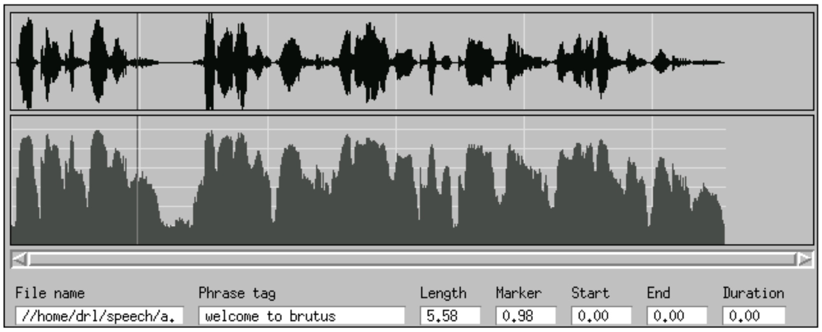
**NOTE:**
You must use the Cut function before the Paste function.

To remove a segment of speech from a speech file and then paste the speech into a different part of the file, perform the following procedure:

1. Begin from the GSE screen.
2. 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.
3. Select `Edit` from the menu bar.
4. Select `Cut` or press Ctrl T.

   The system removes the selected segment of speech from the file. In Figure 3-16, the segment of speech selected in Figure 3-15 is now removed from the speech file.



| 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 3-16. Cutting Segments of Speech**

5. Move the marker to the destination area by using the middle button of the mouse.

6. Select `Edit` from the menu bar.
7. 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.

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

To copy a segment of speech, perform the following procedure:

1. Begin from the GSE screen.

   The system displays the various segments of speech available.
2. Select the segment of speech to be copied.
3. Select `Copy` or press Ctrl C .

   The system copies the selected segment of speech into the buffer. You can now paste the contents of the buffer into different parts of the speech file.
4. Move the marker to the destination area by using the middle button of the mouse.
5. Select `Edit` from the menu bar.
6. 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.

## Adjusting the Volume

Use the Volume function to adjust the volume at which a caller hears a speech file.

To adjust the volume of speech, perform the following procedure:

1. Begin from the GSE screen.
2. Select the segment of speech.
3. Select `Edit` from the menu bar.
4. Click on `Volume` (Figure 3-17), or press Ctrl V.
5. 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.
6. Click on OK to save the volume setting.
7. Select Cancel to return to the speech-editing screen.

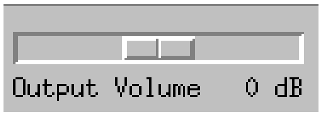


**Figure 3-17. Modifying the Speech Volume**

## Undoing Changes

You can undo the last change you made to a speech file when performing any of the preceding functions.

To undo the last change, perform the following procedure:

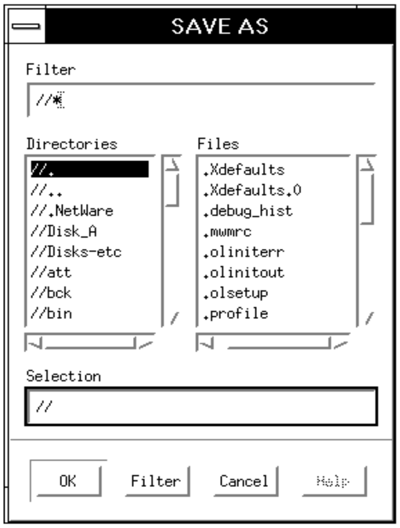
1. Begin from the GSE screen.
2. Select `Edit` from the menu bar.
3. Click on the `Undo` function with the mouse or press Ctrl U.

## Saving Changes

To name and save new speech files, after you have edited a speech file, perform the following procedure:

1. Begin from the GSE screen.
2. Select `File` from the menu bar.
3. Select `Save As`.

   The system displays the Save As screen (Figure 3-18). This screen provides scrolling tools that facilitate the naming of files within a specific directory. Filter enables identification of a path or pattern to which a particular speech file is to be saved.
4. Click OK to complete the save, or click Cancel to return to the speech-editing screen without saving the changes.



Filter: //*

Directories:
//.
//..
//.NetWare
//Disk_A
//Disks-etc
//att
//bck
//bin

Files:
.Xdefaults
.Xdefaults.0
.debug_hist
.mwmrc
.oliniterr
.olinitout
.olsetup
.profile

Selection: //

**Figure 3-18. Save As Screen**

## Exiting

Select `Exit` from the File menu to close the GSE and return you to the system prompt. You can also exit the editor at any time by pressing Ctrl Q. Once you exit the Graphical Speech Editor, click and hold down the left mouse button and press Ctrl to exit windows.

**NOTE:**
If you have been working on a speech file and choose to exit prior to saving the file, a window with a message appears.

## Converting Speech Files

This section describes the utilities provided with INTUITY CONVERSANT system to integrate the GSE into the application development environment. These integration utilities are designed to perform the following functions:

- 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 Script Builder 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 UNIX files which 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 UNIX files in the GSE format.

For single phrases, the **gse_copy** command, copies the phrase (identified by a *talkfile* and *phrase number*) into a UnixWare file. Conversely, the **gse_add** command adds (restores) the phrase to *the talkfile* (identified by the same *talkfile* and *phrase* number).

The **gse_copypl** command uses the **gse_copy** routine to extract individual phrases. Similarly, the **gse_addpl** command uses the **gse_add** routine to restore individual phrases.

The following sections detail the process of retrieving speech from speech pools, converting it to a form that can be edited by the GSE, and reversing the entire process.

## Retrieving Files from Speech Pools

This section includes information on how to extract speech files from speech pools and prepare the files for editing by the GSE. The system must be running to extract speech files for editing.

### Procedure

To retrieve speech files from a speech pool and prepare the files for editing with GSE, enter the following command:

**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, phrase numbers, and output file names for the phrases to be extracted. To determine which speech pools an application uses, examine the Shared Speech pools parameter of the Script Builder application. See “Defining Shared Speech” in Chapter 6, “Defining Parameters,” in *INTUITY™ CONVERSANT® Application Development with Script Builder*, 585-310-760, for more information about the Shared Speech Parameter in Script Builder.

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

You must first create a directory for talk3.files by typing **cd /speech/talk**
Then enter **mkdir talk3.files**

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 3-1). 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, typing the command **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.

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

**Table 3-1. 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` |
| `f1000` | `1002` | `This is the third Phrase Tag` |

## Restoring Files to Speech Pools

This section includes information on how to restore speech files and then add then to a speech pool.

### Procedure

To restore speech files to their original format and add them to the appropriate speech pool, enter the following command:

**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 name of the directory where the GSE edited files are located; *<codetype>* is either "pcm64," "adpcm16," or "adpcm32;" and, *<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, phrase numbers, and file names of the phrases to be added. To determine which speech pools an application uses examine the Shared Speech Pools parameter of the Script Builder application. 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.

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*®

*Application Development with Script Builder*, 585-310-760, for more information about the Shared Speech Parameter in Script Builder.

### Example

In this example, a phrase listfile named **talk3.pl** is 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 the command

```
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 the command

```
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

This section includes information on how to retrieve speech files from the system speech filesystem and how to prepare the files for editing by the GSE. The system must be running to extract speech files for editing.

### Procedure

To retrieve speech files from the speech filesystem and prepare the files for editing with GSE, enter the following command:

**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.

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 and phrase numbers of the phrase to be edited.

If you do not know these numbers, you may be able to determine them with the following commands:

- When no information is available, enter **list talkfile**

  This command provides information on all phrases stored in the speech filesystem sorted by talkfile number. The needed talkfile and phrase numbers can be determined by searching for the name of the phrase (that is, the phrase tag).

- If you know the talkfile number, enter the following command for a list of all phrases in the given talkfile: **list talkfile *<tnum>*** where *<tnum>* is the talkfile number.
- If you know the phrase number, but not the talkfile number, enter the following command for a list of all talkfiles in which the given phrase number is used: **list phrase *<phnum>*** where *<phnum>* is the phrase number.

When the talkfile and phrase numbers are known, use the **gse_copy** to retrieve the phrase for editing.

The **gse_copy** command is implemented as a UNIX shell program that uses existing commands to perform all the needed tasks.

**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 *<file>* be the same as the phrase number and that the directory containing *<file>* be named the same as the talkfile number.

### Example

Entering the command

```
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

To restore speech files to their original speech file, enter the following command:

**gse_add *<talkfile> <phrase> <codetype> <inputfile>***

where *<talkfile>* is the talkfile number; *<phrase>* is the phrase number; *<codetype>* is either “pcm64,” “adpcm16,” or “adpcm32;” and *<inputfile>* is the input file.

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 may determine them by using the **list** command as described previously in the above procedure section.

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

#### Example

Entering the command **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

You can edit speech with Script Builder by trimming off pieces of the recording from one or both ends, and playing the edited recording. Continue playing, trimming and replaying until your speech word or phrase is edited as you desire. Restoring already trimmed speech is possible during any given editing session, if too much is trimmed. For detailed information about editing speech files with Script Builder, see “Editing Speech” in Chapter 9, “Speech Administration,” in *INTUITY™ CONVERSANT® System Version 6.0 Application Development with Script Builder*, 585-310-760.

# Recognizing WholeWord Speech Input

# 4

## Overview

This chapter describes what is involved in recognizing WholeWord speech input.

## Purpose

The purpose of this chapter is to ensure that you are able to use and comprehend WholeWord speech recognition components and 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 the vocabulary “zero” through “nine” and their commonly used synonyms, and the words “yes” and “no.”

## Components of WholeWord Speech Recognition

The following sections describe the components included with the WholeWord speech recognition package.

### Global Support

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

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

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

## Bilingual WholeWord Speech Recognition Capabilities

You can load and operate two WholeWord speech recognition language packages on a single system simultaneously with the correct hardware.

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.

If you are using an SP circuit card, due to size restrictions, you can run only *one* of the following languages at a time:

- German
- Japanese
- UK English

Also when using an SP circuit card, you can run *two* of the following languages simultaneously:

- Australian English
- Canadian French
- Castilian Spanish
- Latin-American Spanish
- US English

If you are using an SSP circuit card, you can run *any two* WholeWord speech recognition packages simultaneously.

**NOTE:**
You *must* use an SSP circuit card to run French, Dutch, or Brazilian Portuguese.

## Standard Vocabulary

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

**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 |
|---|---|---|---|---|---|---|---|
| 0 | zero, oh | zéro | cero | zero | null | nul | zero |
| 1 | one | un, une | uno | ichi | eins | een | um |
| 2 | two | deux | dos | ni | zwei, zwo | twee | dois |
| 3 | three | trois | tres | san | drei | drie | três |
| 4 | four | quatre | cuatro | shi, yon | vier | vier | quatro |
| 5 | five | cinq | cinco | go | fünf | vijf | cinco |
| 6 | six | six | seis | roku | sechs | zes | meia, seis |
| 7 | seven | sept | siete | shichi, nana | sieben | zeven | sete |
| 8 | eight | huit | ocho | hachi | acht | acht | oito |
| 9 | nine | neuf | nueve | ku, kyu | neun | negen | nove |
| yes | yes | oui | sí | hai | ja | ja | sim |
| no | no | non | no | iie | nein | nee | não |

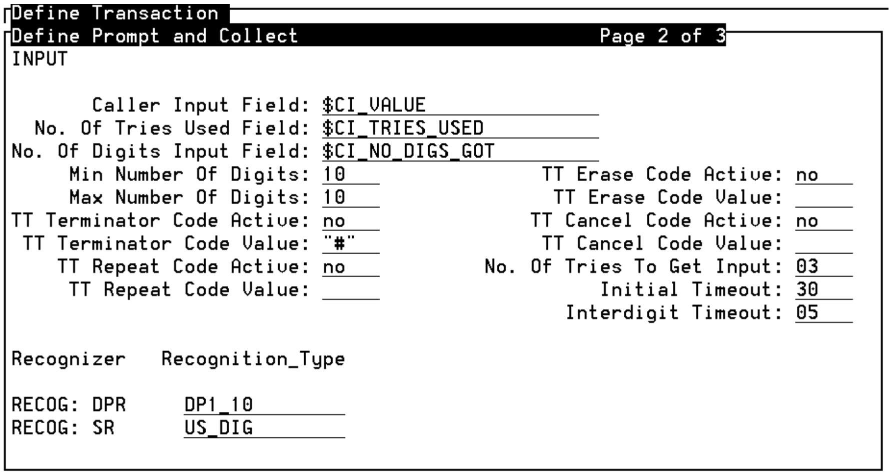
## Standard WholeWord Speech Recognition Types

Recognition types allow you to indicate which key word or words are possible selections for each prompt. The recognition type determines which models the incoming speech is compared. *Keyword* recognition types are used for isolated word recognition, that is, responses of a single word or digit.

A complete recognition type includes mathematical models of the words, a “grammar,” specifications of maximum and minimum numbers of digits, and in some cases, a DIP. All of the recognition types and words used by an application constitute that application’s “vocabulary.” The standard WholeWord speech recognition package provides several commonly used recognition types.

The recognition type is selected in the Recog*:* field on Page 2 of the Define Prompt & Collect screen in Script Builder (Figure 4-1). Tables 4-2 through 4-4 show 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’.”

*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 4-5 lists the connected-word recognition types.



```
Define Transaction
Define Prompt and Collect                            Page 2 of 3
INPUT

       Caller Input Field: $CI_VALUE
    No. Of Tries Used Field: $CI_TRIES_USED
  No. Of Digits Input Field: $CI_NO_DIGS_GOT
      Min Number Of Digits: 10             TT Erase Code Active: no
      Max Number Of Digits: 10              TT Erase Code Value:
  TT Terminator Code Active: no            TT Cancel Code Active: no
   TT Terminator Code Value: "#"            TT Cancel Code Value:
      TT Repeat Code Active: no        No. Of Tries To Get Input: 03
       TT Repeat Code Value:                     Initial Timeout: 30
                                             Interdigit Timeout: 05

Recognizer   Recognition_Type

RECOG: DPR     DP1_10
RECOG: SR      US_DIG
```

Figure 4-1. Define Prompt and Collect Screen

**Table 4-2. Standard WholeWord Speech Recognition Types For US English, Latin-American Spanish, and Castilian Spanish**

| Digit Lengths | US English | Latin-American Spanish | Castilian Spanish |
|---|---|---|---|
| “Yes” or “No” | US_YN | MS_YN | CS_YN |
| One character 1_3 | US_1_3 | MS_1_3 | CS_1_3 |
| One character 1_3 or “no” | US_1_3N | MS_1_3N | CS_1_3N |
| One character 1_5 | US_1_5 | MS_1_5 | CS_1_5 |
| One character 1_5 or “no” | US_1_5N | MS_1_5N | CS_1_5N |
| One character 0_9 or “oh” | US_DIG | MS_DIG | CS_DIG |

**Table 4-3. Standard WholeWord Speech Recognition Types For Dutch, German, Brazilian Portuguese, and French**

| Digit Lengths | Dutch | German | Brazilian Portuguese | French |
|---|---|---|---|---|
| “Yes” or “No” | NL_YN | GR_YN | BP_YN | FR_YN |
| One character 1_3 | NL_1_3 | GR_1_3 | BP_1_3 | FR_1_3 |
| One character 1_3 or “no” | NL_1_3N | GR_1_3N | BP_1_3N | FR_1_3N |
| One character 1_5 | NL_1_5 | GR_1_5 | BP_1_5 | FR_1_5 |
| One character 1_5 or “no” | NL_1_5N | GR_1_5N | BP_1_5N | FR_1_5N |
| One character 0_9 or “oh” | NL_DIG | GR_DIG | BP_DIG | FR_DIG |

Table 4-4. Standard WholeWord Speech Recognition For Japanese, Australian English, UK English, and Canadian French

| Digit Lengths | Japanese | Australian English | UK English | Canadian French |
|---|---|---|---|---|
| "Yes" or "No" | JN_YN | AU_YN | UK_YN | CF_YN |
| One character 1_3 | JN_1_3 | AU_1_3 | UK_1_3 | CF_1_3 |
| One character 1_3 or "no" | JN_1_3N | AU_1_3N | UK_1_3N | CF_1_3N |
| One character 1_5 | JN_1_5 | AU_1_5 | UK_1_5 | CF_1_5 |
| One character 1_5 or "no" | JN_1_5N | AU_1_5N | UK_1_5N | CF_1_5N |
| One character 0_9 or "oh" | JN_DIG | AU_DIG | UK_DIG | CF_DIG |

Table 4-5. WholeWord Speech Connected-Digit Recognition Types

| Language | Fixed Lengths of 1 to 10 digits or Variable Lengths digits of 0-9 digits (US English only) |
|---|---|
| French | FR_DIG |
| Canadian French | CF_DIG |
| US English | US_DIG |
| Latin-American Spanish | MS_DIG |
| Castilian Spanish | CS_DIG |
| Dutch | NL_DIG |
| German | GR_DIG |
| Brazilian Portuguese | BP_DIG |
| Japanese | JN_DIG |
| Australian English | AU_DIG |
| UK English | UK_DIG |

**NOTE:**
Languages other than US English do not allow strings of greater than 10 digits. With US English, we recommend the use of a custom grammar.

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. The string length and grammar type is specified in the `Min Number of Digits` and `Max Number of Digits` fields on Page 2 of the Define Prompt & Collect screen (Figure 4-1). See Chapter 7, "Defining the Transaction," in *INTUITY™ CONVERSANT® System Version 6.0 Application Development with Script Builder*, 585-310-760, for more information about the Prompt & Collect action step.

**NOTE:**
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.

## Models

Each word of the system's vocabulary is represented by one or more mathematical *models* which contain the speech signal characteristics of the word. The model is created from thousands of samples of the spoken word.

## 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 your speech recognition package share the same set of models for words that are common to their package's vocabulary.

## 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) or variable-length (1–24) for US English *only*. 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 4-5). 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. A variable-length string indicates a string of digits that varies. For example, a house number in a street address is a variable-length string.

**NOTE:**
The variable-length string is available with US English package *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, it is recommended that you use a custom grammar. Using the provided variable-length grammar as an alternative will result 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 ask for the number of digits, then ask 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 DIP

A recognition 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 Data Interface Process (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,” need a DIP.

For some recognition types, there may be trade-offs 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 WholeWord speech recognition types, see ‘‘WholeWord Speech Recognition Accuracy’’ above.

For more information about custom DIPs, see “Writing the DIP” in Chapter 4, “Data Interface Process,” in *INTUITY*™ *CONVERSANT® System Version 6.0 Application Development with Advanced Methods*, 585-310-761.

For more information on post-processing, see Appendix E, “Recognition Post-Processing.”

## Word Spotting

WholeWord speech recognition also 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.

## Speaker Independence

The speech recognition process compares a person's voice to a set of predeveloped speech models. These models are constructed from thousands of speech samples. 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.

## 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 recognized the caller's response to be and then requests, "If this is correct, say 'yes,' if not say 'no.'" The caller then confirms whether the system recognized the spoken input properly.

For additional information regarding the confirm option, see "Specifying the Confirm Action" in Chapter 7, "Defining the Transaction," in *INTUITY*™ *CONVERSANT*® *System Version 6.0 Application Development with Script Builder,* 585-310-760.

## Barge-In

Barge-in, also referred to as 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.

### Talkoff

Speech recognition accepts either speech or touchtone input in response to a prompt. Barge-in operates much like the talkoff option for touch-tone input, where a caller can interrupt the prompt by pressing a touch tone. 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 the *INTUITY™ CONVERSANT® System Version 6.0 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 the barge-in feature.

For isolated word recognition, the prompt will not stop until the system recognizes a valid vocabulary word. Once the prompt completes playback, the Initial time-out 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 the script, after the answer instruction, the application developer must 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.

You can enable or disable the barge-in feature within the Prompt & Collect action step (assuming barge-in has been globally enabled with the **SR_Prompt** instruction). When enabled, any prompt with the `Speak with Interrupt` field set to “yes” on page 1 of the Define Prompt & Collect screen is interrupted by voice or by touch tones. (See Chapter 8, ‘‘Using Optional Features,’’ of *INTUITY™ CONVERSANT® System Version 6.0 Application Development with Script Builder*, 585-310-760, for more information) If your script does not use `Speak with Interrupt`, be sure your prompts are worded to make sure that callers know they cannot speak until the prompt is finished. For example, look at the wording and the <pause> length of the following prompts. There are no large gaps of silence to tempt callers to speak during the menu prompts. These prompts are designed for use when barge-in is disabled so that callers must wait until the prompt is finished.

### Example 1 — Discouraging Barge-In

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

Example 1 includes short pauses which tell the caller that the system is waiting for their response. The caller answers at the time of the short pause.

### Example 2 — Discouraging Barge-In

“You may order up to five copies. Please say how may copies you would like.”

**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.

Example 2 does not include pauses. The caller waits until the prompt is completed before answering.

### Example 3 — Encouraging Barge-In

These next prompts are designed to encourage callers to barge-in when barge-in is enabled. Note that the pauses are longer, so callers are encouraged to respond after hearing the desired option and the word “now”.

‘‘For sales say, ‘one’ now. <a 1–1.5 second pause>
For service, say, ‘two’ now. <a 1–1.5 second pause>
For a representative, say ‘three’ now.” <a 1–1.5 second pause>

## 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.

## Custom Grammars

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 SP or SSP circuit card, which results in better accuracy. Up to 32 grammars can fit on one SP circuit card on any system. The SSP has no similar restriction.

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. An example is one that includes 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 data interface process (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 SP or 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 following items have a positive impact on WholeWord speech recognition accuracy:

- Isolated word recognition

  Isolated word recognition is very high. In addition, 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 will help improve the overall accuracy of your application, especially when a longer string is input. For more information on improving accuracy of an application, see “Return Top 4” in Appendix E, “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 will increase.

- Prompt structure

  The prompt structure can greatly affect accuracy by promoting a calm, clear voice, helping the caller barge in at the appropriate time or 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>

  For example:

  "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 will 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/No prompts

  Structure yes/no prompts as a yes/no question. 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:

  "If you would like to hear your order again, say 'yes.' Otherwise, say 'no.'"

  Once again, if you want 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.'"

- — Experienced calling population

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

- — 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.

- ■ Informative prompts

  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 only interact with system prompts infrequently (for example, once or twice a year) are more willing to listen to a lengthy prompt.

## Negative Influences on WholeWord Speech Recognition Accuracy

The following items 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 response

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

- ■ Information type

  Recognition of data that is not normally spoken in the form of the digits “0” through “9” will adversely affect accuracy. For example, dollar amounts or the days of the month are not usually spoken in digit form “0” through “9”. To speak the date December 15, the caller would be required to say “1-2-1-5.” Training callers to speak information in this format may 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%. This means 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” amount 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. Finally, 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

  It is important to understand that there are other limitations to the performance of the WholeWord speech recognition feature. 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 to your satisfaction. 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 require far less hardware and software resources
- The number of simultaneous speech recognition calls expected
- The use of the barge-in or Recognize During Prompt option with WholeWord speech recognition increases the amount of hardware and software required resources to process each transaction

## How WholeWord Speech Recognition Works

This section describes the processing involved during WholeWord speech recognition on the INTUITY CONVERSANT system.

### Caller Response

During a Prompt & Collect action, the system prompts the caller for a response. If the Prompt & Collect action allows for voice input, the system locates and reserves a free recognizer resource. If the caller responds using touch tones, SP or SSP resources 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/sec, digitized, and then transferred to the SP or SSP circuit card via a Time Division Multiplexed (TDM) Bus. The TDM bus provides a communication link between the SP or SSP and the telephone interface cards (T1 or Tip/Ring) and also allows the SP or SSP 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. This allows channels to share resources.

### The Recognizer

The SP/CMP and SSP circuit cards are loaded with the WholeWord speech recognition software, which is often referred to as the "recognizer." The recognizer compares the incoming speech sample to only the word models indicated by the recognition type chosen in the Prompt & Collect action. The recognition type is found in the `Recog:` field on the second page of the Define Prompt & Collect screen. The recognizer computes a score for each of the models. The score represents the likelihood that the incoming speech matched a word model indicated by the recognition type. The recognizer returns the most likely candidate to the variable specified in the `Caller Input Field`. For the following example the variable in the `Caller Input Field` is $CI_VALUE. 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 both a model of the numbers “1, 2, or 3.” A score is computed for “1,” a score for “2,” and a score for “3.” Whichever score is highest determines the recognized result. A value of “1, 2, or 3” is returned in $CI_ VALUE. If no value scores high enough, the recognizer returns a “?” in $CI_ VALUE. The application then decides what to do next, based on the value of `Caller Input 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 the top scoring string whose length falls between Min Number of Digits and Max Number of Digits to the script, which is stored in $CI_VALUE.

## Computational Processing

Lucent Technologies provides a Companion (CMP) circuit card to help the SP circuit card with the intensive processing required for recognition algorithms. The CMP circuit card has 12 digital signal processors and is connected to the SP circuit card via a bus. One SP with one CMP circuit card can provide up to six shared channels of simultaneous connected-digit recognition. With the maximum configuration of two CMP circuit cards, the SP can provide up to 12 shared channels of simultaneous connected-digit recognition. For a listing of SP channel capacities for specific features, see “Capacity Information” in Chapter 1, ‘‘Overview of Speech’’

Lucent Technologies also provides a Speech and Signal Processor (SSP) circuit card. The intent of the design is to have the SSP fully replace and do the work of the SP and two CMP circuit cards. The SSP 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 play back, 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 SP and two companion circuit cards use 61 watts of power while the SSP only uses 19 watts per circuit card. For more information on the SSP channel capacities, see “Capacity” section 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 E, “Recognition Post-Processing,” for further information on **recog_dip** and other recognition post-processing functions.

# Recognizing FlexWord™ Speech Input

# 5

## Overview

This chapter describes

- The process by which FlexWord speech recognition recognizes words
- The process by which FlexWord speech recognition provides suggestions for writing application scripts
- Instructions for creating and modifying wordlists

## Purpose

The purpose of this chapter is to ensure that you know how to prepare, create, and administer a FlexWord speech recognition application.

# Introduction to FlexWord Speech Recognition

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 any Prompt & Collect action step. A vocabulary is a set of wordlists associated with a particular FlexWord speech recognition application script.

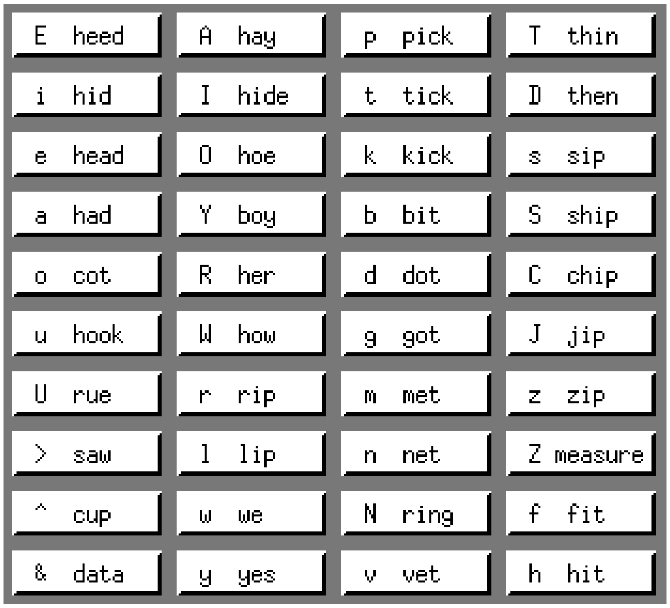
FlexWord speech recognition uses sub-word technology. Sub-word technology relies on phonemic recognition for analyzing and recognizing words.

See Chapter 7, ‘‘Putting It Together’’ for information about getting the most out of your FlexWord speech recognition application.

## Defining Phonemes

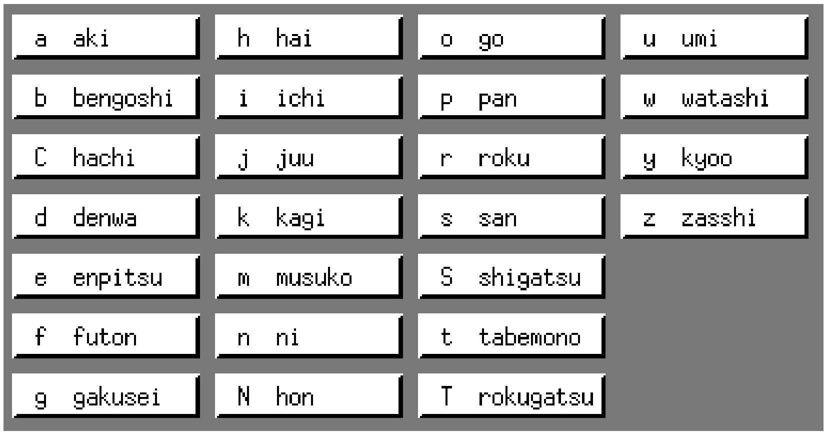
Phonemes are units of sound that form recognizable words when strung together in a particular order. The English language contains 40 phonemes that represent all basic sounds used with the language. The word “sales,” for example, consists of four phonemes: “s-A-l-z.”

Illustrated in Figures 5-1 through 5-5, are FlexWord speech recognition phonemes in US English, Japanese, German, Latin-American Spanish, and French. An example of the phonemes within a word are also provided.



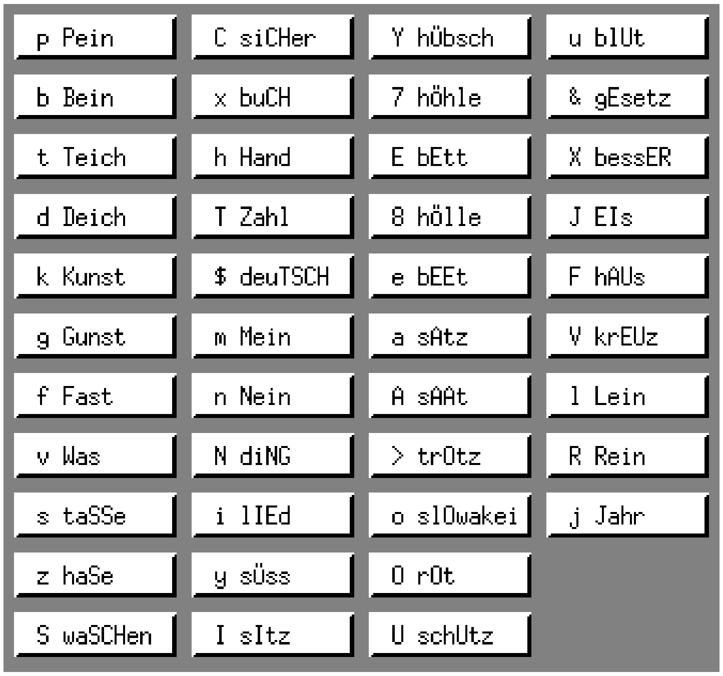
| | | | |
|---|---|---|---|
| E heed | A hay | p pick | T thin |
| i hid | I hide | t tick | D then |
| e head | O hoe | k kick | s sip |
| a had | Y boy | b bit | S ship |
| o cot | R her | d dot | C chip |
| u hook | W how | g got | J jip |
| U rue | r rip | m met | z zip |
| > saw | l lip | n net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

**Figure 5-1. US English Phonemes**



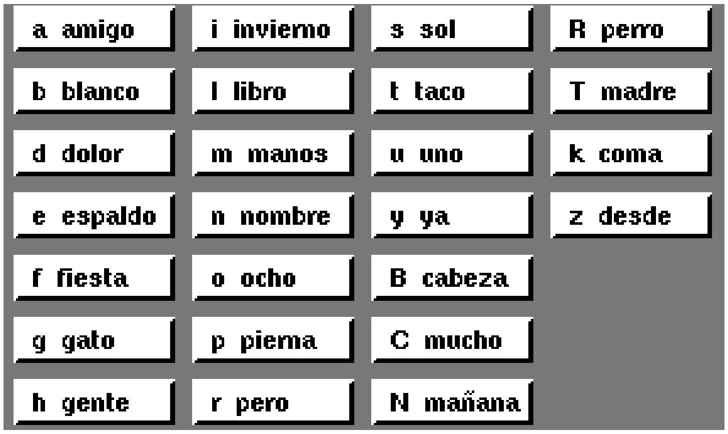
| | | | |
|---|---|---|---|
| a aki | h hai | o go | u umi |
| b bengoshi | i ichi | p pan | w watashi |
| C hachi | j juu | r roku | y kyoo |
| d denwa | k kagi | s san | z zasshi |
| e enpitsu | m musuko | S shigatsu | |
| f futon | n ni | t tabemono | |
| g gakusei | N hon | T rokugatsu | |

**Figure 5-2. Japanese Phonemes**



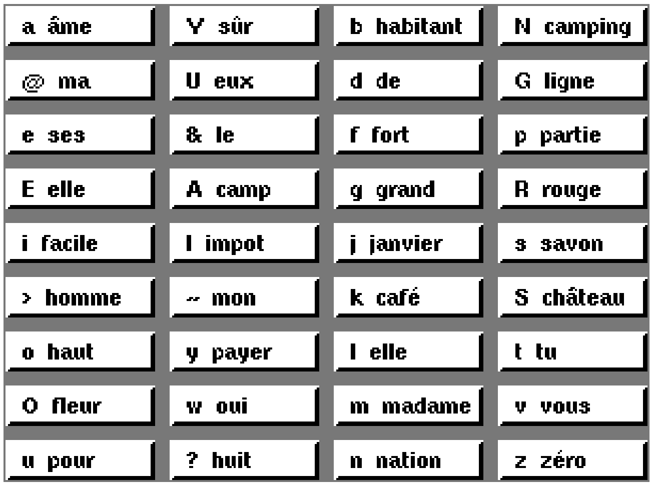
| | | | |
|---|---|---|---|
| p Pein | C siCHer | Y hÜbsch | u blUt |
| b Bein | x buCH | 7 höhle | & gEsetz |
| t Teich | h Hand | E bEtt | X bessER |
| d Deich | T Zahl | 8 hölle | J EIs |
| k Kunst | $ deuTSCH | e bEEt | F hAUs |
| g Gunst | m Mein | a sAtz | V krEUz |
| f Fast | n Nein | A sAAt | 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 amigo | i invierno | 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 pierna | C mucho | |
| h gente | r pero | N mañana | |

**Figure 5-4. Latin-American Spanish 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 5-5. French Phonemes**

## FlexWord's Capacity

*Every* word or phrase in a wordlist is counted as a distinct and separate word. However, the maximum number of words or phrases 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 in 10 of your wordlists, "help" will be counted as 10 separate words. Phrases, such as "loan information," count only as one word. Each wordlist must be given a name tag consisting of 1–14 upper-case characters. For definitions of word, wordlist, or vocabulary terms, see the Chapter , ''Glossary'' in the back of this book.

**Table 5-1. FlexWord's Maximum Capacity**

| | |
|---|---|
| Words/phrases | 2000 |
| Wordlists | 200 |
| Words per wordlist | 500 |
| Characters per wordlist name tag | 64 |
| Phonetic transcription length | 64 |

# Components of FlexWord Speech Recognition

The following sections discuss the components of FlexWord speech recognition.

## FlexWord Toolkit

FlexWord supports fast, low-cost delivery of speech recognition vocabularies on a custom basis through the use of the FlexWord Toolkit. 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/department names. This option gives you a point-and-click graphical environment for adding, deleting, or changing words on new or existing wordlists. A benefit of using the FlexWord Toolkit is that you can use these wordlists in your application immediately after you create them. Additionally, the FlexWord Toolkit supports all available languages. With the release of Version 6.0, running FlexWord Toolkit no longer requires shutting down the voice system.

## Vocabularies

The Lucent Technologies Custom Vocabulary Service is available to build your FlexWord speech recognition vocabularies for a fee.

A FlexWord speech recognition vocabulary can include a maximum of 2000 words/phrases per system, allowing up to 200 wordlists. Each wordlist can include up to 500 words. 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 may be used in more than one word list. 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.

If you have an application that requires more than 2000 words, contact your Lucent Technologies representative.

## 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 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

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 his or her response is invalid and to replay the prompt.

**NOTE:**
A caller must wait until the system is completely finished prompting for information before a response may be spoken.

## Global Support

FlexWord speech recognition supports the following languages:

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

The languages new to version 6.0, French and Japanese, are only available while using the SSP circuit card.

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

### 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 Appendix E, “Recognition Post-Processing,” for further information on **recog_dip** and other recognition post-processing functions.

## Preparing a FlexWord Speech Recognition Application

This section includes information about preparing a FlexWord speech recognition application.

### Designing the FlexWord Speech Recognition Application

The following sections will aid you in defining and using 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 5-6 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 5-6 and Figure 5-7 contain groups of possible words from which callers can choose.

**NOTE:**
All of the wordlists for all of your applications constitute the “vocabulary.”

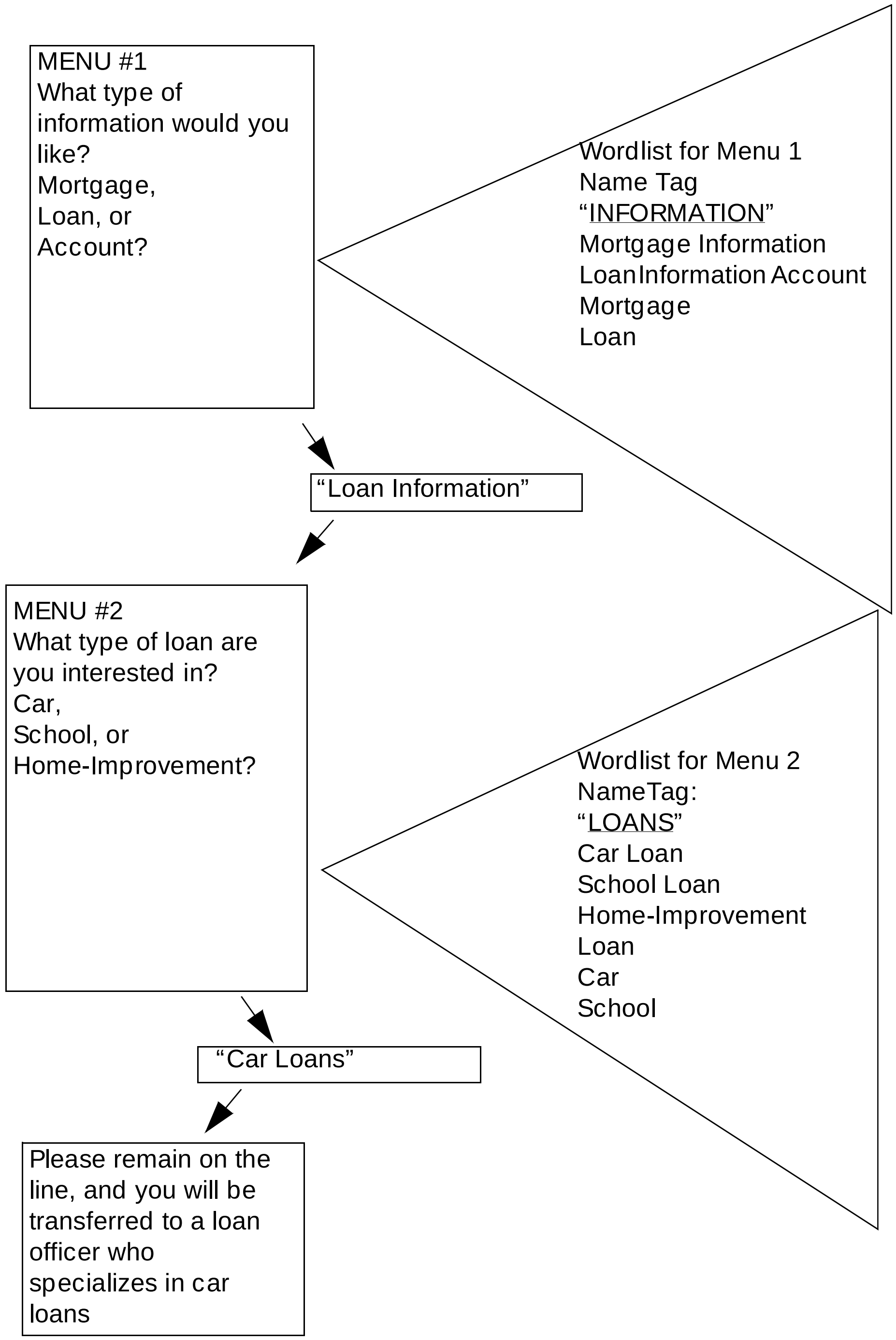


**Figure 5-6. FlexWord Speech Application Example with Menu Prompts**

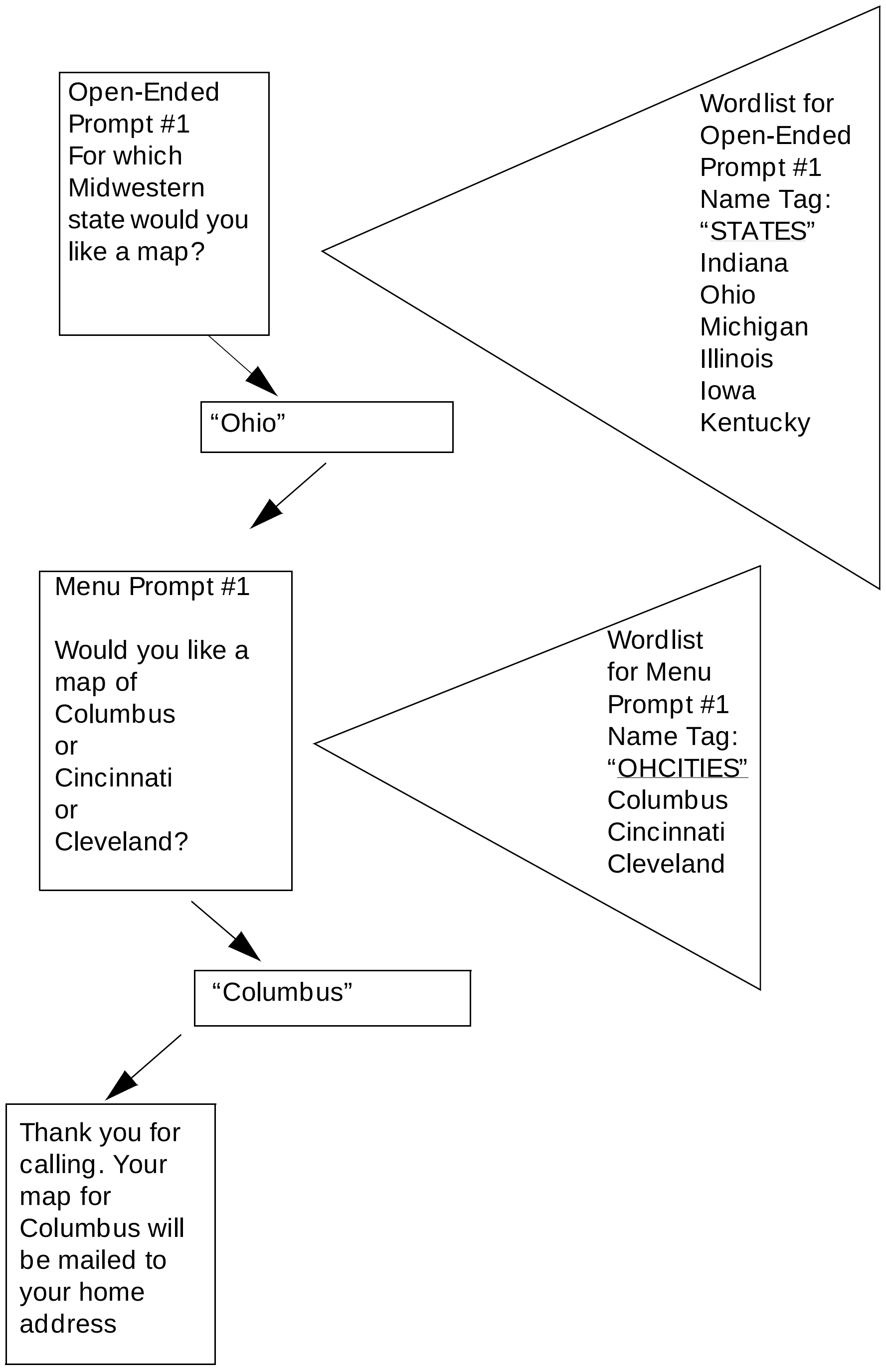


**Figure 5-7. FlexWord Speech Application Example Using Open-Ended and Menu Prompts**

## Using Wordlists in Scripts

Each application can use several wordlists. On the second screen of your Prompt & Collect action step, you must specify the name of the wordlist to be used for the prompt in the `Recog:` field. The Define Prompt & Collect screen shows you all available wordlists. FlexWord speech recognition looks to the wordlist that you have designated by name in the `Recog:` field of the Define Prompt & Collect screen.

The directory **/att/asr/wordlists/active** is the directory for all active wordlists. Wordlist name tags are derived from the 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 also appears as one of the choices in the `Recognition_Type` field of the Define Prompt & Collect screen. 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 wordlist 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. One should be aware that defining your wordlist appropriately is very important for the success of your application.

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

# Creating Wordlists

This section describes the two ways in which you can create FlexWord speech recognition wordlists.

## Purchasing a Custom Speech Package from Lucent Technologies

This procedure involves preparing a FlexWord speech recognition application and then contacting Lucent Technologies for the customization.

Contact Lucent Technologies FlexWord speech recognition technical support staff at (404) 242-1551 after deciding upon and naming your wordlists. Inform them of your wordlists and vocabulary, and they will work with you to customize your vocabulary package.

Once you receive your customized wordlists, you can begin implementing your application.

## Using the FlexWord Toolkit

The FlexWord Toolkit allows you to construct words and phrases by stringing phonemes together. The toolkit includes a standard dictionary, according to the language chosen, as well as three commands for creating and modifying wordlists. The sections that follow include information about hardware and software requirements; configuring the hardware; and creating, editing, and deleting wordlists.

**NOTE:**
The Version 6.0 toolkit and its resulting wordlists can only be used with the Version 6.0 system.

### Software Requirements

The software requirements for the FlexWord Toolkit are as follows:

- FlexWord Toolkit package
- Mouse driver software (serial port connection, COMP 2 *only*)
- Text-to-Speech (TTS) package (available in US English *only*)

For more information, see the INTUITY™ CONVERSANT® maintenance book specific to your system platform.

## Hardware Requirements

The hardware requirements are as follows:

- Mouse
- Tip/Ring circuit card
- Speaker/amp system or telephone line to the Tip/Ring circuit card
- SP circuit card (AYC9) or SSP circuit card (AYC43) with TTS functionality assigned

See Chapters 5 through 8 in the INTUITY CONVERSANT installation book specific to your system platform for more information.

### Configuring Hardware for the FlexWord Toolkit

This section includes configuration information for the FlexWord Toolkit.

- Installing and administering the mouse

  For instructions on administering the mouse, see “Adding a Mouse” in Chapter 1, “Setting Up the Work Environment,” of *System Setup and Configuration* in the Novell UnixWare Documentation Set, 585-350-908.

  In order to save the changes to the mouse configuration that you have just added conduct the following steps. In case you need to remove the installation of the mouse, the procedure is shown in the reverse order in Chapter 1, “Setting Up the Work Environment,” of *System Setup and Configuration* in the Novell UnixWare Documentation Set, 585-350-908. At the conclusion of adding a mouse in the **mouseadmin** command, make sure to select:

  1. Select **U** (Update Mouse Configuration and Quit)
  2. Enter **mouseadmin** before you can test the mouse configuration
  3. Select **T** (Test your mouse configuration)

- Configuring the Tip/Ring circuit card

  The Tip/Ring circuit card plays the phonetic pronunciation of the words, for US English *only*, in a wordlist. It is most convenient to configure your Tip/Ring circuit card with O.S. Index 0. However, it is not required. If your O.S. Index is not zero, you must specify the correct channel number with the **wl_edit** command. See Chapter 5, “Replacing, Installing, or Upgrading Circuit Cards” in the INTUITY CONVERSANT System Version 6.0 maintenance book specific to your platform, for more information about the Tip/Ring circuit card.

- Configuring the speaker/amp system

  If you use an external speaker/amp system for phonetic pronunciation from within the FlexWord editor, the AUDIO OUT jack on the faceplate of the Tip/Ring circuit card must be connected to the speaker/amp. For more information about the Tip/Ring circuit card, see Chapter 3, “Making Cable Connections,” in the INTUITY CONVERSANT System Version 6.0 hardware installation book specific to your platform.

- Connecting the telephone line

  If you use a telephone line for the phonetic playback within the FlexWord editor, 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,” in the INTUITY CONVERSANT system installation book specific to your platform 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/amp 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 SP or SSP Text-to-Speech circuit card

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

  Once configured for TTS, the SP or SSP circuit card generates the phonetic pronunciation from within the FlexWord editor. Configuring your SP or SSP circuit card for TTS functionality involves several steps. First, you must determine which SP or SSP circuit card should be assigned Text-to-Speech functionality. The SP circuit card must be an AYC9. To determine which AYC9 SP circuit card or SSP circuit card should be assigned TTS functionality, perform the following procedure:

  **NOTE:**
  The following procedure applies to both the SP and SSP circuit cards.

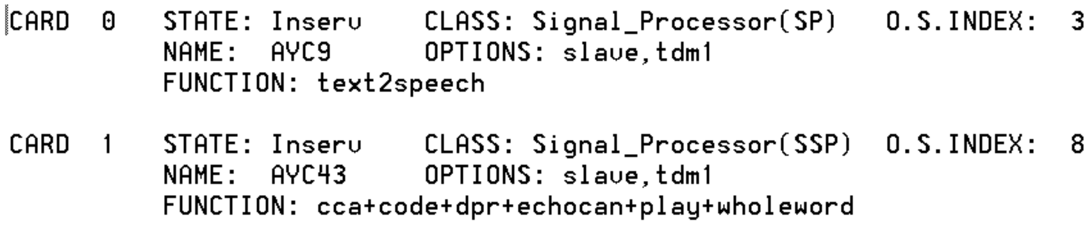
1. Enter **display card sp**

   Once you get the results of the **display card** command, make a note of the card number and the O.S. Index.

2. Assign TTS to the chosen circuit card number by using the cvis_ menu screens.

**NOTE:**
Before you begin using the FlexWord Toolkit, verify that all the circuit cards needed are in service. For more information about verifying circuit cards are inservice, see “Verifying Installed Hardware” in Chapter 5, “Verifying Hardware, Software, and System Status”, in the INTUITY CONVERSANT System Version 6.0 installation book specific for your platform.

The system displays output similar to that shown in Figure 5-8.



```
CARD  0   STATE: Inserv    CLASS: Signal_Processor(SP)   O.S.INDEX:  3
          NAME:  AYC9      OPTIONS: slave,tdm1
          FUNCTION: text2speech

CARD  1   STATE: Inserv    CLASS: Signal_Processor(SSP)  O.S.INDEX:  8
          NAME:  AYC43     OPTIONS: slave,tdm1
          FUNCTION: cca+code+dpr+echocan+play+wholeword
```

**Figure 5-8. Output from the Display Card Command After Assigning TTS Functionality**

## Creating and Modifying FlexWord Wordlists

This section describes how to create vocabularies and wordlists, add words and phrases, delete words and phrases, and make phonetic changes in a wordlist by using the following commands:

- **wl_init**

  The **wl_init** command creates the phonetic spelling for each word or phrase in a wordlist.

- **wl_edit**

  The **wl_edit** command invokes the FlexWord editor, which is a phoneme editor that allows you to hear and change the pronunciation for each of the words in your file.

  **NOTE:**
  **wl_edit** command works with US English *only*

- **wl_copy**

  The **wl_copy** command copies wordlists to disk.

**NOTE:**
You do not have to stop the voice system to run the FlexWord editor.

To create a FlexWord wordlist, perform the following procedure:

1. Select a working directory.

   You may want to create a directory with a name similar to the wordlist that you want to create.

2. Create a file in the working directory using vi or some other editor that contains all of the words you would like to recognize at a particular prompt. The following rules apply when using the FlexWord editor:

   - The filename of the wordlist must be uppercase, alphanumeric, and 14 characters or less in length.
   - Each word or phrase in the wordlist must be on a separate line.
   - White space is not allowed, and you must separate words in a phrase with an underscore (_).
   - The maximum number of words in a wordlist must not exceed that specified in Table 5-1.

   If you are creating wordlists for the first time, see ‘‘Using Wordlists in Scripts’’ above for tips on creating wordlists.

3. Enter **wl_init *<filename>* [ *language*]**

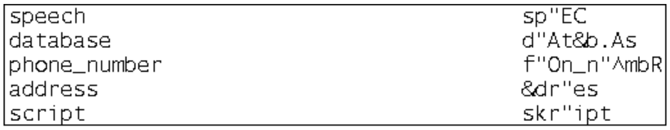
   To add a phonetic pronunciation to each of the words and phrases in the file, enter the language of the filename.

   The system 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.
   ```

   See Figure 5-9 for an example of the output.

   **NOTE:**
   You can also view the phonetic spelling of a word once you invoke the FlexWord editor by clicking on a word in your wordlist and looking at the phonetic spelling in the Phonemes window at the bottom of the screen.



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

**Figure 5-9. Example of a Wordlist File with Phonetic Spelling For US English**

4. To invoke the FlexWord editor (Figure 5-10), enter
   **wl_edit [-l *chan#*] [-D *dirname*] [-O] [-L *language*]** where

   - [-l *chan#]*

     *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.

   - [-D *dirname]*

     *dirname* is the directory in which the FlexWord editor looks for word list files.

   - [-O]

     This gives the maximum messaging output to the screen.

   - [-L *language]*

     *language* specifies the language output that appears on the screen, that is English, French, German, Japanese or Spanish.

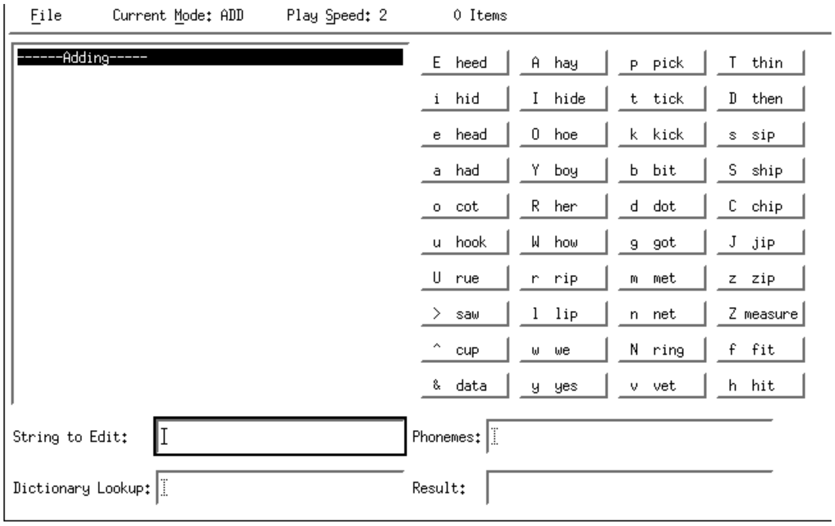


**Figure 5-10. FlexWord Editor**

**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.**

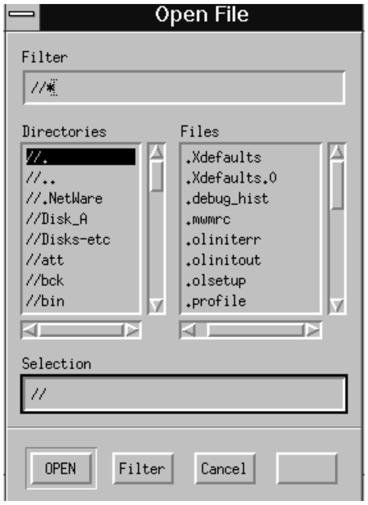
5. Call into the Tip/Ring channel number you previously specified, if you are set up to call into a telephone line.
6. Select `File` from the menu bar at the top of the FlexWord editor.

   The system displays the file menu.
7. Select `Open`.
8. Select the wordlist file you created from the files listing.

**NOTE:**
The directory that appears is the last directory you were working in before invoking the FlexWord editor; therefore, it may not be the directory in which you created your wordlist file. You may need to select the directory in which the wordlist was created.

9. Select `Open`.

   The system displays the Open File menu  (Figure 5-11).



Filter: //*

Selection: //

**Figure 5-11.  Open File Menu in the FlexWord Editor**

> **NOTE:**
> Upon opening your file, TTS pronounces the first word or phrase in your file.

> **NOTE:**
> The following steps are for US English *only*.

10. Listen to the phonetic pronunciation of each word by selecting each word in the wordlist.

> **NOTE:**
> The Wordlist Editor can be used to change the word to reflect regional dialects.  You may want to include two or more versions of the same word or phrase to account for different dialects.

11. Indicate any necessary changes to the phonetic spelling of your words/phrases in the Phonemes window. Use the following emphasis marks as well as the phonemes shown in Figure 5-10.

    - (quote) “

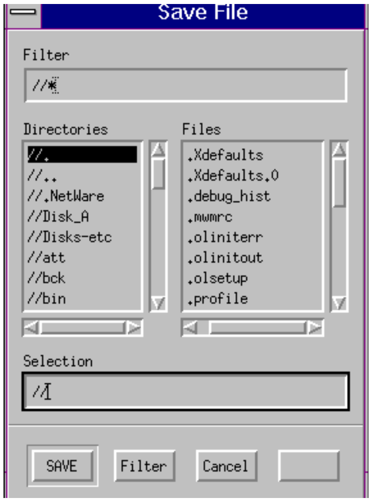
      This mark indicates primary emphasis and must go before a vowel sound.

    - (period) .

      This mark indicates secondary emphasis and must go before a vowel sound.

12. Select `Save` from the File menu to save the changes you made to your wordlist.

    The system displays an example of the Save menu (Figure 5-12).



Filter: //*

Directories:

```
//.
//..
//.NetWare
//Disk_A
//Disks-etc
//att
//bck
//bin
```

Files:

```
.Xdefaults
.Xdefaults.0
.debug_hist
.mwmrc
.oliniterr
.olinitout
.olsetup
.profile
```

Selection: //

**Figure 5-12. Save File Menu in the FlexWord Editor**

13. Select `Quit` from the File menu to quit working in the FlexWord editor.

### Adding Words and Phrases to a Vocabulary

Once you have created your wordlist, you can add words or phrases to the wordlist with the FlexWord editor. To add a word or phrase to your file, perform the following procedure from the FlexWord editor:

1. Select Open from the File menu.
2. Choose your wordlist file from the Files directory.
3. Select Open.

   **NOTE:**
   Upon opening your file, TTS pronounces the first word or phrase in your file.

4. Decide where you want the new word or phrase to be placed in your file. The new word or phrase will be added below the currently selected word or phrase.
5. Select Add from the Current Mode menu.

   The system displays the String to Add window.

6. Type the word or phrase to be added in the String to Add window.
7. Press ENTER to hear the pronunciation.

   The dictionary looks up the new word and adds it to the wordlist and TTS pronounces it phonetically.

   **NOTE:**
   TTS pronounces the word or phrase according to what the dictionary tells TTS. If the word or phrase is not pronounced correctly (for example, “live” has two pronunciations), you can change the phoneme string so that TTS pronounces it the way you want it. However, upon subsequent lookups of the word or phrase, TTS will still pronounce it according to the dictionary look up. You have to manually adjust the way in which TTS pronounces the word by using the Phonemes window.

   You can also listen to the pronunciations of each phoneme by clicking on the phonemes as shown in Figure 5-10.

   **NOTE:**
   The following step is for US English *only*.

8. Indicate any necessary changes to the phonetic spelling of your words/phrases from the Phonemes window. Use the following emphasis marks as well as the phonemes shown in Figure 5-10.

   - (quote) “

     Indicates primary emphasis and must go before a vowel sound.

   - (period) .

     Indicates secondary emphasis and must go before a vowel sound.

9. Press ENTER again to add the word or phrase to the list.
10. Return to Step 3 if you need to add more words or phrases to the wordlist.

### Deleting Words and Wordlists from a Vocabulary

To delete a word or phrase from your file, perform the following procedure:

1. Enter **wl_edit**

   The system displays the Current Mode menu.

2. Select `Edit`
3. Select the word or wordlist that you wish to delete.
4. Select `Delete` from the Current Mode menu.

   The designated word or phrase is removed from the file.

**NOTE:**
You must select `Edit` before selecting `Delete`. If you try to delete a word or phrase while in the Add mode, the system displays the following message:

```
'Cannot DELETE while in ADD mode!'
```

### Copying Wordlists

To copy a wordlist to a target machine with FlexWord speech recognition loaded, perform the following procedure:

1. Move to the directory where your files are located.
2. Enter **wl_copy *<filename> <filename> <filename> ...***

   The system copies the files or directories out to floppy disk. If any of the wordlist names are directory names, the contents of the directories are also copied to floppy disk.

**NOTE:**
Names should be relative pathnames, not absolute pathnames, since they will be used to load the vocabularies onto a FlexWord recognition system.

3. Take the disk to a machine with FlexWord Speech Recognition installed.
4. Enter **wl_install.**

   The system copies the wordlists from your disk to the designated machine.

**NOTE:**
You do not have to use a floppy disk to copy wordlists. If you have FlexWord recognition installed on the same machine, you can copy your wordlists to active and/or inactive directories.

## Special FlexWord Recognition Accents

The following lists the special accents that are used for global languages.

### Spanish FlexWord Recognition Accents

- á
  ‘a
- é
  ‘e
- í
  ‘i
- ó
  ‘o
- ú
  ‘u
- ñ
  ~n

### Japanese FlexWord Recognition Accents

- û
  uu
- n’
  n’

Repeat any phoneme with a circumflex (^) over it. For example: aa

Do not use apostrophes or bars over vowels. For example: í

## French FlexWord Recognition Accents

- â
  (a^
- à
  (à
- á
  (a’
- c’
  (c’
- ç
  (c,
- d’
  (d’
- ë
  (e:
- é
  (e’
- è
  (è
- ê
  (e^
- ï
  (i^
- l’
  (l’
- ô
  (o^
- ù
  (ù
- û
  (u^
- ü
  (u:

### German FlexWord Recognition Accents

- ä
  ae
- ö
  oe
- ü
  ue
- ß
  ss

## FlexWord Recognition Vocabulary Administration

Vocabulary administration basically involves installing wordlists and moving them between the active and inactive directories. This section includes how to install, activate, deactivate, and remove wordlists.

### Installing Wordlists

To install a wordlist onto the voice system, perform the following procedure:

1. Enter **wl_install**
2. Insert the floppy disk containing the wordlists into the drive when prompted.

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

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

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

## Activating Wordlists

Activating a wordlist means creating all of the data files necessary to perform a FlexWord speech recognition. To activate a new wordlist, perform the following procedure:

1. Enter **ln *filename* /att/asr/wordlists/active**
   where *filename* is the name of the wordlist you want to activate.

2. Enter **wl_gen [*language*]**
   where [*language*]= English, French, German, Japanese or Spanish.

   ⚠ **CAUTION:**
   *Do not run ScriptBuilder at the same time as the **wl_gen** command.*

   The **wl_gen** command verifies the format of all active wordlists and creates the data files needed for FlexWord speech recognition. If the **wl_ gen** command finds more than 38 phonemes, including the underscore (_), within any word or phrase within a wordlist, you receive an error message. Once **wl_gen** finds an error within a wordlist, it quits looking at that wordlist, and the wordlist containing that word or phrase is not used when generating the FlexWord recognition data files. Therefore, if you receive an error, you may have to run **wl_gen** several times to locate each error. For more information about the **wl_gen** command, see *INTUITY*™ *CONVERSANT® System Version 6.0 Administration*, 585-310-591.

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

   See *INTUITY*™ *CONVERSANT® System Version 6.0 Administration,* 585-310-591, for more information about the **diagnose card** command.

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

   See “Verifying and Installing the Application,” in Chapter 10, “Application Administration,” *INTUITY*™ *CONVERSANT® System 6.0 Application Development with Script Builder,* 585-310-760, for information about verifying and installing your Script Builder application.

## Deactivating Wordlists

Deactivating a wordlist means removing all of the data files associated with a particular vocabulary. To deactivate a wordlist, perform the following procedure:

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

   For example, enter **rm filename**

   where *filename* is the name of the wordlist that you want to deactivate.
3. Enter **wl_gen[*language*]**

   where [*language*]= English, French, German, Japanese or Spanish.

   For more information about the **wl_gen** command, see *INTUITY™ CONVERSANT® System Version 6.0 Administration*, 585-310-591.
4. Enter **diagnose card** to diagnose the FlexWord speech recognition SP or SSP circuit card.

   See *INTUITY™ CONVERSANT® System Version 6.0 Administration,* 585-310-591, for more information about the **diagnose card** command.
5. Verify and install the application associated with the active wordlist before you call up the application.

   See “Verifying and Installing the Application,” in Chapter 10, “Application Administration,” *INTUITY™ CONVERSANT® System 6.0 Application Development with Script Builder,* 585-310-760, for information about verifying and installing your Script Builder application.

## Removing Wordlists

Removing a wordlist involves deleting both the data files and wordlists from the system. To remove a wordlist, perform the following procedure:

1. Deactivate the wordlist. See ‘‘Deactivating Wordlists’’ above for the procedure.
2. Remove the associated wordlists from **/att/asr/wordlists/inactive**
   a. Enter **cd /att/asr/wordlists/inactive**
   b. 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. Therefore, given the variance in human speech and the statistical properties of recognition algorithms, the speech recognizer will make occasional errors.

Factors that have a positive influence on accuracy rates include effective construction of prompts and wordlists, as well as experienced calling populations. Low recognition accuracy rates are usually caused by inexperienced callers who speak extraneous phrases and/or speak before the prompt is finished, and wordlists that contain short words, rhyming words, or a large number of words.

## Positive Influences on FlexWord Speech Recognition Accuracy

The following items have a positive influence on recognition accuracy:

- 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 (applications that allow employees to speak a colleague’s name rather than enter an extension number) increases recognition accuracy. In other types of applications, syllabic and vowel similarities may contribute to recognition confusion. For example, “women’s wear” and “men’s clothing” are more effective wordlist phrases 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.”

  Pay special attention to 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 customers’ requests.

Also pay attention to words your customers may use that are outside of your vocabulary wordlist. These "out of vocabulary" (OOV) words should be placed in the wordlist preceded by two dashes. This will increase overall accuracy by correctly recognizing those words that are preceded by dashes and properly classifying them as being out of vocabulary. 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 though they were not accurately recognized. A sample OOV wordlist is delivered with each FlexWord speech recognition language. This list contains some of the most frequent words used by callers interacting with a voice response system.

The OOV wordlist supplied by the FlexWord package is
"**</vs/asp/flexword<*language*>.oov**"

where the languages can be English, French, German, Japanese or Spanish. This wordlist or any custom OOV wordlists should be appended to each of your regular vocabulary wordlists.

- Experienced calling population

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

- Prompts with structure and good design

  Prompts offered in calm, clear voices greatly affect recognition accuracy, as do the specific structures of the prompts. Prompts should guide the caller to say desired words/phrases.

  — Prompts that are set up to dissuade callers from barging-in or speaking before the prompt is finished increase recognition accuracy. For example:

  > "Please say the month of your birth, now."

  — For best results, menu prompts should be built with the structure of <desired result> <action required>. 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 the <action required>. Do not list the <action> before the <desired result>. For example, the prompt, "Say 'one' for a description of the upcoming gallery events" encourages the caller to forget the specific <action required>, since the last thing related is the <desired result>.

- 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.’’’

  Thus, the caller can make sure that her or his words matched the recognized response. See Figure 5-13 for an illustration of a confirmation path.

- Informative prompts

  Lengthy prompts providing detailed response instructions may improve recognition accuracy. Generally, from the perspective of frequent users, these types of prompts are annoying, since experienced users will have 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.

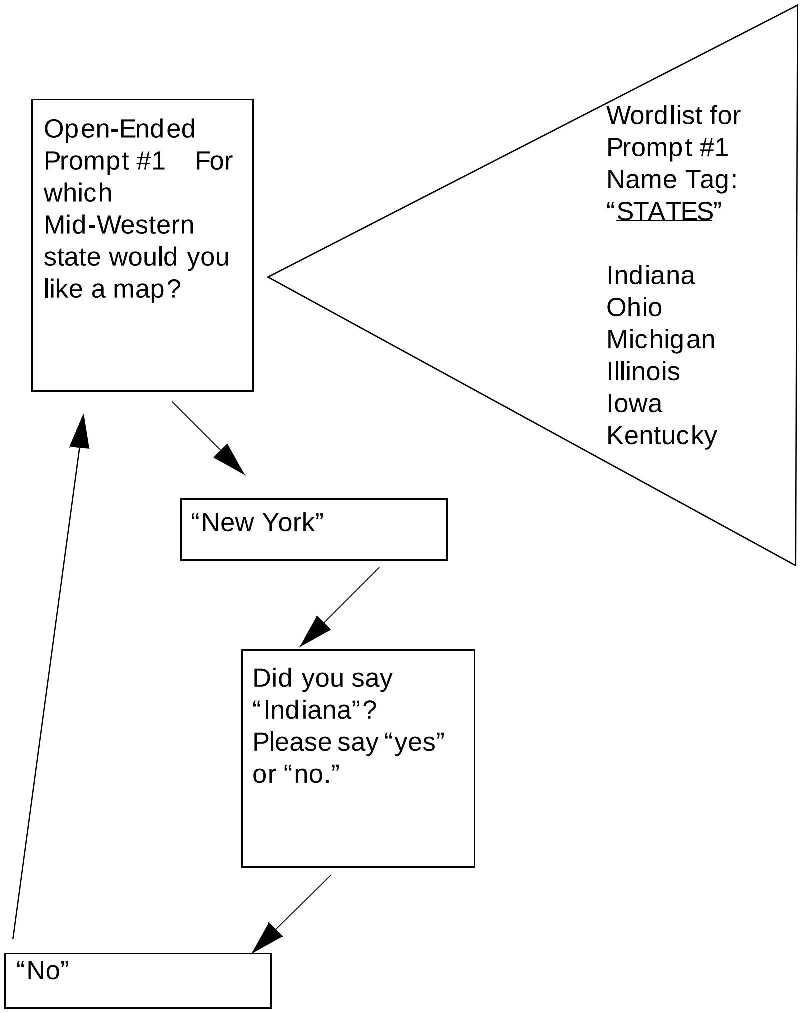

Open-Ended Prompt #1 For which Mid-Western state would you like a map?
Wordlist for Prompt #1 Name Tag: "STATES"
Indiana
Ohio
Michigan
Illinois
Iowa
Kentucky
"New York"
Did you say "Indiana"? Please say "yes" or "no."
"No"

**Figure 5-13. Example of a Repromt in a FlexWord Application**

## Negative Influences on FlexWord Speech Recognition Accuracy

The following items have a negative influence on recognition accuracy:

- Barging-In before the prompt is finished

  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 prompt shows how recognition accuracy can be increased by including a time reference:

  > “Please say the name of the agent with whom you wish to speak, now.”

  The previous prompt encourages the caller to wait until the prompt is finished before responding.

  > “Please say the name of the agent with whom you wish to speak.”

  This second 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.

  **NOTE:**
  If an application is using FlexWord speech recognition, you must specify “no” at the `Speak with Interrupt` on the Define Prompt & Collect screen.

- 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. The following are examples of problems that occur in ineffective wordlists:

  — Short words

  One-syllable words that have the same vowel sounds are more 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.

  — Rhyming words

  Wordlists that include words with similar rhyming vowel sounds can cause a decrease in recognition accuracy. Again, the system matches sounds and strings these sounds together. Pronounce the words on your wordlists and vary the vowel sound of each word if possible.

  — 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.

## WholeWord Speech Recognition Versus FlexWord Speech Recognition Accuracy

For the speech recognition feature, accuracy varies based upon the recognition method and application in the following order from highest to lowest:

1. WholeWord speech recognition isolated digit
2. WholeWord speech recognition connected digit
3. FlexWord speech recognition isolated word

# Recognizing Dial Pulse Input

# 6

## Overview

This chapter describes Dial Pulse Recognition (DPR) and how it allows users with rotary telephones or push-button telephones that generate only dial pulses to respond to the INTUITY™ CONVERSANT® system.

## Purpose

The purpose of this chapter is to ensure that you understand

- The components of DPR
- How DPR works
- The accuracy of DPR

# Introduction to 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/clicks” on the line to dial pulses. DPR is limited to the installation of systems that contain the SSP circuit card. DPR will not run on an SP circuit card. DPR supports digits “0” through “9” on analog and digital interfaces.

# Components of Dial Pulse Recognition

This section gives in-depth details about the components of DPR.

- Global Support
- Simultaneous Dial Pulse Capabilities
- Standard Dial Pulse Recognition Types
- Grammars
- Recognition Confirmation
- Barge-In

## Global Support

Rotary telephones or push-button telephones that generate dial pulses are used in a variety of countries. The DPR feature is available to an assortment of countries to accommodate the dial pulse users.

## Simultaneous Dial Pulse Capabilities

DPR can work simultaneously with either WholeWord 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 keying the number in with dial pulse or touchtone, or using spoken input.

## Standard Dial Pulse Recognition Types

Recognition types are the choices that are associated with the Recog: field on the Script Builder Define Prompt and Collect screen. The system uses the recognition type along with other fields on the Define Prompt and 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, if you had a credit card whose first number had to be a 3 or 5, and the second number had to be a 1 or a 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 from 1 through 3 | DP1_3 |
| Single digit from 1 through 5 | DP1_5 |
| Single digit 2, 5, or 8 | DP258 |
| Single digit 1, 4, 7, or 0 | DP1470 |
| Any digits from 1 through 9, 0 | DP1_10 |
| Any digits from 2 through 9, 0 | DP2_10 |
| Any digits from 3 through 9, 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 rejects the input, if no match fits.

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 are described in “Using Dial Pulse Recognition” in Chapter 8, “Using Optional Features,” of *INTUITY™ CONVERSANT® System Version 6.0 Application Development with Script Builder,* 585-310-760.

## 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, a script prompts a caller to "Please enter your area code," and the caller dials or speaks $\boxed{6}$ $\boxed{1}$ $\boxed{4}$. 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. This example can be constructed using the DP258 recognition type, thereby restricting the expected input to only 2, 5, or 8.

When the system is asking for a long string of numbers such as an account number, it processes one digit at a time. Although this is time consuming, it establishes higher accuracy and decreases the likelihood of the system accepting a faulty account number.

## Barge-In

Barge-in, also referred to as Recognize During Prompt, is the ability of the system to allow callers to interrupt during voice playback by entering DPR digits. At the release of the Version 6.0 INTUITY CONVERSANT system, barge-in is not supported for use with DPR. There is, however, the intent to provide this feature in the future.

# How Dial Pulse Recognition Works

This section describes the processing and the recognizer features of DPR on the INTUITY CONVERSANT system.

## Computational Processing

When the caller enters dial-pulse digits, they are collected by the algorithm on the SSP circuit card and processed according to the selected recognition type. A maximum of 60 channels are available per SSP circuit card.

**NOTE:**
Dial Pulse Recognition *must* be run on the SSP circuit card. This recognition package is *not* available for the SP circuit card.

## The Recognizer

The SSP circuit card is loaded with the DPR software, also referred to as the "recognizer." The recognition type is found in the `Recog:` field on the second page 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

The ability of the system to comprehend the caller's input quickly and accurately is very important. This section discusses recognition and training, the elements that help to ensure the accuracy of the DPR feature.

## Recognition

The dial-pulse recognizer detects the "pops/clicks" on the line that are generated when the caller dials using a rotary telephone capable of generating dial pulses. DPR recognizes the digits "0 through 9" available on dial-pulse telephones.

False recognition can be caused 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, could be recognized as a dial pulse (DP) digit. Speakerphones can aggravate this condition since they amplify room noise. False recognition is reduced once training has been completed (see ''Training'' below for more information).

When using the DEFINITY® G3 switch, you may see some unexpected behavior from the dial-pulse recognizer. DEFINITY G3 converts DP digits to dual tone multifrequency (DTMF) only when the input telephone is directly connected to the DEFINITY switch. This eliminates the need for DPR since the system will always respond to DTMF.

Some Central Office (CO) switches may not pass DP digits through once a call has been established. The callers local switch (CO) may see the loss of loop current as a hang up or register recall and disconnect the caller. Although this is not a common problem, there is nothing the INTUITY CONVERSANT system's DPR can do in this situation.

**NOTE:**
There is no way for DPR applications to receive the pound sign (#), asterisk (*), or the letters "a through d," which are available on DTMF telephones. Applications using DPR must be redesigned accordingly.

## 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 five 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 has the option of being turned on or off. When turned on, training is automatic in that it shuts off after the caller has entered the number greater than five.

## DPR Troubleshooting

*Take the actions directed to correct the following problems with DPR.*

*How can I tell if DPR is 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 a SSP circuit card and is in service. The test will ask you to dial any four DP digits and speak the result back to you.

*Why is DPR not recognizing any digits?*

There are a number of things that could cause DPR not to work. First, make sure all administration and assignments of DPR are correct. Use “**feature_tst**” to determine if it is an application or system issue. The system error logs should be checked and corrective action taken where appropriate.

If the above actions do not reveal the problem, contact your Lucent Technologies Technical support center representative at 1-(800)-242-2121.

*Why is DPR mis-recognizing 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 has two common failure modes, which are explained below.

### Noise on the Line

As previously noted, noise on the line can cause the system to incorrectly recognize a digit. Most often this occurs with a “1” and sometimes a “2.” Providing that you have application control over the requested user input, you can avoid the use of these digits.

### Missing or Extra Pulse

The other failure mode is a missing or extra pulse. When the SSP 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 pin, you may decide to accept the input even though the pin number is off by one digit.

There are standard recognition types for single-digit prompts requiring four or less 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 your speech-related features
- Why certain features work well together
- When you can use them together

## Purpose

The purpose of this chapter is to ensure that you

- Understand how to get the most out of your speech-related features
- Understand which features complement each other

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

For information on how to incorporate WholeWord speech recognition into your touch-tone application, see *INTUITY™ CONVERSANT® System Version 6.0 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. This section offers guidelines for getting the most out of FlexWord speech recognition.

## Building a FlexWord Speech Recognition Vocabulary

As discussed inChapter 5, ‘‘Recognizing FlexWord™ Speech Input’’ you can build FlexWord speech recognition vocabularies two ways:

- Use the optional FlexWord Toolkit to make your own wordlists. Advantages to using the FlexWord Toolkit are that you construct wordlists quickly and easily and use them in your application immediately after you define them as opposed to having to wait for wordlists.
- Have Lucent Technologies build your vocabularies through the Lucent Technologies Custom Vocabulary service.

Consider the size of your vocabularies, the number of FlexWord speech recognition application scripts you plan to have, and how much your vocabulary may change over time to help you decide whether or not to purchase the FlexWord Toolkit or the Custom Vocabulary service.

## 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."

## Sound of a Word

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

This section includes two examples of target FlexWord speech recognition applications.

### 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. See Figure 7-1 for an example of a menu-based application.

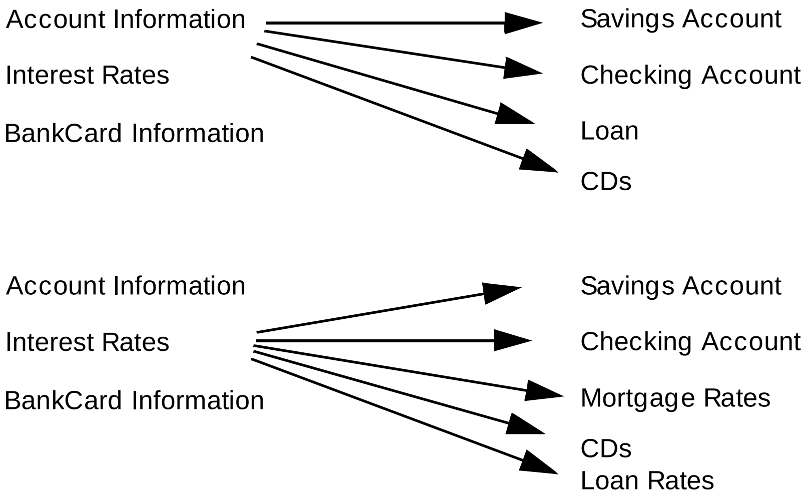


**Figure 7-1. Example of Menu-Based Application**

### Applications Requiring Large Wordlists

Applications that require large, custom wordlists would 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.

## Getting the Most out of Dial Pulse Recognition

As discussed in “Introduction to Dial Pulse Recognition” in Chapter 6, ‘‘Recognizing Dial Pulse Input’’ DPR is a feature that recognizes caller pulse inputs from a rotary telephone or from a push button telephone that generates dial pulses. The following tips will help you get the most out of the Dial Pulse Recognition feature.

- Design your application so that the first digit is a five or higher 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 6.0 Application Development with Script Builder,* 585-310-760.
- 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

This section offers several tips for using Text-to-Speech (TTS) 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.
- Text-to-Speech is available in US English *only*.

## Using FlexWord Speech Recognition and Text-to-Speech

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 in US English *only*.

## Using Text-to-Speech and Prerecorded Speech

A single script can speak prompts and announcements in prerecorded speech only, Text-to-Speech (TTS) only, or a combination of both. Below is an example of a script that uses both.

In the example below, 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 acct. num."
```

```
              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
```

## 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. When using Script Builder, you can specify that a Prompt and Collect action 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.

However, each Prompt and Collect action step must specify either FlexWord or WholeWord speech recognition. Both recognizers cannot be used in the same prompt. See Figure 7-2 for a model of a retail application example that combines FlexWord wordlists and the WholeWord “yes/no” speech recognition feature.

In Figure 7-2, 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 per Define Prompt & Collect screen, each wordlist must contain *all* possible choices for a single prompt.

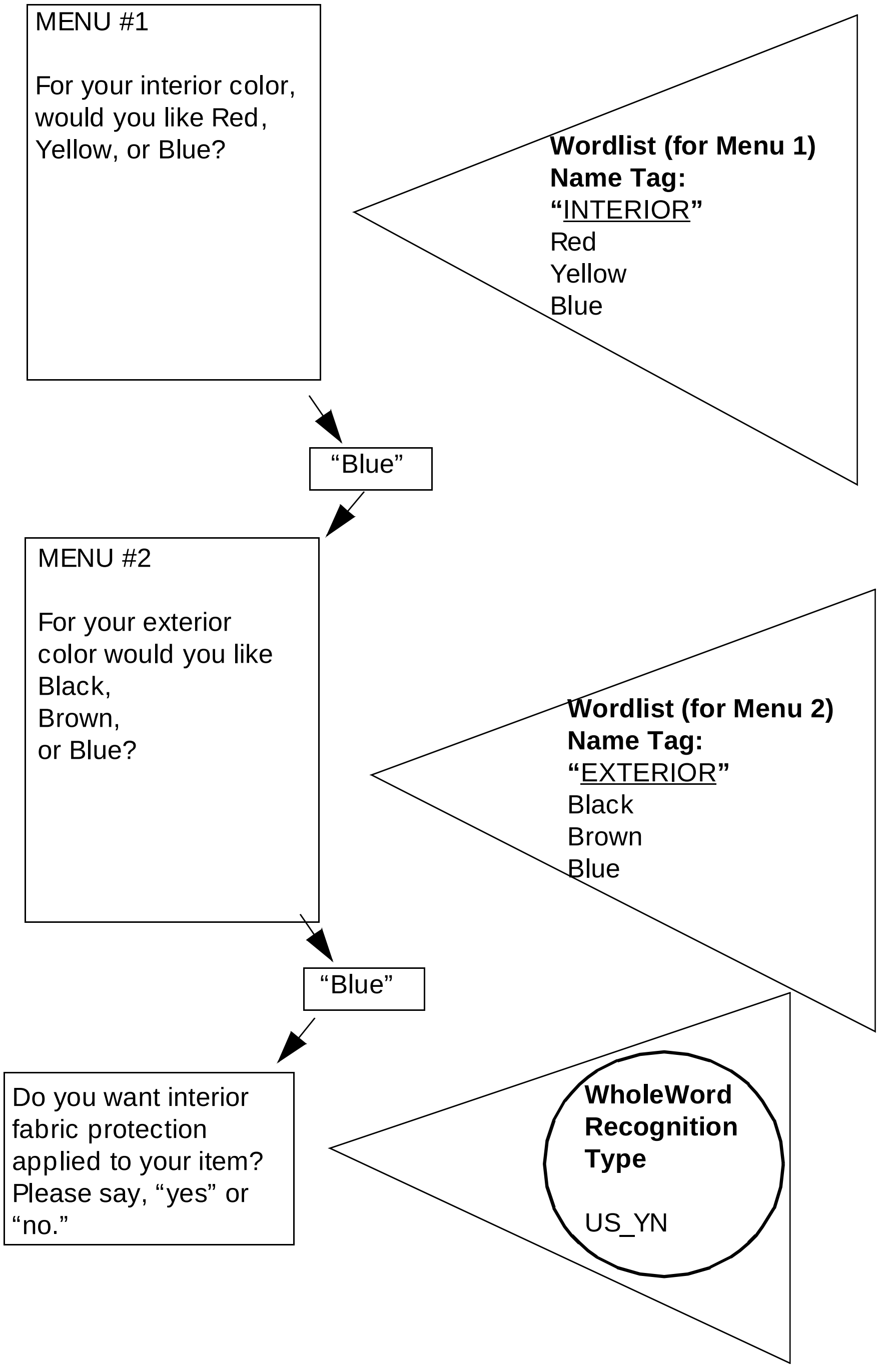

MENU #1
For your interior color, would you like Red, Yellow, or Blue?
Wordlist (for Menu 1)
Name Tag:
"INTERIOR"
Red
Yellow
Blue
"Blue"
MENU #2
For your exterior color would you like Black, Brown, or Blue?
Wordlist (for Menu 2)
Name Tag:
"EXTERIOR"
Black
Brown
Blue
"Blue"
Do you want interior fabric protection applied to your item? Please say, "yes" or "no."
WholeWord Recognition Type
US_YN

**Figure 7-2. Illustration of the Use of FlexWord and WholeWord Speech Recognition in a Single Application Script**

## WholeWord versus FlexWord Speech Recognition

Table 7-1 illustrates the similarities and differences between WholeWord and FlexWord speech recognition.

**Table 7-1. WholeWord versus FlexWord Speech Recognition**

| WholeWord Speech Recognition | FlexWord Speech Recognition |
|---|---|
| Word-based | Phoneme-based |
| Requires data collection in model building[1] | No data collection required in model building |
| Connected digits | Single word or phrase |
| Standard and custom grammars | Customized wordlists |
| Barge-in supported | Barge-in not supported |
| Phrase screening supported | Phrase screening supported |
| Limited vocabulary | 2000 word vocabulary |
| Word spotting supported | Word spotting supported |

[1] WholeWord speech recognition requires data collection to build new vocabulary words if you want words other than “zero” through “nine,” commonly used synonyms for those words, or the words “yes,” and “no.”

## WholeWord Speech Recognition versus Dial Pulse Recognition

Table 7-2 displays the similarities and differences between WholeWord speech recognition and Dial Pulse recognition.

**Table 7-2. WholeWord Speech Recognition versus Dial Pulse Recognition**

| WholeWord Speech Recognition | Dial Pulse Recognition |
|---|---|
| Word-based | Pulse-based |
| Requires data collection in model building[1] | No data collection required in model building |
| Connected digits | Connected digits |
| Standard and custom grammars | Standard grammars |
| Barge-in supported | Barge-in not supported |
| Phrase screening supported | Phrase screening supported |
| Limited vocabulary | Limited vocabulary |
| Word spotting supported | Word spotting not supported |

[1] WholeWord speech recognition requires data collection to build new vocabulary words if you want words other than “zero” through “nine,” commonly used synonyms for those words, or the words “yes,” and “no.”

# Enhanced Basic Speech Formats

# A

## Overview

This appendix describes a complete listing of the formats that are included in the Enhanced Basic Speech package for each language available.

## Purpose

The purpose of this appendix is to ensure that you can

- Build an application by using the Enhanced Basic Speech phrases included in this appendix
- Easily access Enhanced Basic Speech phrases for each language

## Languages Available

Enhanced Basic Speech is available in the following languages:

- Australian English (female)
- Brazilian Portuguese (female)
- Canadian French (female)
- Cantonese Chinese (female)
- Castilian Spanish (female)
- Dutch (female)
- French (female)
- German (female)
- Hindi (female)
- Japanese (female)
- Latin-American Spanish (female)
- Mandarin Chinese (female)
- UK English (female)
- US English (female and male)

## Enhanced Basic Speech Formats

This section provides a complete listing of all Enhanced Basic Speech formats. Three types of inflection exist with speech formats:

- Rising inflection is usually used in questions and at the beginning of some words.
- Medial inflection is usually used in the middle of a word or statement. For example, when you speak the number “101,” the “0” is spoken with 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.

## Speech Pool Names and Abbreviations

Table A-1 shows the Enhanced Basic Speech languages available and phrase names.

**Table A-1. Enhanced Basic Speech Languages**

| Enhanced Basic Speech Version 6.0 | Phrase Pool Name | Abbreviations |
|---|---|---|
| Australian English | AU_English | AE |
| Brazilian Portuguese | BR_Portug | BP |
| Canadian French | Can_French | CF |
| Cantonese Chinese | Cantonese | CT |
| Castilian Spanish | Castil_Span | CS |
| Dutch | Dutch | DT |
| French | French | FR |
| German | German | GR |
| Hindi | Hindi | HD |
| Japanese | Japanese | JP |
| Latin-American Spanish | LA_Spanish | LS |
| Mandrian | Mandrian | MD |
| UK English | UK_English | UE |
| US English | US_English | US |

## Australian English Enhanced Basic Speech Format Set

Table A-2 shows the Australian English Enhanced Basic Speech Formats.

**Table A-2. Australian English Enhanced Basic Speech Format Set**

| 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 is DMSPDY |
| TH24M | Time – 24-hour military time |
| THMAM | Time – hour, minute AM/PM |
| T | Time – default is THMAM |
| Nrmf | Number – total inflection |
| Nmmf | Number – falling inflection |
| Nrmm | Number – rising inflection |
| Nmmm | Number – no inflection |
| N | Number – default is Nmmm |
| ND1 | Number – with 1 digit after decimal point |
| ND2 | Number – with 2 digits after decimal point |
| ND3 | Number – with 3 digits after decimal point |
| ND4 | Number – with 4 digits after decimal point |
| ND5 | Number – with 5 digits after decimal point |
| ND6 | Number – with 6 digits after decimal point |
| ND7 | Number – with 7 digits after decimal point |
| ND8 | Number – with 8 digits after decimal point |
| ND9 | Number – with 9 digits after decimal point |

*Continued on next page*

**Table A-2. Australian English Enhanced Basic Speech Format Set — *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 is Cmmm |

## Brazilian Portuguese Enhanced Basic Speech Format Set

Brazilian Portuguese requires that numbers be spoken differently depending on whether the noun that the number refers to has a masculine, feminine, or neutral gender. Table A-3 shows Brazilian Portuguese formats which allow you to tell the system which form to use.

**Table A-3. Brazilian Portuguese 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, but no year |
| D | Date – default is 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 is 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 | Dec Number no decimal |
| ND1 | Dec Number 1 decimal |
| ND2 | Dec Number 2 decimals |

*Continued on next page*

**Table A-3. Brazilian Portuguese Enhanced Basic Speech Format Set — *Continued***

| Format | Description |
|---|---|
| ND3 | Dec Number 3 decimals |
| ND4 | Dec Number 4 decimals |
| ND5 | Dec Number 5 decimals |
| ND6 | Dec Number 6 decimals |
| ND7 | Dec Number 7 decimals |
| ND8 | Dec Number 8 decimals |
| ND9 | Dec Number 9 decimals |
| N$D2 | 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 is Cmmm |

## Canadian French Enhanced Basic Speech Format Set

Canadian French requires that numbers be spoken differently depending on whether the noun that the number refers to has a masculine, feminine, or neutral gender. Table A-4 shows Canadian French formats which allow you to tell the system which form to use.

**Table A-4. Canadian French 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, but no year |
| D | Date – default is 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 is 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 is 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 |
| TPS | PSTime – default is TH24M |

*Continued on next page*

**Table A-4. Canadian French Enhanced Basic Speech Format Set — *Continued***

| Format | Description |
|---|---|
| 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 is 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 is 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 is 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 is 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 is TH24M |
| THMAMAS | Atlantic Stand Time – hour, minute AM/PM |
| TH24MAS | Atlantic Stand Time – military time - no seconds |
| THMAMASf | ASTime – hour, minute AM/PM, falling inflection |

*Continued on next page*

**Table A-4. Canadian French Enhanced Basic Speech Format Set — *Continued***

| Format | Description |
|---|---|
| TH24MASf | ASTime – military time - no seconds, falling inflection |
| TAS | ASTime – default is 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 | Dec Number – no decimal, "virgule zero" |
| ND1 | Dec Number – 1 decimal |
| ND2 | Dec Number 2 decimals |
| ND3 | Dec Number  3 decimals |
| ND4 | Dec Number  4 decimals |
| ND5 | Dec Number  5 decimals |
| ND6 | Dec Number  6 decimals |
| ND7 | Dec Number  7 decimals |
| ND8 | Dec Number 8 decimals |
| ND9 | Dec Number 9 decimals |
| NDF0 | Dec Number  no decimal |
| NDF1 | Dec Number  1 decimal, decimal part spoken as number |
| NDF2 | Dec Number  2 decimals |
| NDF3 | Dec Number  3 decimals |
| NDF4 | Dec Number  4 decimals |
| NDF5 | Dec Number  5 decimals |
| NDF6 | Dec Number  6 decimals |
| NDF7 | Dec Number  7 decimals |
| NDF8 | Dec Number  8 decimals |

*Continued on next page*

**Table A-4. Canadian French Enhanced Basic Speech Format Set — *Continued***

| Format | Description |
|---|---|
| NDF9 | Dec Number 9 decimals |
| 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 one decimal place |
| N$F3 | Number – dollars with three decimal places |
| N$F4 | Number – dollars with four decimal places |
| N$F5 | Number – dollars with five decimal places |
| N$F6 | Number – dollars with six decimal places |
| N$F7 | Number – dollars with seven decimal places |
| N$F8 | Number – dollars with eight decimal places |
| N$F9 | Number – dollars with nine 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 is Cmmm |

## Cantonese Chinese Enhanced Basic Speech Format Set

For US currency the word “maygum” is spoken first to indicate United States currency. Table A-5 shows the Cantonese Chinese Enhanced Basic Speech Formats.

**Table A-5. Cantonese Chinese Enhanced Basic Speech Format Set**

| Format | Description |
|---|---|
| DYMDD | Date – 4-digit year, month and day |
| DMD | Date month followed by day |
| D | Date – default is DTTMD |
| THMAM | Time – same as T |
| T | Time – time, period followed by time |
| N | Number |
| ND0 | Number – 0 digits after decimal point |
| ND1 | Number – 1 digit after decimal point |
| ND2 | Number – 2 digits after decimal point |
| ND3 | Number – 3 digits after decimal point |
| ND4 | Number – 4 digits after decimal point |
| ND5 | Number – 5 digits after decimal point |
| ND6 | Number – 6 digits after decimal point |
| ND7 | Number – 7 digits after decimal point |
| ND8 | Number – 8 digits after decimal point |
| ND9 | Number – 9 digits after decimal point |
| N$D0 | Number – whole dollars |
| N$D2 | Number – dollars and cents |
| N$D1 | Number – dollars 1 decimal place |
| N$D3 | Number – dollars 3 decimal places |
| N$D4 | Number – dollars 4 decimal places |
| N$D5 | Number – dollars 5 decimal places |
| N$D6 | Number – dollars 6 decimal places |

*Continued on next page*

**Table A-5. Cantonese Chinese Enhanced Basic Speech Format Set — *Continued***

| Format | Description |
|---|---|
| N$D7 | Number – dollars 7 decimal places |
| N$D8 | Number – dollars 8 decimal places |
| N$D9 | Number – dollars 9 decimal places |
| NYD0 | Number – whole yuan |
| NYD1 | Number – yuan 1 decimal place |
| NYD2 | Number – yuan, miao and seen |
| NYD3 | Number – yuan 3 decimal places |
| NYD4 | Number – yuan 4 decimal places |
| NYD5 | Number – yuan 5 decimal places |
| NYD6 | Number – yuan 6 decimal places |
| NYD7 | Number – yuan 7 decimal places |
| NYD8 | Number – yuan 8 decimal places |
| NYD9 | Number – yuan 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 Format Set

Castilian Spanish requires that numbers be spoken differently depending on whether the noun that the number refers to has a masculine, feminine, or neutral gender. Table A-6 shows Castilian Spanish formats which allow you to tell the system which form to use.

**Table A-6. Castilian 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-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 – 0 digits after decimal point |
| ND1 | Number – 1 digit after decimal point |
| ND2 | Number – 2 digits after decimal point |
| ND3 | Number – 3 digits after decimal point |
| ND4 | Number – 4 digits after decimal point |
| ND5 | Number – 5 digits after decimal point |
| ND6 | Number – 6 digits after decimal point |
| ND7 | Number – 7 digits after decimal point |
| ND8 | Number – 8 digits after decimal point |
| ND9 | Number – 9 digits after decimal point |
| N$ | Number – whole pesetas |

*Continued on next page*

**Table A-6. Castilian Spanish Enhanced Basic Speech Format Set — *Continued***

| Format | Description |
|---|---|
| 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 Format Set

Table A-7 shows the Dutch Enhanced Basic Speech Formats.

**Table A-7. Dutch 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, but no year |
| D | Date – default is DMSPDYY |
| TH24M | Time – military time - no seconds |
| T | Time – default is TH24M |
| N | Number – default is NN |
| ND0 | Number – with 0 digits after decimal point |
| ND1 | Number – with 1 digit after decimal point |
| ND2 | Number – with 2 digits after decimal point |
| ND3 | Number – with 3 digits after decimal point |
| ND4 | Number – with 4 digits after decimal point |
| ND5 | Number – with 5 digits after decimal point |
| ND6 | Number – with 6 digits after decimal point |
| ND7 | Number – with 7 digits after decimal point |
| ND8 | Number – with 8 digits after decimal point |
| ND9 | Number – with 9 digits after decimal point |
| 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 Format Set — *Continued***

| **Format** | **Description** |
|---|---|
| Cmmf | Characters – spoken individually; falling inflection |
| Cmmm | Characters – spoken individually; no inflection |
| C | Characters – default is Cmmm |

## French Enhanced Basic Speech Format Set

French requires that numbers be spoken differently depending on whether the noun that the number refers to has a masculine, feminine, or neutral gender. Table A-8 shows French formats which allow you to tell the system which form to use.

**Table A-8. French 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, but no year |
| D | Date – default is 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 is 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 | Dec Number – no decimal, "virgule zero" |
| ND1 | Dec Number – 1 decimal |

*Continued on next page*

**Table A-8. French Enhanced Basic Speech Format Set — *Continued***

| Format | Description |
|---|---|
| ND2 | Dec Number – 2 decimals |
| ND3 | Dec Number – 3 decimals |
| ND4 | Dec Number – 4 decimals |
| ND5 | Dec Number – 5 decimals |
| ND6 | Dec Number – 6 decimals |
| ND7 | Dec Number – 7 decimals |
| ND8 | Dec Number – 8 decimals |
| ND9 | Dec Number – 9 decimals |
| NDF0 | Dec Number – no decimal |
| NDF1 | Dec Number – 1 decimal, decimal part spoken as number |
| NDF2 | Dec Number – 2 decimals |
| NDF3 | Dec Number – 3 decimals |
| NDF4 | Dec Number – 4 decimals |
| NDF5 | Dec Number – 5 decimals |
| NDF6 | Dec Number – 6 decimals |
| NDF7 | Dec Number – 7 decimals |
| NDF8 | Dec Number – 8 decimals |
| NDF9 | Dec Number – 9 decimals |
| N$D2 | Number – francs and centimes |
| N$ | Number – francs and centimes |
| N$F | Number – whole number of francs |
| N$F0 | Number of francs  0 decimal |
| N$F1 | Number of francs  1 decimal |
| N$F2 | Number of francs  2 decimals |
| N$F3 | Number of francs  3 decimals |
| N$F4 | Number of francs  4 decimals |
| N$F5 | Number of francs  5 decimals |
| N$F6 | Number of francs  6 decimals |

*Continued on next page*

**Table A-8. French Enhanced Basic Speech Format Set — *Continued***

| Format | Description |
|---|---|
| N$F7 | Number of francs  7 decimals |
| N$F8 | Number of francs  8 decimals |
| N$F9 | Number of francs  9 decimals |
| 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 is Cmmm |

## German Enhanced Basic Speech Format Set

German requires that numbers be spoken differently depending on whether the noun that the number refers to has a masculine, feminine, or neutral gender. Table A-9 shows German formats which allow you to tell the system which form to use.

**Table A-9. German 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 |
| TH24M | Time – military time-no seconds |
| T | Time – default is THMAM |
| NM | Number – following masculine noun |
| NF | Number – following feminine noun |
| NN | Number – following neutral noun |
| N | Number – default NN |
| ND0 | Number – 0 digits after decimal point |
| ND1 | Number – 1 digit after decimal point |
| ND2 | Number – 2 digits after decimal point |
| ND3 | Number – 3 digits after decimal point |
| ND4 | Number – 4 digits after decimal point |
| ND5 | Number – 5 digits after decimal point |
| ND6 | Number – 6 digits after decimal point |
| ND7 | Number – 7 digits after decimal point |
| ND8 | Number – 8 digits after decimal point |
| ND9 | Number – 9 digits after decimal point |
| N$ | Number – whole marks |
| N$D0 | Number – whole marks |

*Continued on next page*

**Table A-9. German Enhanced Basic Speech Format Set — *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 Format Set

Table A-10 shows the Hindi Enhanced Basic Speech Formats.

**Table A-10. Hindi Enhanced Basic Speech Format Set**

| 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 – 0 digits after decimal point |
| ND1 | Number – 1 digit after decimal point |
| ND2 | Number – 2 digits after decimal point |
| ND3 | Number – 3 digits after decimal point |
| ND4 | Number – 4 digits after decimal point |
| ND5 | Number – 5 digits after decimal point |
| ND6 | Number – 6 digits after decimal point |
| ND7 | Number – 7 digits after decimal point |
| ND8 | Number – 8 digits after decimal point |
| ND9 | Number – 9 digits after decimal point |
| N$D0 | Number – whole dollars |
| N$D1 | Number – dollars 1 decimal place |
| N$D2 | Number – dollars and cents |
| N$D3 | Number – dollars 3 decimal places |
| N$D4 | Number – dollars 4 digits decimal places |
| N$D5 | Number – dollars 5 digits decimal places |
| N$D6 | Number – dollars 6 digits decimal places |
| N$D7 | Number – dollars 7 digits decimal places |

*Continued on next page*

**Table A-10. Hindi Enhanced Basic Speech Format Set — *Continued***

| Format | Description |
|---|---|
| N$D8 | Number – dollars 8 digits decimal places |
| N$D9 | Number – dollars 9 digits decimal places |
| NRD0 | Number – rupees |
| NRD1 | Number – rupees 1 decimal place |
| NRD2 | Number – rupees 2 decimal places |
| NRD3 | Number – rupees 3 decimal places |
| NDR4 | Number – rupees 4 decimal places |
| NDR5 | Number – rupees 5 decimal places |
| NDR6 | Number – rupees 6 decimal places |
| NDR7 | Number – rupees 7 decimal places |
| NDR8 | Number – rupees 8 decimal places |
| NDR9 | Number – rupees 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 |

## Japanese Enhanced Basic Speech Format Set

Table A-11 shows the Japanese Enhanced Basic Speech Formats.

**Table A-11. Japanese Enhanced Basic Speech Format Set**

| 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 | Number – 0 digits after decimal point |
| ND1 | Number – 1 digit after decimal point |
| ND2 | Number – 2 digits after decimal point |
| ND3 | Number – 3 digits after decimal point |
| ND4 | Number – 4 digits after decimal point |
| ND5 | Number – 5 digits after decimal point |
| ND6 | Number – 6 digits after decimal point |
| ND7 | Number – 7 digits after decimal point |
| ND8 | Number – 8 digits after decimal point |
| ND9 | Number – 9 digits after decimal point |
| 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

Latin-American Spanish requires that numbers be spoken differently depending on whether the noun that the number refers to has a masculine, feminine, or neutral gender. Table A-12 shows Latin-American Spanish formats which allow you to tell the system which form to use.

**Table A-12. 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 – 0 digits after decimal point |
| ND1 | Number – 1 digit after decimal point |
| ND2 | Number – 2 digits after decimal point |
| ND3 | Number – 3 digits after decimal point |
| ND4 | Number – 4 digits after decimal point |
| ND5 | Number – 5 digits after decimal point |
| ND6 | Number – 6 digits after decimal point |
| ND7 | Number – 7 digits after decimal point |
| ND8 | Number – 8 digits after decimal point |
| ND9 | Number – 9 digits after decimal point |
| Cm | Character – default Cm |

*Continued on next page*

**Table A-12. Latin-American Spanish Enhanced Basic Speech Format Set — *Continued***

| Format | Description |
|---|---|
| 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 |
| NSOL$D0 | Number – whole soles |

*Continued on next page*

**Table A-12. Latin-American Spanish Enhanced Basic Speech Format Set — *Continued***

| Format | Description |
|---|---|
| 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. |

## Mandrian Enhanced Basic Speech Format Set

Table A-13 shows the Mandrian Enhanced Basic Speech Formats.

**Table A-13. Mandrian Enhanced Basic Speech Format Set**

| 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 | Number - 0 digits after decimal point |
| ND1 | Number - 1 digit after decimal point |
| ND2 | Number - 2 digits after decimal point |
| ND3 | Number - 3 digits after decimal point |
| ND4 | Number - 4 digits after decimal point |
| ND5 | Number - 5 digits after decimal point |
| ND6 | Number - 6 digits after decimal point |
| ND7 | Number - 7 digits after decimal point |
| ND8 | Number - 8 digits after decimal point |
| ND9 | Number - 9 digits after decimal point |
| 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 |

*Continued on next page*

**Table A-13. Mandrian**
**Enhanced Basic Speech Format Set**

| | |
|---|---|
| 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, 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 is Cmmm |

## UK English Enhanced Basic Speech Format Set

Table A-14 shows the UK English Enhanced Basic Speech Formats.

**Table A-14. UK English Enhanced Basic Speech Format Set**

| 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 is DMSPDY |
| THMAM | Time – hour, minute AM/PM |
| T | Time – default is THMAM |
| Nrmf | Number – total inflection |
| Nmmf | Number – falling inflection |
| Nrmm | Number – rising inflection |
| Nmmm | Number – no inflection |
| N | Number – default Nmmm |
| ND1 | Number – 1 digit after decimal point |
| ND2 | Number – 2 digits after decimal point |
| ND3 | Number – 3 digits after decimal point |
| ND4 | Number – 4 digits after decimal point |
| ND5 | Number – 5 digits after decimal point |
| ND6 | Number – 6 digits after decimal point |
| ND7 | Number – 7 digits after decimal point |
| ND8 | Number – 8 digits after decimal point |
| ND9 | Number – 9 digits after decimal point |
| N$ | Number – whole pounds |

*Continued on next page*

**Table A-14. UK English Enhanced Basic Speech Format Set — *Continued***

| Format | Description |
|---|---|
| N$D0 | Number – whole pounds |
| N$D2 | Number – pounds and pence |
| N$D1 | Number – pounds 1 decimal place |
| N$D3 | Number – pounds 3 decimal places |
| N$D4 | Number – pounds 4 decimal places |
| N$D5 | Number – pounds 5 decimal places |
| N$D6 | Number – pounds 6 decimal places |
| N$D7 | Number – pounds 7 decimal places |
| N$D8 | Number – pounds 8 decimal places |
| N$D9 | Number – pounds 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 Format Set

Table A-15 shows the US English Enhanced Basic Speech Formats.

**Table A-15. US English Enhanced Basic Speech Format Set**

| 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 is DMSPDY |
| THMAM | Time – hour, minute AM/PM |
| T | Time – default is THMAM |
| Nrmf | Number – total inflection |
| Nmmf | Number – falling inflection |
| Nrmm | Number – rising inflection |
| Nmmm | Number – no inflection |
| N | Number – default Nmmm |
| ND1 | Number – 1 digit after decimal point |
| ND2 | Number – 2 digits after decimal point |
| ND3 | Number – 3 digits after decimal point |
| ND4 | Number – 4 digits after decimal point |
| ND5 | Number – 5 digits after decimal point |
| ND6 | Number – 6 digits after decimal point |
| ND7 | Number – 7 digits after decimal point |
| ND8 | Number – 8 digits after decimal point |
| ND9 | Number – 9 digits after decimal point |
| N$ | Number – whole dollars |

*Continued on next page*

**Table A-15. US English Enhanced Basic Speech Format Set — *Continued***

| **Format** | Description |
|---|---|
| N$D0 | Number – whole dollars |
| N$D2 | Number – dollars and cents |
| N$D1 | Number – dollar 1 decimal place |
| N$D3 | Number – dollar 3 decimal places |
| N$D4 | Number – dollar 4 decimal places |
| N$D5 | Number – dollar 5 decimal places |
| N$D6 | Number – dollar 6 decimal places |
| N$D7 | Number – dollar 7 decimal places |
| N$D8 | Number – dollar 8 decimal places |
| N$D9 | Number – dollar 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 information about speech files and speech file formats.

## 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:

- Adaptive Differential Pulse Code Modulation (ADPCM) at 32 kbps
- CELP 16
- Pulse Code Modulation (PCM) at 64 kbps in Mu-law encoding format
- Sub Band Coding (SBC) at 16 kbps (not commonly used in system applications)
- SBC at 24 kbps (not commonly used in system applications)
- ADPCM at 16 kbps (not commonly used in system applications)

## PCM Speech File 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 ten 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 quasilogarithmic steps for speech levels, encoding more steps when the speech level is low.

## ADPCM Speech File 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:

- ADPCM-32
- ADPCM-16

Both of the formats follow similar compression algorithms.

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 then 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 (or the 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 ADPCM 32 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 ADPCM 16 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 may 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.

## About Speech Files

Speech files are comprised of two parts:

- A header section of the format shown in Figure B-1 which 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. Header Section**

The following rules apply to 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* is a mandatory field that 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:**
Code and playback of CELP16 works properly only on the hardware that supports it such as the SSP, IVC6, or NGTR circuit card. If either operation (coding or playback) is performed on an unsupported circuit card, the result will be 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 CELP16 is played on a circuit card that supports CELP16, the result is hearing the phrase being played.
- 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 O.S. Index

# C

## Overview

This appendix describes the procedure for computing channel numbers for the Graphical Speech Editor (GSE).

## Purpose

The purpose of this appendix is to ensure that you are able to accurately and correctly calculate the Operating System (O.S.) Index.

# Calculating the O.S. Index

The Graphical Speech Editor command **gse [-l<chan>] | [-p<chan>-r<chan>]** requires the identification of a channel number (chan) for each respective channel designation (-l, -p, and/or -r). The Graphical Speech Editor does not recognize the same language as the Unix system. Therefore, the information that the UnixWare system supplies is not understood by the Graphical Speech Editor. However, by conducting some calculations using the information supplied by the UnixWare system, we can determine the Operating System (O.S.) Index which the Graphical Speech Editor can understand.

To determine the appropriate channel numbers for the l, p, and r channel designations on the Tip/Ring circuit card, perform the following procedure:

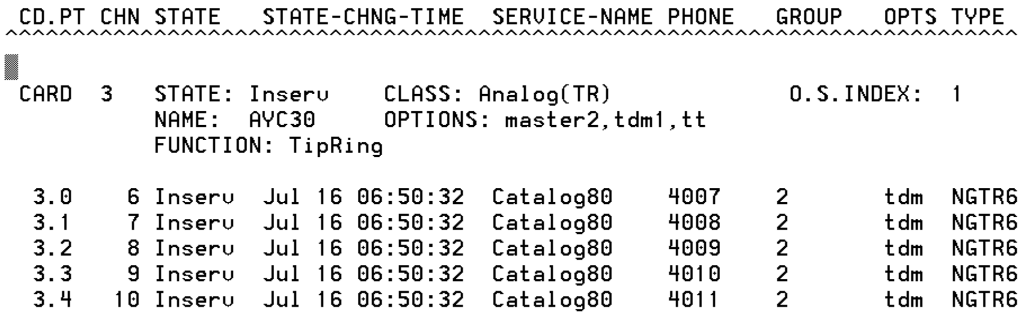
1. Identify the system channel that you want to use before accessing the system.
   For the sake of this exercise, assume #7 to be the identified channel.
2. Log in to the system by performing the following:
   a. Enter root at the system prompt.

      The system displays the password prompt.

   b. Enter your valid password.

      The system displays the UnixWare system prompt (#).
3. Display the present channel settings on the Tip/Ring card by entering **disp ca tr**

   The system displays the 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 for the card on which the identified channel is located. In this example, Channel #7 (under `CHN` — second column) is located on Card 1 which has “1” as its O.S. Index number.
5. Find the channel offset. This number ranges from 0 through 5 and is the one digit suffix listed under `CD.PT` (first column). The offset for channel 7 is 1.
6. Use the following formula to compute the O.S. Index for the channel.

   Channel O.S. Index = 6 x (card’s os index) + offset

   In the example, the O.S. Index for channel 7 is 6 x 1 + 1 = 7.

This O.S. Index can now be used as an argument when invoking GSE.

# Advanced Text-to-Speech Features

# D

## Overview

This appendix describes several ways to customize synthesized speech by including escape sequences.

## Purpose

The purpose of this chapter is to ensure that you know how to use escape sequences and how to change class detector modes.

## Introduction

With some Text-To-Speech (TTS) applications, you may need to further customize the use of synthesized speech by adding silence delays, changing the speaking rate, or by marking text as members of a more specific text category. While you can alter the way in which TTS speaks a phrase, some applications require that you use escape sequences placed before, after, or within the spoken text.

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

Not all applications require these advanced features. If you have questions about the use of advanced TTS features, call the Customer Support Hotline at 1-800-242-2121.

**CAUTION:**
*Use the escape sequences exactly as they are described in this appendix. Using them differently could create problems in your application.*

## Using Escape Sequences

You can alter the manner in which TTS speaks a phrase within the text by using specific escape sequences. There are several methods for adding escape sequences to your text. One method is to edit your text directly. Another method is to concatenate the specific escape sequence to your text string. Examples of these two methods are provided at the end of this section.

### Silence Delays

There are several ways to introduce silent pauses or delays during the speaking of synthesized text. Most application silence needs can be handled by using the methods described in “Hints for Writing Applications Using Text-to-Speech” in Chapter 8, “Using Optional Features,” of *INTUITY*™ *CONVERSANT® System Version 6.0 Application Development with Script Builder,* 585-310-760. However, if your application requires more advanced silence delays, you can use an escape sequence. With escape sequences, silence delays can be added to the beginning, end, or within a text phrase. The application developer indicates the silence duration in centiseconds (one hundredth of a second).

**NOTE:**
The escape sequences shown in Table D-1 are used for silence delays on a system having an SP or an SSP circuit card.

**CAUTION:**
Using values outside of the limits could cause problems in your application.

**Table D-1. Escape Sequences for Silence Delays**

| Escape Sequence | Definition | Example |
|---|---|---|
| \!si < *n* > | For *n* centiseconds of silence at the beginning of a phrase, where 1< *n* < 100. This escape sequence can only be used at the beginning of a text line, as it is used in the example where 0.7 seconds of silence occurs before the word "Hello" is spoken. The maximum allowance is 100 centiseconds. | `Announce`<br>`Speak with Interrupt`<br>`Text: "\!si70 Hello! Welcome to your Home Shopping Line!"` |
| \!sf < *n* > | For *n* centiseconds of silence at the end of a phrase, where 1< *n* < 100. This escape sequence can only be used at the beginning of a text line, as it is used in the example where 0.5 seconds of silence occurs after the first sentence is spoken but before the second sentence is spoken. The maximum allowance is 100 centiseconds. | `Announce`<br>`Speak with Interrupt`<br>`Text: "\!sf50 Hello, welcome to your Home Shopping Line!`<br>`Text: For information about this service, press 1.`<br>`Text: To start your order, press 2.` |
| \(**[<n>]\) | For *n* centiseconds of silence within a phrase, where 1< *n* < 60. In the example, 50 centiseconds of silence occurs after the word "Hello" is spoken and before the word "and" is spoken. The maximum allowance is 60 centiseconds. | `Announce`<br>`Speak with Interrupt`<br>`Text: "Hello \(**[50]\) and`<br>`Text:  welcome to your`<br>`Text: Home Shopping Line!"` |

## Speaking Rate on an SP Circuit Card

You can control the rate of the synthesized voice through the use of an escape sequence. Five rates are available, from 0 (fastest) to 4 (slowest). The default rate is 2. Each rate setting is effective until the end of the sentence (text ending with punctuation such as a period, question mark, or exclamation mark). To change an escape sequence before the end of a sentence, you simply re-enter another escape sequence. The escape sequences shown in Table D-2 are used to change the speaking rate on a system having an SP circuit card.

**NOTE:**
Table D-2 sequences will *not* work on an SSP circuit card running TTS.

**Table D-2. Escape Sequences for Speaking Rate On Systems with an SP Circuit Card**

| Escape Sequence | Definition | Example |
|---|---|---|
| \!r 0 (fastest)<br>\!r 1<br>\!r 2<br>\!r 3<br>\!r 4 (slowest) | In the example, the slower rate is in effect only while the order number is spoken. The first sentence, up to the rate change, and the second sentence are spoken with the default speaking rate. | `Announce`<br>`Speak with Interrupt`<br>`Text: Your order number is \!r4 104235\!r2 .”`<br>`Text: Please use this number when checking on your order.”` |

## Speaking Rate on an SSP Circuit Card

A second grouping of escape sequences are available for systems running with an SSP circuit card. Similar to the above, five rates are available. They range from “fastest” to “slowest”. The default rate is “normal”. Each rate setting is effective until the end of the sentence (text ending with punctuation such as a period, question mark, or exclamation mark). To change an escape sequence before the end of a sentence, you simply re-enter another escape sequence. The escape sequences shown in Table D-3 are used to change the speaking rate on a system having an SSP circuit card.

**NOTE:**
Table D-3 sequences will *not* work on an SP circuit card running TTS.

**Table D-3. Escape Sequences for Speaking Rate on Systems with an SSP Circuit Card**

| Escape Sequence | Definition | Example |
|---|---|---|
| \!rfastest<br>\!rfast<br>\!rnormal<br>\!rslow<br>\!rslowest | In the example, the slower rate is in effect only while the order number is spoken. The first sentence, up to the rate change, and the second sentence are spoken with the default speaking rate. | `Announce`<br>`Speak with Interrupt`<br>`Text: “Your order number is \!rslowest 104235\!rnormal.`<br>`Text: Please use this number when checking on your order.”` |

## Additional Text Classes

With the Prompt & Collect and Announce action steps, you can indicate the format of the text to be spoken. The text can be marked to be spoken in the following formats:

- Character
- Number
- Time
- Date
- Dollar amounts

Some applications may require a further classification of the text to be spoken. The following additional class types are available for further classifications:

- Addresses
- Telephone numbers
- Fractions
- Proper names

To speak text from the above additional class types, you must activate one of the following *class detector*s for a specific text class.

- Off
- Conservative (normal mode)
- Risky (high-priority mode)

Initially, all the class detectors are set in the conservative mode. Class detector escape sequences are not embedded in or connected to other text to be activated. Change the mode by using an "Announce" action step and speaking only the escape sequence. The class detector remains in the new state until it is changed or until the script ends.

The following example demonstrates when the off detection mode is used. The system is trying to speak "Today's date is 1/4" in conservative mode, but the system is failing to do so correctly. This is because the system can not distinguish if the "1/4" is a fraction or possibly a date. To solve the problem, turn conservative fraction mode off and continue running the conservative date to have this spoken as a date.

**NOTE:**
Use Risky mode only after the system fails to pronounce a word or phrase correctly. Always try conservative mode first.

The system has to determine how to classify text and abbreviations transmitted from a database. If the text or abbreviations are applicable to more than one class, use the risky mode. For example, separate fields of information containing a person's name, street address, and city, state, and zip code are given to the system from a database in the following form:

```
Columbus, OH  43213

Bob Aurt

22587 S. Hamilton St.
```

In this case, the system may not know whether to classify the "22587" as an address or a number because it is a separate field from the rest and it could be both. Risky mode can alleviate this confusion. If you want the system to classify this information as an address, turn on risky address by entering " `\!nar` ". If you want the system to classify this information as a number, turn on risky number and speak "`\!nar`" by using the Announce mode. The following shows these steps.

```
Announce
  Speak with Interrupt
   Text: “\!nar“
```

Risky mode allows a priority to be set to a designated classification to avoid confusing the system on how the information is to be classified. The setting of risky mode, such as risky address, will be the highest priority at the time for the unidentified field. The system will then know that "22587" is part of an address.

Conservative mode requires more contextual information to resolve ambiguous abbreviations. Abbreviations such as "Sr." and "Jr." as in "John Paul Sr." or "John Paul Jr." will expand correctly in risky name mode or conservative mode because there is only one possible interpretation of the abbreviations.

Here are the steps to take to ensure the system speaks words or phrases correctly:

1. Listen to what the system is speaking.
2. Re-enter the information to correct the pronunciation. Sometimes this may require spelling a field incorrectly so that the machine pronounces it correctly. For example, you may have to change the spelling of "machine" to "masheen" for the system to pronounce it correctly.
3. Listen again and continue with this process until you are pleased with the way the system speaks the particular field.

## Changing Class Detection

You can change text class detection modes using escape sequences with the following general format:

```
\!n < c >< m >
```

The *c* variable is one of the following text classes:

- a (Address)
  Use risky mode to detect partial addresses and to correctly pronounce ambiguous address abbreviations, such as "Dr." for "Drive " correctly.

  **NOTE:**
  Standard post office abbreviations apply.

- f (Fraction)
  Use risky mode so that, for example, "1/4" is pronounced as "one fourth."
- n (Proper name)
  Use risky mode so that titles such as "Mr." for "Mister" or ambiguous proper name abbreviations such as "Dr." for "Doctor" are pronounced correctly.
  - — An example of a proper name that may cause confusion is "John Paul III". If entered into the system this way the system will pronounce it as three I's. For the system to pronounce this field successfully, you can either enter "John Paul third" or "John Paul 3rd". This can be accomplished in either conservative or risky mode because the "3rd or third" cannot be misinterpreted.
- p (Telephone numbers)
  Use risky mode to ensure that "vanity" telephone numbers such as 1 (800) CALL ATT are spelled out.

The *m* variable is one of the three possible class detector mode settings for the specified text class:

- o (turn off the class detector)
- c (set the class detector to conservative mode — default)
- r (set the class detector to risky mode)

### Setting Specific Class Detector Modes

This section includes examples of setting specific class detector modes.

- Proper name detector

  `\!nnr` (Set the proper name detector to risky mode)

  `\!nnc` (Set the proper name detector to conservative mode — default)

  `\!nno` (Set the proper name detector to off)

  The example that follows shows setting the proper name detector to risky mode, speaking a customer name from a database field, and then setting the proper name detector back to conservative mode. The risky mode setting is needed only if your data is known to have name-related abbreviations or titles. Otherwise, use the default mode.

```
Announce
  Speak with Interrupt
   Text:  “\!nnr”
   Text:  “Your name is“
  Field:   customer_name as Bob Aurt
   Text:  “\!nnc“
```

- Telephone number detector

  `\!npr` (Set the telephone detector to risky mode)

  `\!npc` (Set the telephone detector to conservative mode — default)

  `\!npo` (Set the telephone number detector to off)

  The example that follows shows setting the telephone number detector to risky mode, speaking a “vanity” telephone number, and turning the telephone number 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 detector

  `\!nar` (Set the address detector to risky mode)

  `\!nac` (Set the address detector to conservative mode — default)

  `\!nao` (Set the address detector to off)

  The example that follows shows setting the address detector to risky mode, speaking out an address from a database field, and then turning the address detector to conservative mode. Risky mode needs to be used at the beginning to rank possibilities of the address so that "CT" does not get interpreted as "court" or "count," but as Connecticut. Lastly, 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 6200 E. Broad St.
 Field:  customer_state as CT
  Text: “\!nac“
```

## Sample Application Using Escape Sequences

The following is a sample Script Builder application that uses escape sequences.

```
1.   Answer telephone
2.   Set Field Value
           Field: ADDRESS_RISKY_MODE = “\!nar”
           Field: ADDRESS_DEFAULT_MODE = “\!nac“
           Field: PHONE_RISKY_MODE = “\!npr“
           Field: PHONE_DEFAULT_MODE = “\!npc“
           Field: NAME_RISKY_MODE = “\!nnr“
           Field: NAME_DEFAULT_MODE = “\!nnc“
           Field: BEG_SILENCE = “\!si70“
           Field: MIDDLE_SILENCE = “\!(*[50])”
           Field: END_SILENCE = “\!sf70“
           Field: SPEAKING_RATE_FOUR = “\!r4“
3.   Announce
        Speak With Interrupt
        Text: “Hello! Welcome to your home shopping line!“
4.   Prompt & Collect
        Prompt
           Speak With Interrupt
       Text: “For information about this service press 1.”
       Text: “To start your order, press 2.“
        Input
           Max Number Of Digits: 01
        Checklist
           Case: “1”
              Goto SPEAK_INFO
           Case: “2”
```

```
              Goto TAKE_ORDER
           Case: “Not On List”
              Reprompt
           Case: “Initial Timeout”
              Reprompt
           Case: “Too Few Digits”
              Reprompt
           Case: “No More Tries”
              Quit
     End Prompt & Collect
     SPEAK_INFO:
5.   External Function
        Function Name: tts_file
        Use Arguments: 1  “/att/trans/sb/shopper/text”
6.   Prompt & Collect
        Prompt
           Speak With Interrupt
              Text: “To start your order, press 1.“
              Text: “To complete this call, press 2.“
        Input
           Max Number Of Digits: 01
        Checklist
           Case: “1”
              Goto INITIATE_ORDER
           Case: “2”
              Goto GOODBYE
           Case: “Not On List”
              Reprompt
           Case: “Initial Timeout”
              Reprompt
           Case: “Too Few Digits”
              Reprompt
           Case: “No More Tries”
              Quit
     End Prompt & Collect
     INITIATE_ORDER:
7.   Prompt & Collect
        Prompt
           Speak With Interrupt
              Text: “Using touch tones, please enter your
                     5-digit customer account number.“
        Input
           Min Number Of Digits: 05
           Max Number Of Digits: 05
        Checklist
           Case: “Input Ok”
              Continue
           Case: “Initial Timeout”
              Reprompt
           Case: “Too Few Digits”
              Reprompt
           Case: “No More Tries”
              Quit
     End Prompt & Collect
```

```
8.   Read Table
        Table Name:  cust_db     Search From Beginning
           Field: account_number = $CI_VALUE
9.   Evaluate
     If $MATCH_FOUND != 0
10.     Prompt & Collect
           Prompt
              Speak With Interrupt
                 Field: NAME_RISKY_MODE As A
                 Text: “Your name is: “
                 Field: customer_name As A
                 Field: NAME_DEFAULT_MODE As A
                 Text: “Press 1 for yes, 2 for no.“
                            Input
              Max Number Of Digits: 01
           Checklist
              Case: “1”
                 Goto CHECK_ADDRESS
              Case: “2”
                 Goto TRANSFER_CALL
              Case: “Not On List”
                 Reprompt
              Case: “Initial Timeout”
                 Reprompt
              Case: “Too Few Digits”
                 Reprompt
              Case: “No More Tries”
                 Quit
        End Prompt & Collect
     Else
11.     Goto INITIATE_ORDER
     End Evaluate
     CHECK_ADDRESS:
     ### Verify the caller’s address
12.  Prompt & Collect
        Prompt
           Speak With Interrupt
              Field: ADDRESS_RISKY_MODE As A
              Text: “Your address is “
              Field: customer_address As A
              Phrase: “sil.500”
              Field: customer_city As A
              Field: NAME_DEFAULT_MODE As A
              Text: “Press 1 for yes, 2 for no.“
        Input
           Max Number Of Digits: 01
        Checklist
           Case: “1”
              Goto TAKE_ORDER
           Case: “2”
              Goto TRANSFER_CALL
           Case: “Not On List”
              Reprompt
           Case: “Initial Timeout”
```

```
              Reprompt
           Case: “Too Few Digits”
              Reprompt
           Case: “No More Tries”
              Quit
     End Prompt & Collect
     TAKE_ORDER:
13.  Prompt & Collect
        Prompt
           Speak With Interrupt
        Text: “Enter the item number, or the pound
key to complete your order.”
        Text Input:
           Max Number Of Digits: 05
        Checklist
           Case: “#”
              Goto WRAPUP_ORDER
           Case: “nnnnn”
              Continue
           Case: “Not On List”
              Reprompt
           Case: “Initial Timeout”
              Reprompt
           Case: “Too Few Digits”
              Reprompt
           Case: “No More Tries”
              Quit
     End Prompt & Collect
14.  Read Table
        Table Name:  grocery_db     Search From Beginning
           Field: item_number = $CI_VALUE
15.  Evaluate
     If $MATCH_FOUND   != 0
        ### Speak the item description and cost
16.     Announce
           Speak With Interrupt
              Field: item_description As A
              Text: “for“
              Field: item_price As AN$D2
        ### Add the item to the customer’s order
17.     Modify Table
           Table Name: order_db Operation: Add
              Field: order_number = account_number
              Field: order_item = item_number
18.     Goto TAKE_ORDER
     End Evaluate
     WRAPUP_ORDER:
19.  External Function
        Text:  “Your order number is: “
        Field: SPEAKING_RATE_FOUR as A
        Field: order_number as A
        Text:  “ .   “
        Text:  “Please use this number when checking your
order.“
```

```
     GOODBYE:
20.  Announce
        Speak With Interrupt
          Text: “Thanks for using the Home Shopping Line.“
           Text: “For further inquiries, please call“
           Text: “ npr 1 800-SHOPPER.  npc“
           Text: “Goodbye.“
21.  Disconnect
22.  Quit
     TRANSFER_CALL:
23.  Announce
        Speak With Interrupt
          Text: “Your call will be transferred to the next
                 available agent.“
24.  Transfer To AGENT_EXT Type: Intelligent
        Maximum Number of Rings: 3
        Case: “Answer”
           Complete
        Case: “Busy”
           Reconnect
        Case: “No Answer”
           Reconnect
        Case: “Error”
           Reconnect
     End Transfer
25.  Disconnect
26.  Quit
```

# Recognition Post-Processing

# E

## Overview

This appendix describes

- Data interface processes (DIPs) associated with WholeWord and FlexWord™ speech recognition
- Five processing features new to Version 6.0 that increase the system accuracy of recognition: Return Top 4, Prefix List Check, Luhn Check, Expected Value, and Close Match

## Purpose

The purpose of this appendix is to help you understand how these processing features can be beneficial in increasing recognition accuracy and how to use them in your application.

## Further Processing

There is a data interface process (DIP) associated with DPR, WholeWord speech recognition, and FlexWord speech recognition. This DIP is called **recog_dip** and is located in **/vs/bin/vrs/recog_dip**. The **recog_dip** is an optional, post-processing mechanism for enhancing the performance of some recognition tasks. It is intended to furnish additional recognition processing beyond what is already provided by the models and grammars of the recognizer. Whether or not the **recog_dip** is used during a particular recognition task is determined in the **/vs/data/sr_file** file. To determine if the **recog_dip** is activated for a particular grammar, check the **sr_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 `Min and Max` digit constraint. 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 accuracy improvement.

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 E-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 E-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 is 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)

Return Top 4 is a processing feature that returns up to four choices, separated by the pound (#) character, of 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 assure greater correctness.

In the following examples of returns from Return Top 4 processing, N refers to a digit {0-9}.

NNN#NNN#NNN#NNN#     4 results of length 3

?#NNN#     best result was reject ("?"), with only 1 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 Return Top Four recognition DIP:

```
start:
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
        Use 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
```

There are many important points in the above example:

- In the Prompt & Collect action, the `Recog:` is set to **RT4_DIG**. To enable the Return Top Four DIP, the first three characters of the `Recog:` 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`. Note that `parse` also strips the trailing “#” symbol from the value.

## Prefix List Check (LSC)

The list check 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 will mark as valid only those results which 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.” A check digit is a 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 you get 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 which accompanies the WholeWord recognition package.

## Expected Value (EXV)

An expected value for a recognition can be passed from a script to the recognition post-processing function. The expected value 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 & 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, if at a point in the call flow it was been determined that the caller is located in Columbus, Ohio, and the caller is asked to speak his or her US English five digit zip code, then the expected value can be set to 432 because 432 are the first three digits for all zip codes in this city. 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)

This is a function available to the customer-editable post processing which 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 close match and expected value can be used to improve application accuracy. For example, at a point in the call flow the caller has been identified and a caller profile read into the script. If the script requests verification of the house number of the caller's address, the expected value 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 expected value 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 accept the result as correct.

## Further Processing Prefixes and Combinations

Each of the above features and all combinations are invoked by specifying a custom recognition type. Selection of the functionality is done by naming the recognition type with an appropriate prefix.

Table E-1 shows the prefix names for the further processing functions.

**Table E-1. Prefix Names For Further Processing Functions**

| Function | Prefix |
|---|---|
| Return top 4 results to script | RT4 |
| Luhn (**mod10**) checksum calculation | LNC |
| Prefix list check | LSC |
| Expected value checking | EXV |
| Close match checking | CLM |

For combinations, just continue to place prefixes at the beginning of the recognition type.

Table E-2 shows examples and explanations of combination prefixes/functions.

**Table E-2. Combination Prefixes and Functions**

| Function | Prefix |
|---|---|
| Return (up to) top 4 results that satisfy Luhn check | LNCRT4 |
| Return (up to) top 4 results that satisfy close match checking | CLMRT4 |
| Return (up to) top 4 results that satisfy Luhn check and list check | LNCLSCRT4 |

# Abbreviations

## A

**AC**
Alternating current

**ACD**
Automatic call distributor

**AD**
Application dispatch

**AD-API**
Application dispatch application programming interface

**ADPCM**
Adaptive differential pulse code modulation

**ADU**
Asynchronous data unit

**AGL**
Application generation language

**ALERT**
System alerter process

**ANI**
Automatic number identification

**API**
Application programming interface

**ARU**
Alarm relay unit

**ASAI**
Adjunct/Switch Application Interface

**ASCII**
American Standard Code for Information Interchange

**ASI**
Analog switch integration

**ASR**
Advanced Speech Recognition

**AYC2C**
The signal processor (SP) circuit card

**AYC3B**
A T1 (digital) circuit card

**AYC5B**
The IVP6 Tip/Ring (analog) circuit card

**AYC6B**
The IVP4 Tip/Ring (analog) circuit card.

**AYC7**
The companion (CMP) circuit card.

**AYC9**
The Text-to-Speech circuit card

**AYC10**
The IVC6 Tip/Ring (analog) circuit card

**AYC11**
A T1 (digital) circuit card

**AYC16**
The IVP6-IU Tip/Ring (analog) circuit card

**AYC21**
The E1/T1 (digital) circuit card

**AYC26**
The IVP6-IA Tip/Ring (analog) circuit card

**AYC27**
The IVP6-ID Tip/Ring (analog) circuit card

**AYC28**
The IVP6 Tip/Ring (analog) circuit card

**AYC30**
The NGTR (analog) circuit card

**AYC43**
The speech and signal processor (SSP) circuit card

## B

**BB**
Bulletin board

**bps**
Bits per second

**BRDG**
Call bridging process

**BSC**
Binary synchronous communication

## C

**CCA**
Call classification analysis

**CDH**
Call data handler

**CELP**
Code Excited Linear Prediction

**CGEN**
Voice system general message class

**CICS**
Customer Information Control System

**CMP**
Companion circuit card

**CMS**
Call Management System

**CO**
Central office

**CPE**
Customer provided equipment or customer premise equipment

**CPN**
Calling party number

**CPT**
Call progress tones

**CPU**
Central processing unit

**CSU**
Channel service unit

**CVS**
Converse vector step

# D

**dB**
Decibel

**DB**
Database

**DBC**
Database checking process

**DBMS**
Database management system

**DC**
Direct current

**DCE**
Data communications equipment

**DCP**
Digital communications protocol

**DIMM**
Dual in-line memory module

**DIO**
Disk input and output process

**DIP**
Data interface process

**DMA**
Direct memory access

**DNIS**
Dialed number identification service

**DPR**
Dial Pulse Recognition

**DSP**
Digital signal processor

**DTE**
Data terminal equipment

**DTMF**
Dual tone multifrequency

**DTR**
Data terminal ready

# E

**E&M**
Ear and Mouth

**EBCDIC**
Extended Binary Coded Decimal Interexchange Code

**ECS**
Enterprise Communications Server

**EIA**
Electronic Industries Association

**EISA**
Extended Industry Standard Architecture

**EMI**
Electromagnetic interference

**ESD**
Electrostatic discharge

**ESDI**
Extended Serial Data Interface

**ESS**
Electronic Switching System

**ET**
Error tracker

**EXTA**
External alarms feature message class

## F

**FCC**
Federal Communications Commission

**FDD**
Floppy disk drive

**FEP**
Front end processor

**FFE**
Form Filler Plus feature message class

**FIFO**
First-in-first-out processing order

**foos**
Facility out-of-service state

**FTS**
File transfer process message class

## G

**GEN**
PRISM logger and alerter general message class

**GSE**
Graphical Speech Editor

**GUI**
Graphical user interface

## H

**HDD**
Hard disk drive

**HLLAPI**
High Level Language Application Programming Interface

**HOST**
Host interface process message class

**hwoos**
Hardware out-of-service state

**Hz**
Hertz

## I

**IBM**
International Business Machines

**ICK**
Integrity checking process message class

**ID**
Identification

**IDE**
Integrated Disk Electronics

**IE**
Information element

**INIT**
Voice system initialization message class

**inserv**
In-service state

**IPC**
Interprocess communication

**IPC**
Intelligent Ports Card (IPC-900)

**IPCI**
Integrated personal computer interface

**IRAPI**
INTUITY Response Application Programming Interface

**IRQ**
Interrupt request

**ISA**
Industry Standard Architecture

**ISDN**
Integrated Services Digital Network

**ISV**
Independent Software Vendor

**ITAC**
International Technical Assistance Center

**IVP4**
Integrated Voice Processing card with 4 analog channels

**IVP6**
Integrated Voice Processing card with 6 analog channels

**IVPSS**
Integrated Voice Processing System Software

## K

**Kbps**
Kilobites per second

**Kbyte**
Kilobyte

## L

**LAN**
Local area network

**LDB**
Local database

**LED**
Light-emitting diode

**LIFO**
Last-in-first-out processing order

**LN**
Load number

**LOG**
VIS logger process message class

**LSE1**
Line side E1

**LST1**
Line side T1

**LU**
Logical unit

## M

**manoos**
Manually out-of-service state

**MAP/100**
Multi-Application Platform 100

**MAP/100C**
Multi-Application Platform 100C

**MAP/40**
Multi-Application Platform 40

**Mbps**
Megabits per second

**Mbyte**
Megabyte

**MF**
Multifrequency

**ms**
Millisecond

**msec**
Millisecond

**MHz**
Megahertz

**MTC**
Maintenance process

## N

**NCP**
Network Control Program

**NEBS**
Network Equipment Building Standards

**NEMA**
National Electrical Manufacturers Association

**netoos**
Network out-of-service state

**NFAS**
Non-Facility Associated Signaling

**NFS**
Network file sharing

**NGTR**
Next Generation Tip/Ring

**NMVT**
Network Management Vector Transport

**NM-API**
Network Management - Application Programming Interface

**nonex**
Nonexistent state

**NRZ**
Non Return to Zero

**NRZI**
Non Return to Zero Inverted

## O

**OEM**
Original equipment manufacturer

**OGA**
Operator generated alert

## P

**P & C**
Prompt and Collect

**PBX**
Private branch exchange

**PC**
Personal computer

**PCB**
Printed circuit board

**PCM**
Pulse code modulation

**PEC**
Price element code

**PRI**
Primary rate interface

**PSTN**
Public switch telephone network

**PS&BM**
Power supply and battery module

## R

**RAM**
Random access memory

**RECOG**
Speech recognition feature message class

**RDBMS**
ORACLE relational database management system

**REN**
Ringer equivalence number

**RFS**
Remote file sharing

**RM**
Resource manager

**RMB**
Remote maintenance circuit card

**RTS**
Request to send

## S

**SBC**
Sub-band coding

**SCCS**
Switching Control Center System

**SCSI**
small computer system interface

**SDLC**
Synchronous Data Link Control

**SDN**
Software Defined Network

**SID**
Station identification

**SIMM**
Single in-line memory module

**SLIP**
Serial Line Interface Protocol

**SNA**
Systems Network Architecture

**SNMP**
Simple Network Management Protocol

**SP**
Signal processor circuit card

**SPIP**
Signal processor interface process

**SPPLIB**
Speech processing library

**SQL**
Structured Query Language

**SR**
Speech recognition

**SYS**
UNIX system calls message class

**sysgen**
System generation

## T

**TAS**
Transaction Assembler Script

**TCC**
Technology Control Center

**TCP/IP**
Transmission control protocol/internet protocol

**TDM**
Time division multiplexing

**TE**
Terminal emulator

**THR**
Threshold message class

**TKR**
Token Ring

**TLI**
Transport layer interface

**TLP**
Transmission level plan

**T/R**
Tip/Ring circuit card

**TRIP**
Tip/Ring interface process

**TSO**
Technical Service Organization

**TSO**
time share operation

**TSM**
transaction state machine process

**TTS**
Text-to-Speech

**TWIP**
T1 interface process

## U

**UK**
United Kingdom

**US**
United States of America

**USOC**
Universal service ordering code

**UVL**
Unified Voice Library

## V

**VDC**
Video display controller

**VPC**
Voice processing comarketer

**VRU**
Voice response unit

**VROP**
Voice response output process

# 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

**adaptive differential pulse code modulation (ADPCM)**
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 (ASAI)**
An optional feature package that provides an Integrated Services Digital Network-based interface between Lucent Technologies PBXs and adjunct processors.

**advanced speech recognition (ASR)**
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.

**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.

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

**American Standard Code for Information Interchange (ASCII)**
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.

**announcement**
A message the system plays to the caller to provide information. The caller is not asked to give a response. Compare to “prompt.”

**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 Graphical Designer, 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.

**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 (ACD)**
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 (ANI)**
A method of identifying the calling party by automatically receiving a string of digits that identifies the calling station of a particular customer.

**AYC2C**
The signal processor (SP) circuit card.

**AYC3B**
A T1 (digital) circuit card.

**AYC5B**
The IVP6 Tip/Ring (analog) circuit card.

**AYC6B**
The IVP4 Tip/Ring (analog) circuit card.

**AYC7**
The companion (CMP) circuit card.

**AYC9**
The Text-to-Speech circuit card.

**AYC10**
The IVC6 Tip/Ring (analog) circuit card.

**AYC11**
A T1 (digital) circuit card.

**AYC16**
The IVP6-IU Tip/Ring (analog) circuit card.

**AYC21**
The E1/T1 (digital) circuit card.

**AYC26**
The IVP6-IA Tip/Ring (analog) circuit card.

**AYC27**
The IVP6-ID Tip/Ring (analog) circuit card.

**AYC28**
The IVP6 Tip/Ring (analog) 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.

**binary synchronous communications (BCS)**
A character-oriented synchronous link protocol.

**blind transfer protocol**
A protocol in which a call is completed as soon as the extension is dialed, without having to wait to see if the telephone is busy or if the caller answered.

**bridging**
The process of connecting one telephone network connection to another 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.

**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 (CCA)**
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 (CHD) 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 progress tones (CPT)**
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.

**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.

**central office (CO)**

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 (CPU)**

See “processor.”

**channel**

See “port.”

**channel associated signaling (CAS)**

A type of signaling that can be used on E1 circuit cards. It occurs on channel 16.

**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.

**CMP (AYC7)**

The companion circuit card to the signal processor (SP).

**code excited linear prediction (CELP)**

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.

**CompuLert/SCCS interface**

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

**configuration**

The arrangement of the software and hardware of a computer system or network. The INTUITY CONVERSANT 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.

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

**custom speech**
Unique words or phrases to be used in INTUITY CONVERSANT 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 (CICS)**
Part of the operating system that manages resources for running applications (for example, IND$FILE). Note that TSO and CMS provide analogous functionality in other host environments.

# D

**danger**
An admonishment or advisory statement used in INTUITY CONVERSANT system documentation to alert the user to the possibility of personal injury or death.

**data interface process (DIP)**
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.

**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 (DPR)**
A method of recognizing caller pulse inputs from a rotary telephone.

**dialed number identification service (DNIS)**
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.

**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.

**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 (DTMF)**

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

**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 (E&M)**

A common T1 trunking protocol for connection between two "switches."

**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."

**editor system**

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

**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 (ESDI)**

A software- and hardware-controlled method used to store data on magnetic peripherals.

**Enterprise Communications Server (ECS)**

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.

**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.

**external actions**

Specific predefined system tasks that Graphical Designer or 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.

**external functions**

Specific predefined (or customer-created) system tasks that can Graphical Designer 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 (they are not provided in a choices list). Examples are concat, getarg, length, substring, etc.

## F

**FAX Actions**

An optional feature package that allows the system to send fax messages.

**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.

**field**

See “database field.”

**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.”

**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.

**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

**grammar**
The inputs that a recognizer can match (identify) from a caller.

**Graphical Speech Editor (GSE)**
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.

**Graphical Designer**
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*).

## 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.

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

**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.

# I

**iCk**
The system integrity checking process.

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

**IND$FILE**
The standard SNA file transfer utility that runs as an application under CICS, TSO, and CMS. IND$FILE is independent of link-level protocols such as BISYNC and SDLC.

**independent software vendor (ISV)**
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 (ISA)**
A PC bus standard that allows processors and other circuit cards to communicate with each other.

**initialize**
To start up the system for the first time.

**Integrated Services Digital Network (ISDN)**
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 IVP4 or 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 (IRAPI)**
A library of commands that provide a standard development interface for voice-telephony applications.

**IVC6 circuit card (AYC10)**
A Tip/Ring (analog) circuit card with six channels.

**IVP4 circuit card (AYC6 or AYC6B)**
A Tip/Ring (analog) card with four channels.

**IVP6 circuit card (AYC5, AYC5B, or AYC28)**
A Tip/Ring (analog) card with six channels.

## K

**keyboard mapping**
In emulation mode, this feature enables the keyboard to send 3270 keyboard codes to the host according to a configuration table set up during installation.

**keyword spotting**
A capability provided by WholeWord speech recognition that allows the system to recognize a single word in the middle of an entire phrase spoken by a caller in response to a prompt.

## L

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

**line side E1 (LSE1)**
A digital method of interfacing an INTUITY CONVERSANT system to a PBX or "switch" using E1-related hardware and software.

**line side T1 (LST1)**
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 (LAN)**
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.

**logical unit (LU)**
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.

## 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 (MTC)**

A software process that runs temporary diagnostics and maintains the state of circuit cards and channels.

**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.

**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.

**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-DOS**

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

**multifrequency (MF)**

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

**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 (NGTR) Tip/Ring (AYC30) circuit card**
An analog circuit card with six channels.

**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.

**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.

**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 (OAG)**
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

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

**peripheral component interconnect (PCI)**
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."

**Primary Rate Interface (PRI)**
An ISDN term for connections over E1 or T1 facilities that are usually treated as trunks.

**private branch exchange (PBX)**
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.

**pulse code modulation (PCM)**

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

# R

**record**

See “database record.”

**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.

**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 and MAP/40 purchases.

**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.

**roll back**

To cancel changes to a database since the point at which changes were last committed.

**rollback segment**

A portion of the database that records actions that should be undone under certain circumstances. Rollback segments are used to provide transaction rollback, read consistency, and recovery.

# S

**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.

**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.

**signal processor (SP) 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).

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

**single-threaded application**
An application that runs on a single voice channel.

**slave**
A circuit card that depends on the TDM bus for clock information.

**small computer system interface (SCSI)**
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.

**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 (SSP) 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 (SR)**
The ability of the system to understand input from callers.

**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 (SQL)**
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 “whole-word technology.”

**switch**
A software and hardware device that controls and directs voice and data traffic. A customer-based switch is known as a "private branch exchange (PBX).

**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."

**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).

**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 (TTS)**
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 Graphical Designer 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 (T/R)**
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 (TAS)**

The computer program code that controls the application operating on the voice response system. The code can be produced from Graphical Designer, Script Builder, or by writing directly in TAS code.

**transaction state machine (TSM) process**

A multi-channel IRAPI application that runs applications controlled by TAS script code.

**transient process**

A process that is created dynamically only when needed.

**troubleshooting**

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

## U

**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.

## V

**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 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 (VROP)**
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 (VRU)**
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.

# 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 (SR)**
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 FlexWords that are available for recognition by an application during a Prompt & Collect action step.

**word spotting**

The ability to search through extraneous speech during a recognition.

# Index

## A

Accessing the graphical speech editor, 3-3
Accuracy
    negative influences on FlexWord speech recognition, 5-32
    positive influences on FlexWord speech recognition, 5-28
Accuracy of FlexWord speech recognition, 5-28
Activating a FlexWord wordlist, 5-26
Adaptive differential pulse code modulation, 3-2
Adding a second speech disk
    reference for, 1-4
Adding words and phrases to a FlexWord vocabulary, 5-21
Administering and installing the mouse, 3-3, 5-13
ADPCM-16, B-3
ADPCM-32, B-3
Alpha characters
    recording, 2-14
application using escape sequences
    text-to-speech, D-10

## B

Barge-in for WholeWord speech recognition, 4-12
Bilingual speech recognition capabilities, 4-3
Building a FlexWord vocabulary, 7-3

## C

C application
    installing speech, 2-17
Caller error
    FlexWord speech recognition, 7-4
changing class detection
    using escape sequences in text-to-speech, D-8
Changing the size of the speech filesystem
    reference for, 1-4
Channels
    adding a second speech disk, 1-4
Choosing a FlexWord vocabulary, 7-3
Coding rates for speech phrases, 1-2
Combining WholeWord speech recognition and touch tone, 7-3
Commands
    spres, 1-3
    spsav, 1-3
Computing channel numbers
    FlexWord toolkit, C-2
Concepts about speech, 1-4
Configuring hardware for the FlexWord toolkit, 5-13
Connected digit recognition
    WholeWord speech recognition accuracy, 4-4
Connected digit recognition for WholeWord speech recognition, 4-9
Considering the sound of a word in a FlexWord speech recognition application, 7-4
Converting speech files for editing, 3-21
Copying wordlists with the FlexWord toolkit, 5-22
Corrupted speech disk
    recovering, 1-4
Creating filesystems
    reference for, 1-4
Creating FlexWord wordlists, 5-16
Creating speech
    overview of, 2-2
Creating wordlists
    FlexWord speech recognition, 5-11
Custom DIPs for WholeWord speech recognition, 4-14
Custom phrase tags
    definition of, 1-6
Custom recognition types for WholeWord speech recognition, 4-10
Custom speech
    producing with Script Builder, 2-9
Custom speech package, 2-10
Custom vocabularies for WholeWord speech recognition, 4-14
Customizing speech with text-to-speech, 2-11

## D

Deactivating wordlists, 5-27
Decreasing the size of the speech filesystem
    reference for, 1-4
Default speech directory, 1-2
Deleting words and wordlists from a FlexWord vocabulary, 5-22
Designing a FlexWord speech recognition application, 5-8
Developing speech
    overview of, 2-2
Dial pulse recognition
    barge-in, 6-4
    capabilities, 6-2
    components of, 6-2
    confirmation, 6-4
    failure modes, 6-8
    how works, 6-5
    introduction to, 6-2
    processing, 6-5
    recognition accuracy, 6-6
    recognition types, 6-3
    recognizer, 6-5
    training, 6-7
    troubleshooting, 6-7
Digitizing speech phrases, 2-16

Digitizing speech with Script Builder, 2-16
Digitizing with the graphical speech editor, 2-16
Directory
- default speech, 1-2
- default speech filesystem, 1-7

Disk
- adding a second disk for speech, 1-4
- recovering from a corrupted speech disk, 1-4

Displaying file system attributes
- reference for, 1-4

Dual tone multifrequency support for WholeWord speech recognition, 4-13

E

Editing a speech file with the graphical speech editor, 3-13
Editing with Script Builder, 3-26
Enhanced Basic Speech Formats, A-1
- australian english, A-4
- brazilian portuguese, A-6
- canadian french, A-8
- cantonese chinese, A-12
- castilian spanish, A-14
- dutch, A-16
- french, A-18
- german, A-21
- hindi, A-23
- japanese
  , A-25
- latin-american, A-26
- mandrian, A-29
- uk english, A-31
- us english, A-33

Equipment
- recording, 2-13

Error messages
- FlexWord toolkit, 5-18

Escape sequences
- sample Text-to-Speech application, D-10
- Text-to-Speech, D-2

Examples of FlexWord applications, 7-5
Experienced calling population, 5-29

F

Falling inflection, 2-7
Files
- restoring, 1-3
- saving, 1-3

Filesystem
- changing the size of, 1-4
- creating and removing, 1-4
- speech, 1-2

Filesystem attributes
- displaying, 1-4

Fixed-length connected digit
- WholeWord speech recognition, 4-9

FlexWord applications requiring number-intensive wordlists, 7-5
FlexWord Editor, 5-16
FlexWord speech recognition
- accuracy, 5-28
- activating wordlists, 5-26
- building a FlexWord vocabulary, 7-3
- caller error, 7-4
- choosing a FlexWord vocabulary, 7-3
- considering synonyms, 7-4
- considering the sound of a word, 7-4
- deactivating wordlists, 5-27
- FlexWord toolkit, 5-6
- getting the most out of FlexWord, 7-3
- installing wordlists, 5-25
- introduction, 5-2
- keyword spotting, 5-7
- overview of, 1-12
- phrase screening, 5-7
- preparing an application, 5-8
- prompting for input, 7-4
- removing wordlists, 5-27
- using with text-to-speech, 7-7
- using with WholeWord speech recognition, 7-9
- word length, 7-4

FlexWord toolkit
- adding words and phrases to a vocabulary, 5-21
- configuring hardware, 5-13
- configuring the text-to-speech circuit card, 5-14
- connecting the phone line, 5-14
- copying wordlists, 5-22
- creating and modifying wordlists, 5-16
- deleting words and wordlists from a vocabulary, 5-22
- error messages, 5-18
- for FlexWord speech recognition, 5-6
- overview of, 1-13
- software requirements, 5-12

FlexWord versus WholeWord speech recognition, 7-11
FlexWord vocabulary administration, 5-23
Framing
- examples of, 2-5
- how to use in voice scripts, 2-5
- placing words, 2-6
- voiceless stops and sounds, 2-6

G

Getting the most out of text-to-speech, 7-6
Graphical speech editor, 2-11
- accessing, 3-3
- adjusting the volume, 3-19
- channel numbers, 3-4
- copying and pasting speech, 3-18

- cutting and pasting speech, 3-17
- edit menu, 3-7
- exiting, 3-20
- file menu, 3-6
- information bar, 3-12
- introduction, 3-2
- listening to speech, 3-16
- menu bar, 3-5
- opening a file, 3-14
- options, 2-11
- overview of producing and editing speech, 1-9
- requirements of, 1-10
- saving changes, 3-20
- scroll bar, 3-11
- selecting a segment of speech, 3-15
- setting the marker, 3-15
- speech display, 3-10
- speech energy, 3-10
- speech envelope, 3-10
- supported speech encoding formats, 3-2
- title bar, 3-5
- tool bar, 3-8
- undoing changes, 3-19
- user interface, 3-4, 3-5
- using the mouse, 3-13
- voice menu, 3-6
- window, 3-4

Gse command, 3-4
GSE features
- copy, 2-11
- cut, 2-11
- output volume, 2-11
- record, 2-11

gse_add command, 3-21
gse_addpl, 3-21
gse_copy command, 3-21
gse_copypl command, 3-21

## H

Hardware installation of AYC9 SP
- Text-to-Speech, 1-8

Hiring a professional speaker
- guidelines for, 2-8

## I

Importing speech, 2-12
Increasing the size of the speech filesystem
- reference for, 1-4

Inflection
- falling, 2-7
- medial, 2-7
- rising, 2-7

Inflections
- recording, 2-15
- speech, 2-7

Initiating the graphical speech editor, 3-4
Installing
- administering the mouse, 3-3
- FlexWord speech recognition wordlist, 5-25
- speech in a C Application, 2-17
- speech in a Script Builder application, 2-17
- speech in a TSM application, 2-17
- speech in an IRAPI application, 2-17

## K

Keyword spotting for FlexWord speech recognition, 5-7

## L

Listening to speech with the graphical speech editor, 3-16
Listening to the phonetic pronunciation of a word, 5-19
Lucent Technology custom vocabulary service for FlexWord speech recognition, 5-6

## M

Medial inflection, 2-7
Menu-based FlexWord speech recognition applications, 7-5
Models
- definition of, 4-8

Modifying FlexWord wordlists, 5-16
Mouse
- installing and administering, 3-3, 5-13
- using with GSE, 3-2

Mu-law PCM format, B-2

## N

Negative influences on WholeWord speech recognition accuracy, 4-17
Numbers
- recording, 2-14

## O

O.S. Index
- creating an, C-2

Opening a file with the FlexWord toolkit, 5-19
Output volume, 3-10

P

Phonemes
    definition of, 5-2
Phonetic spelling
    viewing, 5-17
Phrase rejection
    WholeWord speech recognition, 4-11
Phrase screening
    FlexWord speech recognition, 5-7
    WholeWord speech recognition, 4-11
Phrases
    digitizing, 2-16
    long, 2-4
    planning the voice script, 2-4
    short, 2-4
Planning
    a recording session
        environmental conditions, 2-13
        equipment specifications, 2-13
    the voice script
        suggestions for, 2-3
        types of phrases, 2-4
Play, 3-9
Positive influences on WholeWord speech recognition accuracy, 4-15
Predefined phrase tags
    Script Builder, 1-6
Preparing a FlexWord application, 5-8
Prerecorded speech
    using with text-to-speech, 7-7
Processing recorded speech
    overview of, 2-2
Producing self-recorded custom speech, 2-9
Professional speaker
    hiring, 2-8
Prompting for input
    FlexWord speech recognition, 7-4
Prompts
    informative, 5-30
    structure and design, 5-29
Pulse code modulation, B-2
Purchasing
    a custom speech package, 2-10
    Lucent Technology custom speech package
        creating wordlists, 5-12
    standard speech package, 2-9

R

Recognition confirmation for WholeWord speech recognition, 4-11, 6-4
Record, 3-9
Recording
    alpha and numeric characters with inflections, 2-15
    alpha characters, 2-14
    equipment, 2-13
    magnetic tape, 2-13
    numbers, 2-14
    session
        environmental conditions, 2-13
    speech
        options for, 2-13
        with a professional speaker, 2-13
    spoken phrases on magnetic tape, 2-13
Recovering from a corrupted speech disk
    reference for, 1-4
Rejection
    FlexWord speech recognition, 5-7
Removal of
    filesystems, 1-4
    FlexWord speech recognition wordlists, 5-27
Resources
    related, xix
Restoring files, 1-3
Retrieving files from speech pools, 3-22
Rising inflection, 2-7
Risky mode, D-6, D-8
    off, D-6
    when to use, D-7

S

Sample application using escape sequences
    Text-to-Speech, D-10
Saving files, 1-3
Script Builder
    editing speech files, 3-26
    installing speech, 2-17
    overview of producing speech, 1-7
Second speech disk
    adding, 1-4
Sharing speech, 2-12
Silence delays
    using escape sequences in Text-to-Speech, D-2
Software requirements
    FlexWord toolkit, 5-12
Space requirements for speech phrases, 1-2
Speaker
    hiring a professional, 2-8
Speaker independence for WholeWord speech recognition, 4-11

Speech
- filesystem
  - changing the size of, 1-4
- importing, 2-12
- inflections, 2-7
- installing, 2-17
- installing in a C application, 2-17
- sharing, 2-12

Speech administration tools, 1-3
Speech and Signal Processor (SSP), 4-2
- memory supported, 1-16

Speech development
- features, 1-7
- hiring a professional speaker, 2-8
- importing speech, 2-12
- producing self-recorded custom speech, 2-9
- purchasing the custom speech package, 2-10
- selecting a method, 2-7
- sharing speech, 2-12
- using Text-to-Speech, 2-10
- using text-to-speech, 2-10
- using the graphical speech editor, 2-11

Speech directory
- default, 1-2
- organization of, 1-2
- specifying, 1-3

Speech disk
- recovering from a corrupted, 1-4

Speech encoding formats, B-2
Speech files
- definition of, 1-7
- formats, B-2

Speech filesystem, 1-2
- definition of, 1-7

Speech phrases
- coding rates, 1-2
- digitizing, 2-16
- space requirements, 1-2

Speech pools
- retrieving files from, 3-22

Speech processing
- overview of, 2-2

Speech recognition
- bilingual speech recognition capabilities, 4-3
- limitations, 5-8
- overview of, 1-11
- sub-word, 5-2

Speech terminology, 1-4
SPEECHDIR variable, 1-3
SSP, see Speech and Signal Processor
Standard speech package, 2-9
Standard vocabulary
- WholeWord speech recognition, 4-4

Stop, 3-9
Storing speech files, 1-2
Sub band coding, B-2
Synonyms
- FlexWord speech recognition, 7-4

Synthesized speech, 1-8

**T**

Talkfiles
- definition of, 1-6
- extension of, 1-6

Telephone, 3-8
Text Classes
- using escape sequences in Text-to-Speech, D-6

Text-to-speech
- customizing, 2-11
- examples of use, 1-8
- functionality, 2-11
- getting the most out of text-to-speech, 7-6
- how to use, 1-8
- overview of producing speech, 1-8
- requirements of, 1-8
- software installation of, 1-8
- technology, 2-11
- using with FlexWord speech recognition, 7-7

Tool bar
- output volume, 3-10
- play, 3-9
- record, 3-9
- stop, 3-9
- telephone, 3-8

Tools for speech administration, 1-3
Transaction
- definition of, 2-3
- determining, 2-3

TSM application
- installing speech, 2-17

**U**

Using escape sequences with Text-to-Speech, D-2
Using FlexWord speech recognition and text-to-speech, 7-7
Using Script Builder for editing, 3-26
Using the FlexWord toolkit to create wordlists, 5-12
Using TTS and prerecorded speech together, 7-7
Using wordlists in FlexWord speech recognition scripts, 5-9

**V**

Variable-length connected digit
- WholeWord speech recognition, 4-9

Viewing the phonetic spelling of a word, 5-17
Voice script
- planning, 2-3
- writing, 2-5

Voiceless stops

writing the voice script, 2-6

W

WholeWord custom vocabulary
overview of, 1-12
WholeWord speech recognition
accuracy, 4-15
associated DIP, E-2
barge-in, 4-12
bilingual speech recognition capabilities, 4-3
computational processing, 4-20
connected digit recognition, 4-9
custom DIPs, 4-14
custom recognition types, 4-14
custom vocabularies, 4-14
dual tone multifrequency support, 4-13
fixed-length and variable-length connected digits, 4-9
getting the most out of WholeWord, 7-2
grammars, 4-8
how it works, 4-19
introduction, 4-2
models, 4-8
overview of, 1-11
phrase screening, 4-11
recognition confirmation, 4-11, 6-4
speaker independence, 4-11
standard vocabulary, 4-4
the recognizer, 4-19
types, 4-5
using with FlexWord speech recognition, 7-9
WholeWord standard vocabulary
overview of, 1-12
WholeWord versus FlexWord speech recognition, 7-11
wl_copy command, 5-16
wl_edit command, 5-16
wl_init command, 5-16
Word length
FlexWord speech recognition, 7-4
Word meaning
FlexWord speech recognition, 7-4
Word spotting
WholeWord speech recognition, 4-10
Wordlists
creating, 5-12
effective construction of, 5-28
ineffective, 5-33
name tag characters, 5-5
using in scripts, 5-11
words per wordlist, 5-5
Writing the script
framing, 2-5
Writing the voice script
suggestions for, 2-5